UNIVERSIDADE ESTADUAL DO SUDOESTE DA BAHIA
DEPARTAMENTO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

ALEXANDRE ANTONIO FRANÇA SOUZA

SEGMENTAÇÃO DE PISTA USANDO DEEP LEARNING

Vitória da Conquista, Bahia, Brasil
2024

ALEXANDRE ANTONIO FRANÇA SOUZA

SEGMENTAÇÃO DE PISTA USANDO DEEP LEARNING

Trabalho de conclusão de curso, apresentado ao curso de Ciências da Computação, da Universidade do Sudoeste da Bahia, em Vitória da Conquista - Bahia, como requisito parcial para a obtenção do título de Bacharel em Ciências da Computação.

Orientador: Prof. Dr. Roque Mendes Prado Trindade

Coorientador: Mst. Iago Pachêco Gomes

Vitória da Conquista, Bahia, Brasil

Fevereiro - 2024

Este trabalho é dedicado à minha família e
amigos da UESB

AGRADECIMENTOS

Agradeço primeiramente a Deus, por me guiar com fé e perseverança ao longo desta jornada. Seu amparo incondicional foi meu refúgio e fortaleza nos momentos de dúvida e incerteza.

À minha família, expresso minha mais profunda gratidão. O amor, apoio e compreensão de vocês foram fundamentais para minha formação pessoal e acadêmica. Cada sacrifício compartilhado, cada palavra de incentivo e cada gesto de carinho pavimentaram o caminho que me trouxe até aqui.

Aos meus amigos do CIPEC, agradeço pela camaradagem, pelas trocas de conhecimento e pelos momentos compartilhados, tanto nos estudos quanto na vida. A amizade de vocês enriqueceu minha experiência acadêmica, tornando esta jornada não apenas educativa, mas também extremamente gratificante.

Aos meus colegas de curso, obrigado pela parceria e pelo ambiente colaborativo que construímos juntos. As discussões, os projetos em grupo e o compartilhamento de ideias foram essenciais para o meu desenvolvimento acadêmico e pessoal.

Aos meus professores, minha eterna gratidão pela orientação, paciência e conhecimento compartilhados. Vocês não apenas transmitiram informações valiosas, mas também inspiraram em mim a busca contínua pelo conhecimento e a paixão pela minha área de estudo. Sua dedicação e comprometimento foram fundamentais para o meu crescimento intelectual.

Este trabalho é o resultado de um esforço coletivo, e cada um de vocês tem uma parcela significativa nesta conquista. Minha jornada até aqui não teria sido a mesma sem a presença e o apoio de cada um de vocês. Por isso, meu sincero obrigado.

“Podemos apenas ver um curto trecho à frente,
mas podemos ver muito que precisa ser feito”
Alan Turing

RESUMO

Este estudo aborda a segmentação de faixas usando técnicas de aprendizagem profunda e destaca sua aplicação na navegação autônoma. Neste estudo, treinamos e avaliamos um modelo de segmentação semântica utilizando diversos conjuntos de dados, incluindo o conjunto de dados KITTI e imagens do campus da UESB. O objetivo principal é desenvolver algoritmos para identificar faixas e áreas navegáveis sob diferentes cenários e condições de iluminação, técnicas de transferência de conhecimento e aumento de dados. A relevância desta investigação relaciona-se com a crescente necessidade de soluções robustas de navegação autônoma que possam operar numa variedade de condições e situações ambientais e promover a segurança rodoviária, a eficiência dos transportes e a mobilidade nas cidades.

Palavras-chaves: KITTI, UESB, Segmentação semântica.

ABSTRACT

This study addresses lane segmentation using deep learning techniques and highlights its application in autonomous navigation. In this study, we trained and evaluated a semantic segmentation model using several datasets, including the KITTI dataset and UESB campus images. The main objective is to develop algorithms to identify navigable lanes and areas under different scenarios and lighting conditions, knowledge transfer techniques and data augmentation. The relevance of this research relates to the growing need for robust autonomous navigation solutions that can operate in a variety of environmental conditions and situations and promote road safety, transport efficiency and mobility in cities.

Key-words: KITTI, UESB, semantic segmentation

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

IA	Inteligência Artificial
ADAS	Advanced Driver Assistance Systems
JPL	Jet Propulsion Laboratory
RNA	Redes Neurais Artificiais
CNN	Convolutional Neural Network
FC	Fully Connected
Pooling	Agrupamento
Downsampling	Subamostragem
ROI	Region of Interest
GPU	Unidade de Processamento Gráfico
ResNeXt	Uma variante da arquitetura de rede neural
ResNet	Rede Neural Residual
RNG	Rede Neural Gráfica
RGC	Convolução de Grafos Residual
DRGC	Dupla Rede de Convolução de Grafos
RCN	Rede de Convolução Tradicional
RN4G	Rede Neural para Grafos
RTC	Rede Totalmente Convolucional
U-Net	Arquitetura de encoder-decoder simétrica
RefineNet	Uma arquitetura de rede neural para segmentação de imagens
VGG	Visual Geometry Group
ReNet	Abordagem que substitui camadas convolucionais por redes neurais recorrentes
DenseNet	Dense Convolutional Network

MobileNetV1/V2/V3	Arquiteturas de redes neurais otimizadas para dispositivos móveis
DeconvNet	Uma arquitetura de rede neural para segmentação de imagens
DeepLab	Arquitetura de rede neural convolucional para segmentação semântica
YOLO	You Only Look Once
LDWS	Lane Departure Warning System
IPM	Inverse Perspective Mapping
SciELO	Scientific Electronic Library Online
ECCV	European Conference on Computer Vision
arXiv	Um repositório de preprints eletrônicos
CRFs	Conditional Random Fields
IEEE	Institute of Electrical and Electronics Engineers
CVPR	Conference on Computer Vision and Pattern Recognition
WACV	Winter Conference on Applications of Computer Vision
PASCAL VOC	Pattern Analysis, Statistical Modeling and Computational Learning Visual Object Classes
ASPP	Atrous Spatial Pyramid Pooling
OpenCV	Open Source Computer Vision Library
Adam	Adaptive Moment Estimation
IoU	Intersection over Union

SUMÁRIO

1 INTRODUÇÃO

A inteligência artificial (IA) é um campo em constante evolução que busca emular a capacidade de raciocínio, aprendizado, percepção e criatividade humana, aplicando-a a máquinas. Segundo Eduka.AI (2023), a IA surgiu do desejo humano de criar máquinas capazes de pensar e agir como seres humanos, desenvolvendo-se ao longo do tempo para resolver problemas complexos e melhorar nossa forma de viver e trabalhar. Além disso, a IA tem o potencial de transformar praticamente todos os aspectos de nossa vida, automatizando tarefas e proporcionando novas formas de interação entre humanos e máquinas. Como destacado por SAS (2023), a inteligência artificial desempenha um papel crucial na sociedade contemporânea, revolucionando diversos setores por meio da automação e da análise avançada de dados. A conferência de Dartmouth de 1956, considerada o nascimento da inteligência artificial (IA), foi um evento seminal que reuniu pesquisadores para discutir o futuro da IA. O termo "inteligência artificial" foi cunhado durante este encontro, organizado por John McCarthy, Marvin Minsky, Nathaniel Rochester e Claude Shannon. Eles propuseram que todos os aspectos da aprendizagem ou qualquer outra característica da inteligência poderiam, em teoria, ser tão precisamente descritos que uma máquina poderia ser feita para simular tais capacidades (Dartmouth College, 1956).

Os carros autônomos e a robótica móvel avançaram rapidamente graças aos avanços na visão computacional, na inteligência artificial e no aprendizado de máquina. A percepção é especialmente importante na robótica móvel porque esses dispositivos estão em um ambiente dinâmico e expostos a obstáculos. Neste contexto, diversas técnicas foram desenvolvidas para melhorar a percepção da máquina. Com o rápido desenvolvimento da indústria automóvel global, o problema da identificação de estradas ou faixas é um factor importante para permitir sistemas avançados de assistência ao condutor (ADAS)(LIANG et al., 2020) e condução autónoma. A figura 1 mostra exemplos de veículos autônomos usados em pesquisa e indústria.

Atualmente os veículos autônomos utilizam diversos sensores como GPS, LIDAR e câmeras para navegar em ruas e estradas. Com o desenvolvimento do transporte inteligente, a percepção do ambiente, como uma tarefa essencial para a condução autônoma, tornou-se um foco de pesquisa (TANG; LI; LIU, 2021). Dentre as diversas tarefas de percepção, a segmentação das faixas e áreas navegáveis é um componente importante para garantir a segurança dos passageiros e outros participantes do trânsito.Muitos esforços foram feitos nas últimas décadas, visto que, há muitas variáveis, como neblina, chuva, variação de iluminação e oclusão parcial. Neste sentido, as redes neurais artificiais são ótimas em resolver problemas de classificação. No contexto da classificação de pista, técnicas mais sofisticadas precisam ser utilizadas para delimitar com exatidão a região da pista e assim fornecer a um agente autônomo a capacidade de se locomover autonomamente (TANG; LI; LIU, 2021).

Figura 1 – Exemplos de veículos robóticos.

(a) Stanley

(b) CARINA II

(c) UGV Google

Fonte: (WILLIAMS, 2024; MATSUBARA, 2024; Laboratório de Robótica Móvel - ICMC-USP, 2024)

Diferentes técnicas são utilizadas na navegação autônoma, entre elas a navegação visual que busca extrair características do ambiente para tomada de decisão. Com as técnicas de segmentação semântica é possível classificar uma região de uma pista, mostrando para a uma rede neural a região navegável (LIANG et al., 2020). No processo de segmentação, diversos cálculos são realizados para obtenção da saída desejada. No que se refere a segmentação de pista, criar um sistema capaz de gerar imagens de segmentadas em tempo real envolve muito processamento.

Esse trabalho abordará a segmentação de pista por meio das técnicas de aprendizado de máquina e visão computacional, para isso, diferentes conjuntos de dados serão utilizados para fornecer à rede uma maior capacidade de generalização. Esta pesquisa discute as técnicas utilizadas na segmentação semântica de pista, utilizando diferentes bases de dados, como o KITTI Dataset (GEIGER et al., 2013) e imagens do campus da UESB, onde se busca avaliar posteriormente o modelo treinado em ambiente não rotulado no conjunto de dados.

Oliveira et al. (2018) apresentam algumas arquiteturas de uma rede neural convolucional capaz de melhorar consideravelmente a tarefa de segmentação, por exemplo, a M-Net que maximiza as características de robustez das abordagens de segmentação semântica por meio da fusão multirresolução. Outra rede é a Fast-Net, projetada especificamente para fornecer a menor carga computacional, possibilitando a execução em GPUs. Uma das maiores dificuldades na técnica de segmentação é o processo de rotulação dos dados, já que se trata de um aprendizado supervisionado. Neste sentido, ao utilizar conjuntos de dados gerados por simulador e outros

Figura 2 – Exemplo de Segmentação Semântica de Cenário

Fonte: (CORDTS et al., 2016)

dataset, como Kitti e Apollo, gera o questionamento se a rede mantêm ou não, uma acurácia aceitável, a Figura 2 ilustra o processo de rotulagem dos dados.

O objetivo geral deste trabalho é apresentar possíveis soluções relacionadas ao problema da navegação autônoma por meio das técnicas de aprendizagem de máquina e visão computacional. Nesse contexto, utilizamos as técnicas de segmentação semântica, para elaborar um algoritmo e testá-lo com diferentes conjuntos de dados. Como objetivos específicos, avaliamos e comparamos o comportamento de uma rede neural convolucional em diferentes bases de dados.

1.1 Objetivos

Desenvolver e avaliar um algoritmo de segmentação semântica aplicado à navegação autônoma, empregando técnicas de aprendizagem de máquina e visão computacional, com ênfase na identificação de pistas e áreas navegáveis em diferentes cenários e condições de iluminação.

1.1.1 Objetivos Específicos

- Analisar arquiteturas de controle para robôs móveis; (Revisão Bibliográfica).
- Implementar Técnicas de Transferência de Conhecimento: Explorar o uso de transferência de aprendizado (transfer learning) para adaptar modelos pré-treinados a novos conjuntos de dados específicos, visando reduzir o custo e o tempo necessários para o treinamento de redes neurais profundas.
- Aplicar Técnicas de Aumento de Dados (Data Augmentation): Utilizar técnicas de aumento de dados para enriquecer o conjunto de treinamento, aumentando a robustez e a generalização do modelo treinado, especialmente em condições adversas como neblina, chuva, variações de iluminação e oclusões parciais.

- Validar o Algoritmo em Cenários Reais: Testar e validar o algoritmo desenvolvido em cenários reais de navegação, analisando sua capacidade de identificar corretamente as pistas e áreas navegáveis, bem como sua eficácia em diferentes condições ambientais.

1.2 Justificativa

A evolução tecnológica recente tem impulsionado significativamente o desenvolvimento de sistemas autônomos, especialmente no contexto da navegação autônoma de veículos. Esta área de pesquisa se beneficia grandemente das inovações em visão computacional e aprendizado de máquina, com a segmentação semântica desempenhando um papel crucial na interpretação precisa do ambiente por tais sistemas. A segmentação semântica, que visa compreender e delinear diversas partes de uma imagem em níveis semânticos, é essencial para a identificação de pistas, obstáculos e áreas navegáveis, facilitando a tomada de decisões seguras e eficientes por veículos autônomos.

Este trabalho justifica-se pela necessidade crescente de soluções robustas e eficazes para a navegação autônoma, capazes de operar em diversas condições ambientais e situacionais. A complexidade do ambiente de navegação, marcada por variações de iluminação, presença de obstáculos dinâmicos e condições meteorológicas adversas, exige modelos de segmentação semântica altamente adaptáveis e precisos. Além disso, a construção de datasets abrangentes para treinamento de modelos é um desafio significativo, devido ao custo elevado e à necessidade de uma grande variedade de exemplos anotados. O uso de técnicas como transferência de aprendizado e aumento de dados (data augmentation) surge como uma solução viável para superar essas barreiras, permitindo a adaptação de modelos pré-treinados a novos contextos com um investimento menor de tempo e recursos.

A relevância deste trabalho é reforçada pelo potencial impacto da navegação autônoma na segurança do trânsito, na eficiência do transporte e na mobilidade urbana. Ao avançar o estado da arte em segmentação semântica aplicada à navegação autônoma, este estudo contribui não apenas para o progresso científico na área de visão computacional e aprendizado de máquina, mas também para o desenvolvimento de soluções tecnológicas que poderão transformar a sociedade, promovendo um futuro mais seguro, sustentável e conectado.

1.3 Estrutura do Documento

Além deste capítulo, o trabalho está dividido em:

- No **Capítulo 2** é apresentado a fundamentação teórica necessária para a conclusão do trabalho;

- No **Capítulo 3.1** é descrito Levantamento Bibliográfico usando como base para realização do trabalho;
- No **Capítulo 3.2** discute-se detalhes Implementação;
- No **Capítulo 3.3** apresenta-se o Ambiente de Desenvolvimento;
- No **Capítulo 3.4, 3.5, 3.6** mostrado os resultados obtidos;
- No **Capítulo 3.8** é feita as considerações finais;

2 REFERENCIAL TEÓRICO

2.1 PROCESSAMENTO DIGITAL DE IMAGENS

As primeiras aplicações do processamento de imagens ocorreu no início do século XX, para melhorar a qualidade da impressão de imagens transmitidas pelo sistema *Bartlane*, um cabo submarino que ligava Londres a Nova Iorque. Os sistemas *Bartlane* iniciais da década de 1920 codificava uma imagem em 5 níveis distintos de intensidade, a partir de 1929 começou a ser codificada em 15 níveis. Cerca de três décadas mais tarde começou a impulsionar a área de processamento de imagens com o advento dos primeiros computadores de grande porte e o início do programa espacial norte-americano (FILHO; NETO, 1999).

Para Filho e Neto (1999), o aprimoramento de imagens por meio de sistemas computacionais iniciou-se no *JPL (Jet Propulsion Laboratory)* em Pasadena, Califórnia – EUA, quando as imagens da lua transmitidas pela sonda Ranger eram corrigidas para tirar distorções da câmera acoplada na sonda. Os mesmos autores ainda relatam que o processamento de imagens está presente em quase todos os ramos, por exemplo, na medicina através do diagnóstico realizado por meio dos exames de imagens; biologia, geografia, geologia, astronomia, publicidade e marketing, e muitos outros ramos.

A amostragem significa medir o valor de uma imagem em um número finito de pontos, normalmente corresponde à extensão do número de pixels nas direções vertical e horizontal. A quantização é a representação do valor medido no ponto amostrado por um número inteiro. (FURHT; AKAR; ANDREWS, 2018).

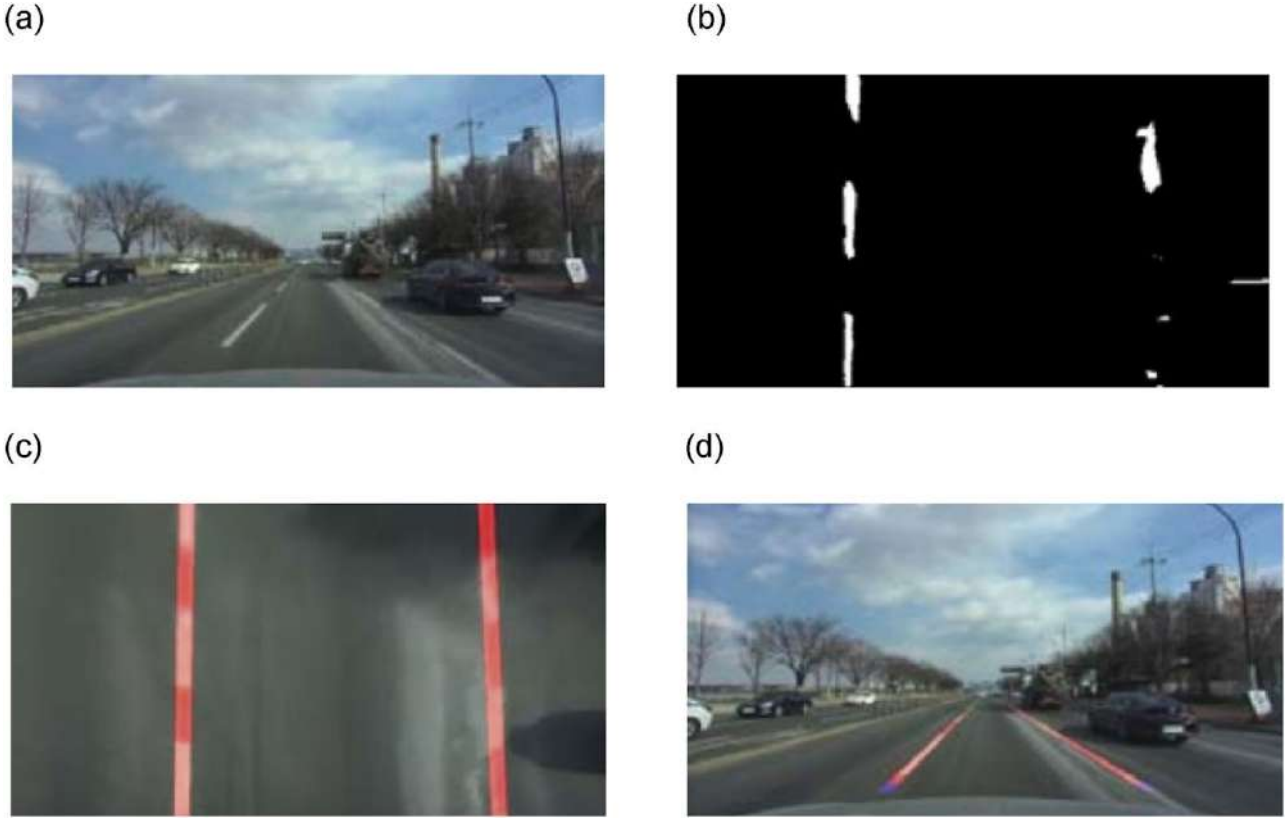
- Segmentação

A segmentação é aplicada para subdividir uma imagem em suas regiões, componentes ou objetos. Esses algoritmos baseiam-se em duas propriedades básicas de valores de intensidade, isto é, a descontinuidade e a similaridade. A descontinuidade identifica mudanças bruscas de intensidade, e a similaridade, relaciona as partições de uma imagem em regiões semelhantes de acordo com um conjunto de critérios predefinidos (SHAWAL; SHOYAB; BEGUM, 2014).

Diversas técnicas podem ser utilizadas para segmentação de imagem, embora o método tradicional utilizando processamento de imagem tenha sido gradualmente substituído pelos métodos de segmentação semântica, como o método de aprendizado profundo (LIU, 2014; SHIRKE; UDAYAKUMAR, 2019). Na Figura 3 mostramos um exemplo de detecção de linha usando apenas processamento de imagem.

Figura 3 – Procedimento de detecção de linha de faixa (CHOI; PARK; JUNG, 2018).

(a) Imagem original (b) Imagem binarizada para detecção de bordas. (c) mudança de perspectiva (d) Resultado da detecção.



Fonte: (CHOI; PARK; JUNG, 2018)

Os algoritmos de processamento de imagem apresentam bons resultados, mas são sensíveis a vários fatores que interfere diretamente no seu desempenho, a imagem (c) por exemplo, mostra a área da pista delimitada por um retângulo, com o intuito de reduzir ruídos do ambiente, no entanto, isso gera outros problemas, como mudança de perspectiva, o que pode fazer com que a pista não esteja enquadrada nesse retângulo, nesse ponto o algoritmo falha.

- Classificação

A classificação de imagens pode ser definida como o processo de redução de uma imagem a classes de informação. Esta etapa é comumente usada na interpretação de fotos e análise quantitativa. Um dos principais objetivos do processamento digital de imagens é interpretar os dados observados e classificar as características para análise.

2.2 REDES NEURAIS

2.2.1 Redes Neurais Artificiais

As Redes Neurais Artificiais *(RNA's)* são inspiradas em um modelo biológico. Nosso cérebro tem uma grande capacidade de processar informações em um curto intervalo de tempo, alta capacidade de aprendizagem e armazenamento de informações, com desenvolvimento que perdura por toda vida. As *(RNA's)* são uma subárea da inteligência artificial inspirada no modelo biológico dos neurônios dos seres vivos (HAYKIN; ENGEL, 2007). Neste sentido, as *(RNA's)* são estruturas capazes de compreender, registrar e generalizar determinadas situações e problemas que são apresentados (BRAGA; PONCE; LUDERMIR, 2007). As *RNA* são sistemas paralelos compostos por unidades elementares, denominadas neurônios, que calculam determinadas funções matemáticas, geralmente não-lineares, cujo funcionamento é inspirado no funcionamento do neurônio biológico.

As soluções apresentadas pelas *RNA* podem se assemelhar ou até superar as apresentadas pela programação tradicional. Uma *RNA* passa por um processo de aprendizado, no qual amostras de entrada e saída são apresentadas às suas unidades elementares, que por si só encontram as características necessárias para representar a informação fornecida. Em seguida, é necessário definir o sistema resultante. Dessa forma, conforme o (GONZALEZ; WOODS, 2002), podemos dizer que redes neurais artificiais são uma forma de abordar a solução de problemas de inteligência artificial. É possível classificar amostras de dados desconhecidas, mas que se assemelham às informações aprendidas durante a etapa de treinamento, utilizando *RNA* Elas podem extrair características que não estão explicitamente apresentadas sob a forma de exemplos.

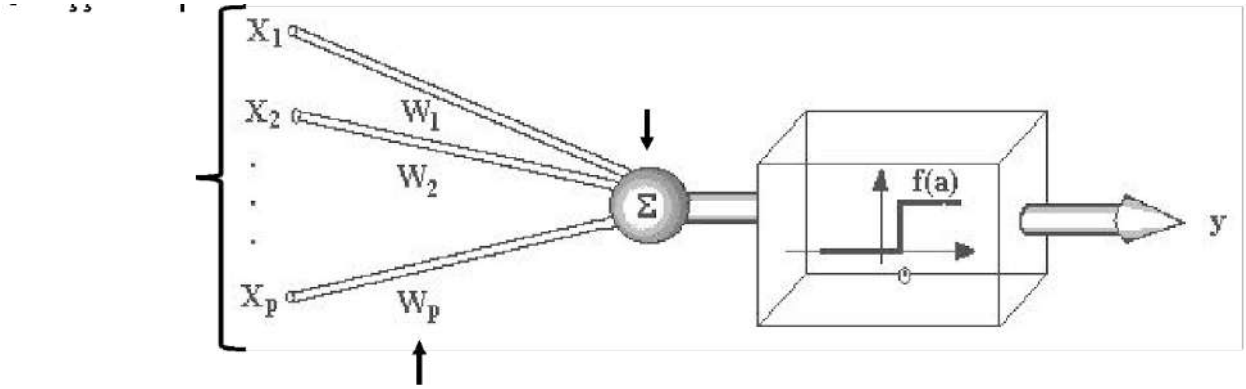
De acordo com (HAYKIN; ENGEL, 2007), uma rede neural é constituída por uma camada de entrada, com um ou mais neurônios, por uma ou mais camada oculta e uma camada de saída. A maioria das redes neurais possuem a estrutura mencionada anteriormente, diferenciando-se apenas pela quantidade de neurônio entre as camadas e pela quantidade de camadas ocultas. A unidade básica de uma *RNA* são os elementos de processamento, também chamados de neurônios, existem diferentes tipos de neurônios artificiais, entre eles o Perceptron e o Adaline, ambos possuem a mesma estrutura, diferenciando-se apenas pelo algoritmo de treinamento.

A operação de uma unidade de processamento, proposta por McCulloch e Pitts (1943) conforme figura 4, pode ser resumida da seguinte maneira:

1. sinais são apresentados à entrada;
2. cada sinal é multiplicado por um número, ou peso, que indica a sua influência na saída da unidade;
3. é feita a soma ponderada dos sinais que produz um nível de atividade;

4. se este nível de atividade exceder um certo limite (*threshold*) a unidade produz uma determinada resposta de saída.

Figura 4 – Esquema de Unidade Processadora de (MCCULLOCH; PITTS, 1943)



Fonte: (HAYKIN, 2001)

Nesta imagem $x_1, x_2, \ldots, x_p$ são as entradas da rede, e $w_1, w_2, \ldots, w_p$ são os pesos associados. Então, para toda entrada x_p é feito o somatório $\sum$ com seu peso w_p correspondente. A etapa seguinte é aplicada uma função de ativação. Além das entradas e dos pesos uma constante chamada "bias" de valor igual a 1 é adicionada ao resultado da multiplicação de x_p por w_p, essa constante tem como finalidade gerar uma excitação no neurônio e por fim temos a saída y.

2.2.2 Deep Learning

As técnicas de aprendizado profundo (do inglês *Deep Learning*) são redes neurais com muitas camadas e parâmetros. A maioria dos métodos de aprendizado de máquina utiliza arquiteturas de rede neural e, desta forma, também é referido como redes neurais profundas. Em resumo, o *Deep Learning* usa uma grande variedade de unidades de processamento linear para extração e transformação de recursos. As camadas inferiores próximas à entrada de dados adquirem habilidades básicas, enquanto as camadas superiores adquirem habilidades mais sofisticadas derivadas de recursos da camada inferior. Isso significa que o *Deep Learning* é adequado para analisar e extrair conhecimento útil de abundância de dados, a partir de dados coletados de diferentes fontes (ZHANG; WANG; LIU, 2018).

As principais razões para a popularidade do *Deep Learning* hoje são, habilidades de processamento de chip drasticamente aumentadas (por exemplo, unidades de GPU), o baixo custo significativo de hardware de computação e avanços recentes em *Machine Learning* e pesquisa de processamento de sinal/informação (SHINDE; SHAH, 2018). Desta forma, os domínios de aplicação para o *Deep Learning* estão relacionados à visão computacional, previsão, análise

semântica, processamento de linguagem natural, recuperação de informações e gerenciamento de relacionamento com o cliente.

2.2.3 Rede Neural Convolucional

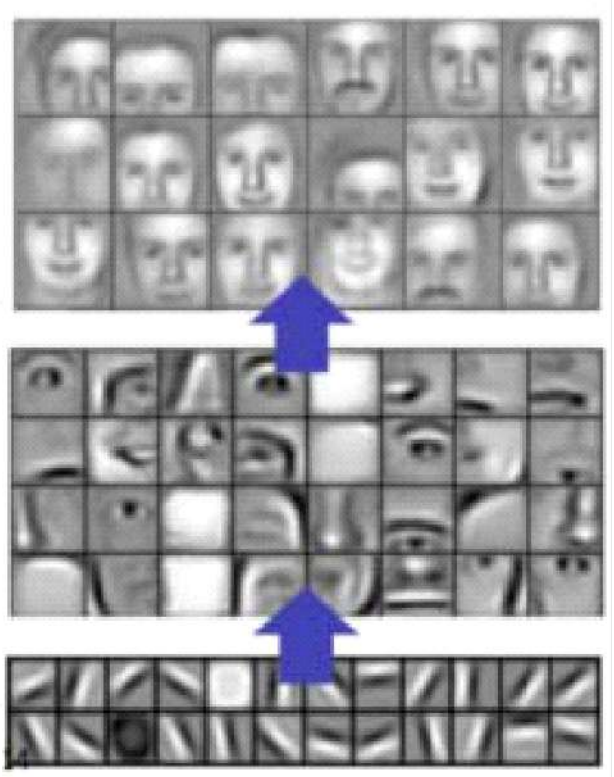
A Rede Neural Convolucional *(CNN, do inglês Convolucional Neural Networks)* teve resultados inovadores na última década em uma variedade de campos relacionados ao reconhecimento de padrões, desde o processamento de imagem ao reconhecimento de voz. O aspecto mais benéfico das *CNN* é reduzir o número de parâmetros na das *RNA* a partir de operações convolucionais. Essa conquista levou pesquisadores e desenvolvedores a abordar modelos maiores para resolver tarefas complexas, o que não era possível com *RNAs* clássicas (SU; LIU; WANG, 2016). A suposição mais importante sobre problemas resolvidos pelas *CNN* não deve ter características que sejam espacialmente dependentes. Em outras palavras, por exemplo, em um aplicativo de detecção de rostos, não precisamos prestar atenção em onde os rostos estão localizados nas imagens. A única preocupação é detectá-los independentemente de sua posição nas imagens fornecidas. Outro aspecto importante da *CNN* é obter recursos abstratos quando a entrada se propaga em direção às camadas mais profundas (ALBAWI; MOHAMMED; AL-ZAWI, 2017). Por exemplo, na classificação de imagens, a borda pode ser detectada nas primeiras camadas e, em seguida, as formas mais simples nas segundas camadas e, em seguida, os recursos de nível superior, como faces nas próximas camadas.

As *CNNs* estão organizadas em sucessivas camadas computacionais, alternando entre convolução e agrupamento. Em comparação com outros tipos de redes neurais profundas, elas são relativamente fáceis de treinar com retro propagação *(backpropagation)*, principalmente porque possuem uma conectividade muito esparsa em cada camada convolucional (BENGIO, 2009). Em uma camada convolucional, filtros lineares são usados para convolução. Para reduzir o número de parâmetros, é adotada uma estratégia de compartilhamento de parâmetros. Embora o compartilhamento de parâmetros reduza a capacidade das redes, ele melhora sua capacidade de generalização. As camadas computacionais, ou seja, camadas convolucionais, podem ser aprimoradas substituindo o filtro linear com uma função não linear: perceptron multicamada raso *(PMR)* (LIN; CHEN; YAN, 2014). A *CNN* com *PMR* superficial é chamada de rede em rede *(NiN)*. Com unidades ocultas suficientes, o *PMR* pode representar funções arbitrárias complexas, mas suaves e, portanto, pode melhorar a separabilidade dos recursos extraídos. Assim, o *(NiN)* consegue fornecer um erro de reconhecimento menor do que o *RNC* clássico.

Em geral, as *CNNs* são organizadas principalmente em camadas entrelaçadas de dois tipos: camadas convolucionais e camadas de agrupamento (subamostragem) com uma camada convolucional ou várias camadas convolucionais seguidas por uma camada de agrupamento (CHANG, 2015). O papel das camadas convolucionais é a representação de recursos com o nível semântico dos recursos, aumentando conforme a profundidade das camadas. Cada camada convolucional consiste em vários mapas de características, também conhecidos como canais.

Cada mapa de características é obtido deslizando *(i.e, convolução)* um filtro sobre os canais de entrada com passo predefinido, seguido de uma ativação não linear conforme a figura 5.

Figura 5 – Recursos aprendidos de uma rede neural convolucional.



Fonte: (ALBAWI; MOHAMMED; AL-ZAWI, 2017)

Diferentes mapas de recursos correspondem a diferentes parâmetros de filtros que possuem um mapa de recursos que compartilha os mesmos parâmetros. Os filtros podem ser aprendidos com algoritmo de retro propagação *(i.e., backpropagation)*. *Pooling* é um processo que substitui a saída de suas camadas convolucionais correspondentes em determinado local, com estatísticas resumidas das saídas próximas (ZHANG et al., 2016). O agrupamento sobre regiões espaciais contribui para que a representação de características se torne invariante de translação e rotação e, também, contribui para melhorar a eficiência computacional da rede. As camadas após a última camada de *Pooling* geralmente são totalmente conectadas, destinadas à classificação. O número de camadas é chamado de profundidade da rede e o número de unidades de cada camada é chamado de largura da rede. O número de mapas de recursos (canais) em cada camada também pode representar a largura (amplitude) das *CNNs* e, a profundidade e a largura determinam a sua capacidade. De modo geral, existem seis direções para melhorar o desempenho das *CNNs* com algumas delas sobrepostas (LEE; GALLAGHER; TU, 2015):

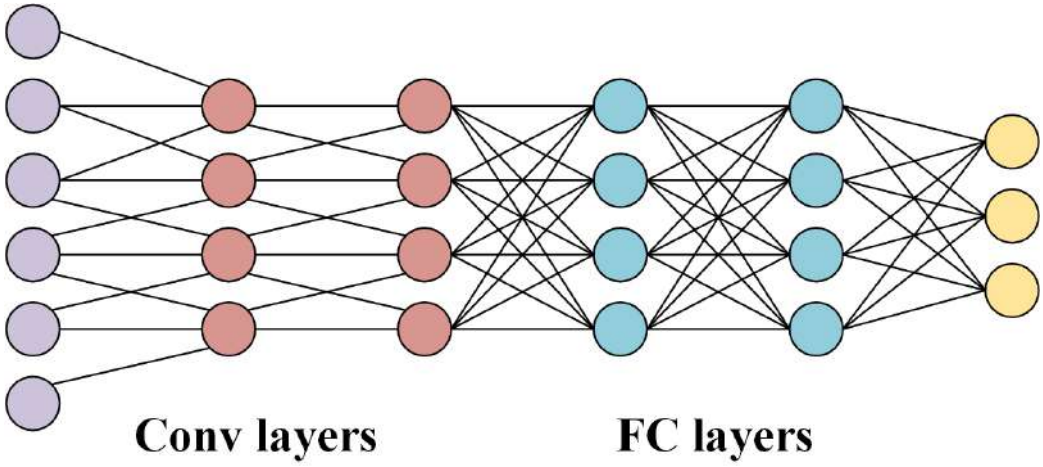
1. aumentar a profundidade;
2. aumentar a largura;

3. modificar a operação de convolução;
4. modificar a operação de agrupamento;
5. redução do número de parâmetros;
6. modificar a função de ativação.

Comparado com as redes totalmente conectadas (FC) na figura 8, a *CNNs* possui muitas vantagens:

1. conexões locais: cada neurônio não está mais conectado a todos os neurônios da camada anterior, mas apenas a um pequeno número de neurônios, sendo eficaz em reduzindo parâmetros e acelerando a convergência;
2. Compartilhamento de peso: um grupo de conexões pode compartilhar os mesmos pesos, o que reduz ainda mais os parâmetros;
3. Redução da dimensão do *Downsampling*: uma camada de *pooling* aproveita o princípio da correlação local da imagem para reduzir o tamanho de uma imagem, o que pode reduzir a quantidade de dados enquanto retém informações úteis. Ele também pode reduzir o número de parâmetros removendo recursos triviais. Essas três características atraentes fazem da *RNC* um dos algoritmos mais representativos no campo de *Deep Learning*, a figura 6 mostra o diagrama de camadas *RNC*.

Figura 6 – Diagrama de camadas *RNC* e camadas *FC*.



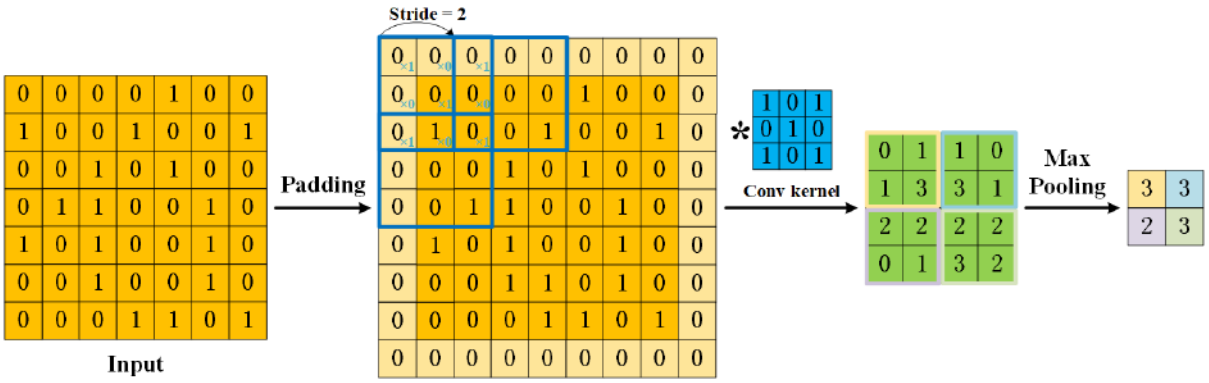
Fonte: (LI et al., 2021a)

Para ser específico, para construir um modelo *CNN*, normalmente são necessários quatro componentes. A convolução é uma etapa fundamental para a extração de recursos. As saídas da convolução podem ser chamadas de mapas de recursos. Ao definir um kernel de convolução

com um determinado tamanho, perderemos informações na borda. Assim, o preenchimento é introduzido para ampliar a entrada com valor zero, que pode ajustar o tamanho indiretamente. Além disso, quanto maior o passo, menor é a densidade da convolução. Após a convolução, os mapas de recursos consistem em muitos recursos que são propensos a causar problemas de *over-fitting* (HAWKINS, 2004). Como resultado, o *pooling* (também conhecido como downsampling) é proposto para evitar a redundância, incluindo o *pooling* máximo e o *pooling* médio.

O procedimento de uma CNN é mostrado na Figura 7. Além disso, para que os kernels de convolução percebam áreas maiores, a convolução dilatada deve ser proposta(YU; KOLTUN, 2015).

Figura 7 – Procedimento de uma RNC 2-D.



Fonte: (YU; KOLTUN, 2015)

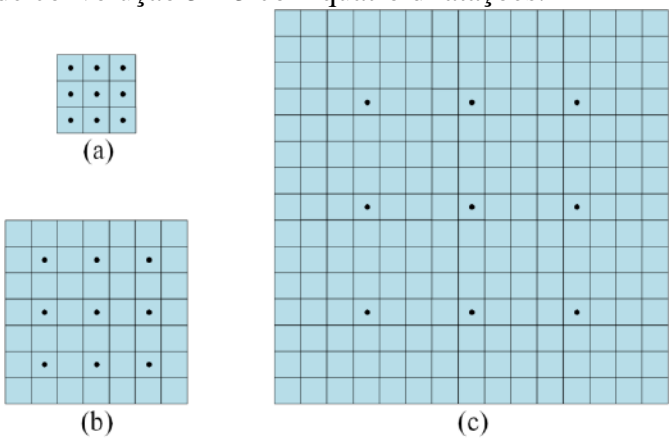
Um kernel de convolução 3 x 3 geral é mostrado na Figura 8(a), e um kernel de convolução 3 x 3 com duas dilatações e um kernel de convolução 3 x 3 com quatro dilatações são mostrados na Figura 8(b) e c. Observe que há um valor vazio (preenchendo com zero) entre cada ponto do kernel de convolução. Mesmo que os pontos kernel válidos ainda sejam 3 x 3, uma convolução com duas dilatações tem um campo receptivo de 7 x 7 e uma convolução com quatro dilatações tem um campo receptivo de 15 x 15.

A capacidade das redes neurais de lidar com variações geométricas é fundamental em visão computacional, mas é limitada por suas estruturas estáticas. Soluções comuns incluem a adição de dados transformados ou técnicas de aumento de dados, que são eficazes, mas demandam muitos recursos. Alternativamente, o uso de algoritmos adaptativos e a seleção de características robustas podem oferecer resistência a essas transformações sem a necessidade de extensos conjuntos de dados, embora possam afetar a generalização do modelo (MODELO; EXEMPLO, 2024).

Dai et al. (2017), introduziram dois módulos que podem melhorar a capacidade da CNN para a transformação geométrica. O primeiro é o módulo de convolução deformável. Conforme mostrado na Figura 9, a convolução deformável foi proposta para lidar com o cenário onde as

Figura 8 – Comparação entre kernel de convolução geral e kernel de convolução dilatado.

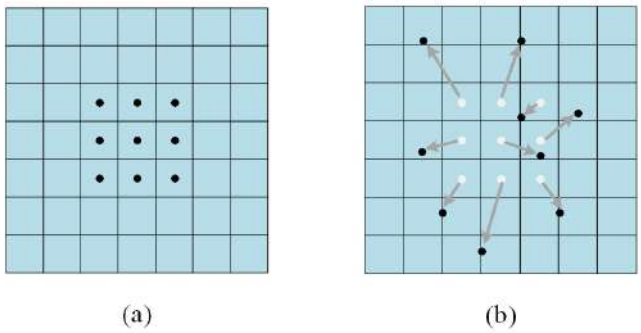
(a) Núcleo de convolução geral 3 x 3.
(b) Núcleo de convolução 3 x 3 com duas dilatações.
(c) Núcleo de convolução 3 x 3 com quatro dilatações.



Fonte: (LI et al., 2021b)

formas dos objetos são geralmente irregulares. A convolução deformável só pode se concentrar no que lhes interessa, tornando os mapas de recursos representativos.

Figura 9 – Comparação entre kernel de convolução geral e kernel de convolução deformável. (a) Núcleo de convolução geral 3 x 3. (b) Núcleo de convolução deformável 3 x 3.



Fonte: (DAI et al., 2017)

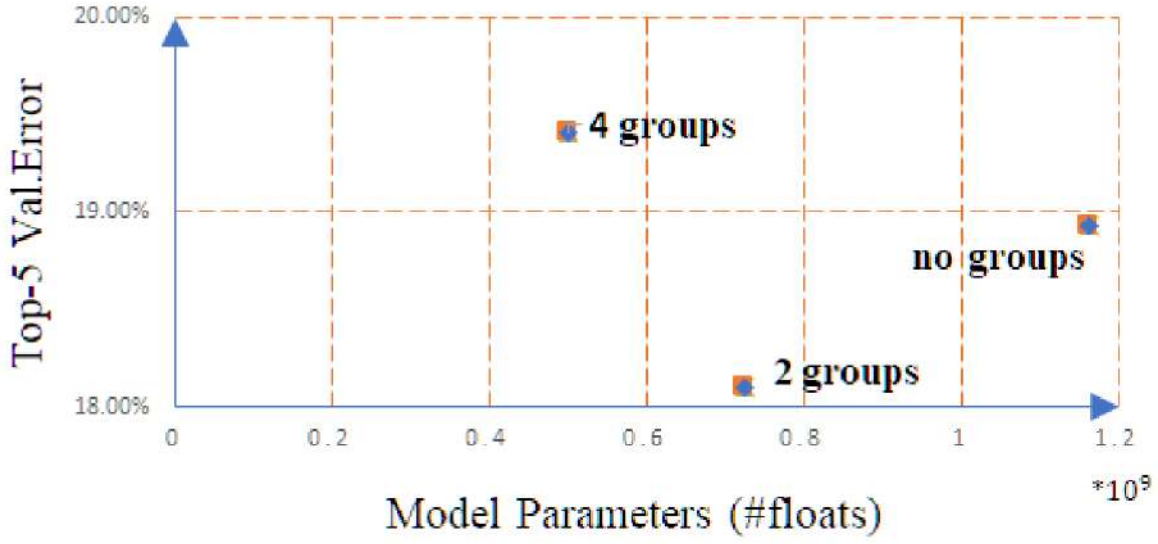
Dai et al. (2017), usaram uma camada de convolução paralela para aprender os deslocamentos e a adicionaram à posição original do *kernel* de convolução para realizar a transformação de escala, como translação e rotação, e então realizou a convolução. Esses deslocamentos se adaptam à transformação de escala do objeto de destino e aumentam o campo receptivo do *kernel* de convolução. O segundo é um módulo de agrupamento de região de interesse (ROI) deformável. O módulo de *pooling* deformável usa uma camada de conexão completa para aprender os deslocamentos dos recursos e, em seguida, realizar o *pooling*.

Zhu et al. (2019), descobriram que, embora a convolução deformável possa se adaptar à transformação geométrica e expandir o campo receptivo movendo a unidade do kernel de

convolução, o treinamento é afetado por muitas características que não estão relacionadas ao domínio disponível. Portanto, é necessário aumentar o volume deformável. Além disso, os autores propuseram um *convNet v2* deformável. Ao expandir o uso da convolução deformável na rede e adicionar um mecanismo de modulação à convolução deformável, o módulo de rede pode variar tanto a distribuição espacial quanto a influência relativa de suas amostras. Para atender aos requisitos de *hardware*, Dong et al. (2021), combinou convolução profunda com convolução deformável. Todas as camadas de convolução para prever os deslocamentos são substituídas por convolução em profundidade para reduzir ainda mais o custo computacional.

A convolução de grupo foi aplicada pela primeira vez na arquitetura *AlexNet* para treinar *CNNs* profundas com baixo consumo de recursos de GPU. Substituir a convolução comum pela convolução em grupo pode construir redes mais amplas e reduzir os parâmetros. Além disso, a convolução em grupo pode aprender melhor as representações e garantir a precisão. Conforme mostrado na Figura 10, o *AlexNet* sem convolução de grupo é menos eficiente e menos preciso do que os outros dois grupos (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

Figura 10 – Comparação do *AlexNet* com convolução de grupo e sem convolução de grupo.



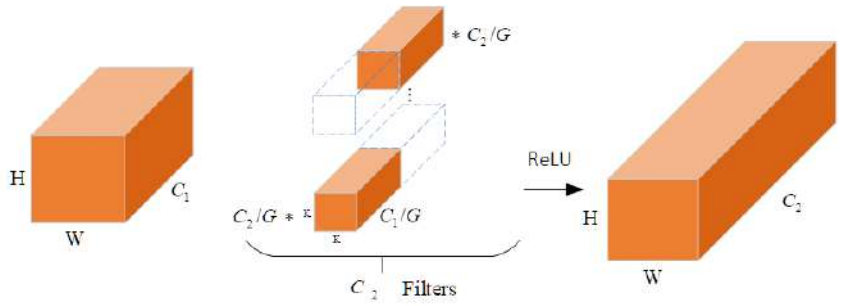
Fonte: (KRIZHEVSKY; SUTSKEVER; HINTON, 2012)

A convolução em grupo é a melhor forma para aprender recursos. Como visto na Figura 11, o banco de filtros de convolução de grupo aprende características em uma estrutura de bloco diagonal esparsa na dimensão do canal. Além disso, a operação análoga à regularização é usada para aprender redes profundas mais precisas e úteis. Como a convolução em grupo é mais eficaz que o padrão, o *ResNeXt* sobrepõe blocos de convolução menores para aproximar os blocos de convolução do *ResNet*, o que pode aumentar a precisão e reduzir a complexidade do modelo (XIE et al., 2017).

Para minimizar a carga de projeto, os blocos convolucionais menores usados no *ResNeXt* têm a mesma topologia. Os resultados experimentais do *ResNeXt* também provaram que aumen-

tar o número de blocos de empilhamento é mais eficaz do que apenas aumentar a largura e a profundidade da rede. (HUANG et al., 2018), dividiram o treinamento de convolução do grupo em várias etapas. A primeira etapa é a fase de compressão. Após o treinamento de regularização indutor de esparsidade, os filtros menos essenciais com pesos menores são removidos. A segunda etapa é a fase de otimização. Depois que os pesos de cada grupo são fixados, o treinamento contínuo garante que a convolução do mesmo grupo compartilhe o mesmo padrão de esparsidade. No entanto, este método de selecionar manualmente a convolução do grupo ainda carece de investigação. Assim, (ZHANG et al., 2019), propuseram *ResNeXt Groupable ConvNet* para combinar pesquisa de arquitetura de rede com convolução de grupo, que pode aprender automaticamente o número de grupos usando uma abordagem de ponta a ponta.

Figura 11 – Módulo de Convolução de Grupo. Uma camada de convolução é dividida em G grupos de pequenos filtros, e a saída da convolução de pacotes é composta pela produção de todo o mapa de características.



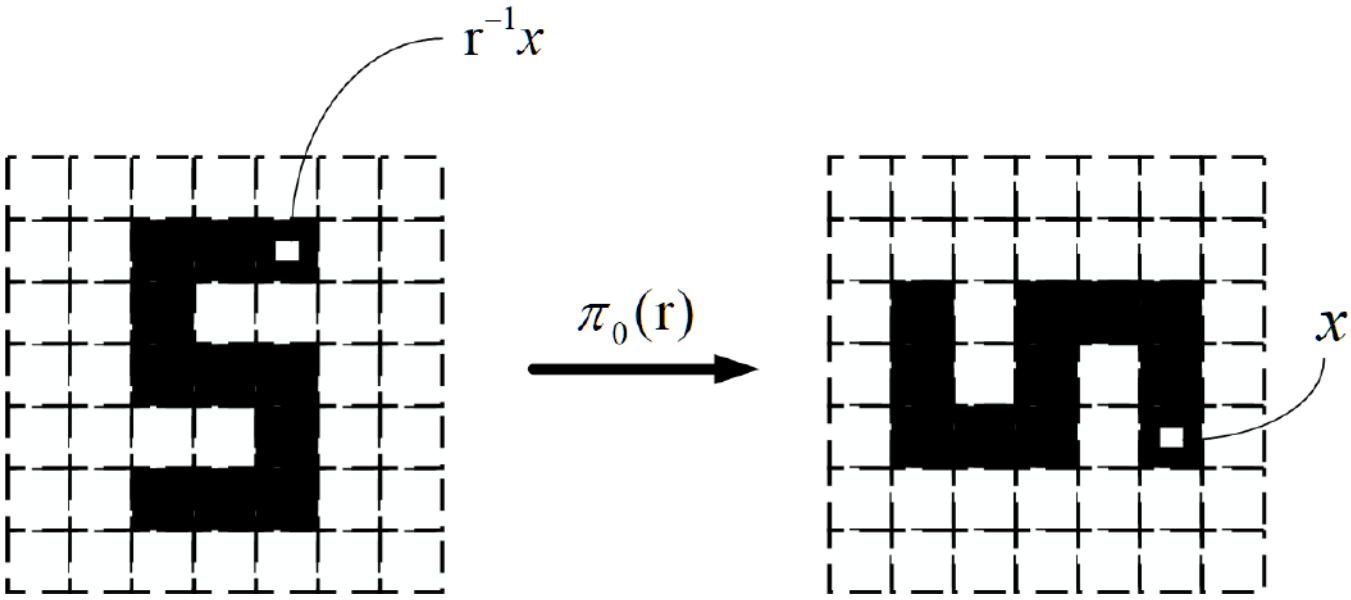
Fonte: (XIE et al., 2017)

Conforme mostrado na Figura 12, após girar uma imagem pertencente ao espaço linear, a rede invariante não realiza um reconhecimento mais preciso (LI et al., 2021a). Especificamente, um rosto com características faciais anormais, por exemplo, em uma pintura abstrata, ainda pode ser reconhecido como uma amostra positiva em uma rede de convolução invariável. Portanto, precisamos que as *RNC's* tenham equivariância para produzir uma representação linear previsível na transformação de entrada. Os filtros se adaptam não apenas às mudanças de posição como uma *RNC* padrão, mas também às mudanças de pose.

Cohen e Welling (2016), avançaram uma teoria geral de representação manipulável e propuseram *RNC* orientável que aplicasse a teoria a operações de convolução. Conforme mostrado na Figura 13, eles introduziram várias formas de transformação linear em um grupo. Isso ajuda a aumentar a flexibilidade da *RNC* equivalente e dissociar o tamanho dos grupos e a complexidade computacional. Além disso, a *RNC* orientável pode ser facilmente estendida para configurações contínuas, e a avaliação da *RNC* orientável para grupos grandes e de alta dimensão é uma parte essencial do trabalho futuro.

Com base na *RNC* orientável, Weiler et al. (2018) propuseram a *RNC* orientável 3-D que é equivalente ao movimento de corpo rígido, representando dados em espaço euclidiano

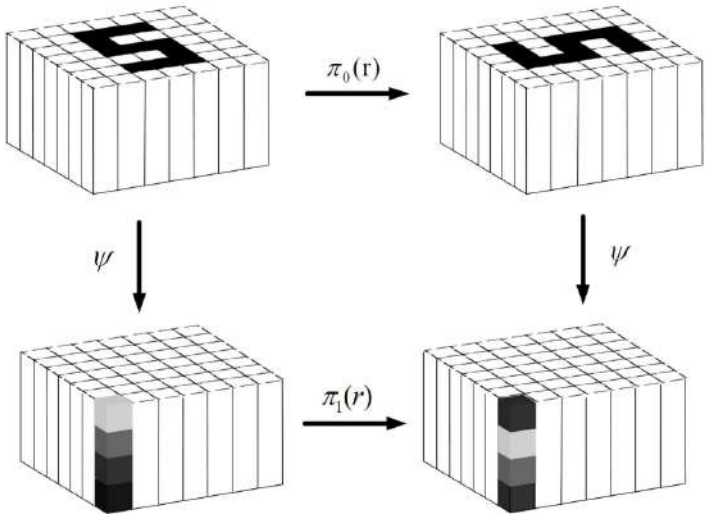
Figura 12 – A imagem usa o operador linear $R(r)$ para girar r.



Fonte: (LI et al., 2021a)

3-D usando campos escalares, vetoriais e tensoriais e mapeando essas representações usando convolução de variantes iguais. Além disso, o grupo euclidiano E(2) e seus subgrupos como soluções gerais do espaço kernel são dados para se adaptar à rotação e reflexão de imagens planares (WEILER; CESA, 2021).

Figura 13 – Ψ é a transformação com equivariância. Transforma uma lista de recursos em uma nova posição através de 1 e, em seguida, usa uma matriz de permutação equivalente a uma rotação de 90° para trocar ciclicamente quatro canais.



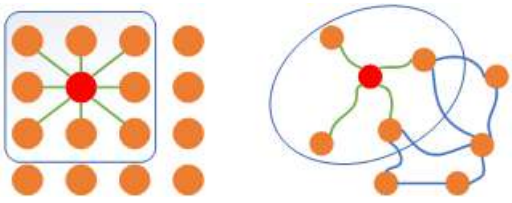
Fonte: (COHEN; WELLING, 2016)

A Rede Neural Gráfica *(RNG)* é um dos tópicos populares nos últimos anos. A primeira

RNG baseada na teoria do ponto fixo proposta por (GORI; MONFARDINI; SCARSELLI, 2005), é usado para resolver problemas. Eles converteram o gráfico em uma matriz *1-D* no estágio de pré-processamento, levando à perda de informações topológicas. (SCARSELLI et al., 2009), estendeu o método de rede neural existente para processar os dados do domínio do gráfico. Eles propuseram um modelo *RNG* que pode processar diretamente mais tipos de gráficos, como gráficos cíclicos, gráficos direcionados e gráficos não direcionados. Como as redes neurais de convolução não podem aprender dados em espaço não euclidiano, o modelo *RNG* anterior não pode ser combinado com a convolução. Torna-se um entrave que as *RNG's* precisam superar. Bruna et al. (2014), descobriram que a *RNG* pode usar a invariância de tradução local da classe de sinal em seu domínio, o que é muito útil em tarefas de reconhecimento de imagem e áudio. Eles propuseram dois tipos de estruturas *RNG*. Um é baseado em um agrupamento hierárquico do domínio e o outro é baseado no espectro do grafo Laplaciano. Kipf e Welling (2017), propuseram uma rede de convolução de grafos *(RNG)*. Eles motivam a escolha da arquitetura de convolução por meio de uma aproximação de primeira ordem localizada de convoluções de grafos espectrais. O modelo *RNG* expande linearmente o número de arestas do gráfico e aprende a representação da camada oculta da estrutura do gráfico local codificado e dos recursos do nó. Os métodos de convolução do grafo podem ser divididos em dois tipos: convolução no domínio espacial e convolução no domínio da frequência.

Em comparação com a convolução no domínio espacial na Figura 14, a convolução no domínio da frequência tem uma base teórica factual para o processamento do sinal gráfico. *RNG* é um método baseado na convolução no domínio da frequência, que filtra uma rede de convolução e mapeia os sinais processados pela transformada de *Fourier* para o domínio da frequência (KIPF; WELLING, 2017). Do ponto de vista do domínio espacial, o *RNG* é considerado agregando informações de recursos da vizinhança de um nó e gradualmente aprimorado ao encontrar matrizes simétricas alternativas.

Figura 14 – Comparação entre convolução padrão e convolução de grafos no domínio espacial.



Fonte: (KIPF; WELLING, 2017)

RGC adaptativo proposto por Li et al. (2018), constrói um gráfico residual usando uma função de distância aprendida com recursos de dois nós como entrada. A matriz de adjacência do gráfico residual pode aprender relações estruturais ocultas não especificadas. Ao integrar saídas de camadas de convolução de gráfico duplo, o *(DRGC)* captura informações estruturais locais e globais sem alavancar várias camadas de convolução de gráfico. A *(RCN)* tradicional faz a convolução dos píxeis na imagem, enquanto as convoluções de grafos baseadas no domínio

espacial definem a convolução pelas relações espaciais dos nós nos grafos. Ele envolve a representação do nó central com a dos nós vizinhos e obtém a representação atualizada do nó central. A informação do nó da convolução do domínio espacial propaga-se essencialmente ao longo das arestas. Micheli (2009), propôs a primeira rede neural de convolução de grafos baseada no domínio espacial, denominada rede neural para grafos *(RN4G)*. Ele realiza convoluções acumulando as informações de um nó vizinho diretamente. Para registrar as características de cada camada, o *(RN4G)* usa estrutura residual e conexões de salto.

2.3 SEGMENTAÇÃO SEMÂNTICA

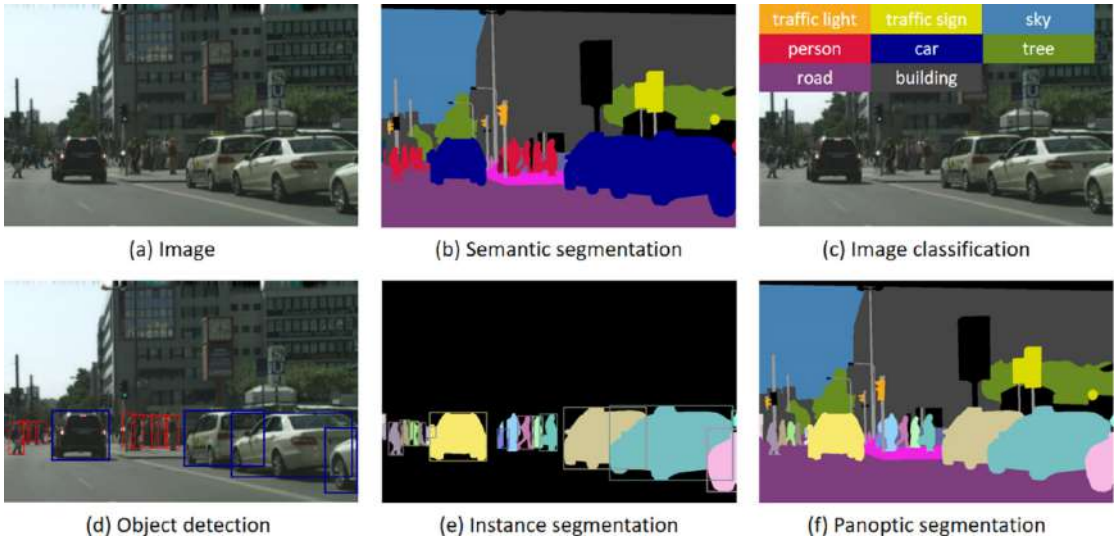
Os trabalhos recentes em *Deep Learning* lidando com segmentação semântica foram significativamente aprimorados com o uso de redes neurais. As redes neutras fizeram um grande progresso devido à abundância de dados disponíveis, graças ao surgimento de câmeras digitais, câmeras de telefones celulares e o poder da computação, que está ficando mais rápido à medida que as *GPUs* se tornam ferramentas de computação de uso geral (COX; DEAN, 2014).

O surgimento da terminologia "segmentação semântica" pode ser datado da década de 1970 (OHTA; KANADE; SAKAI, 1978). Naquela época, essa terminologia era equivalente à segmentação de imagens, mas enfatizava que as regiões segmentadas deveriam ser "semanticamente significativas". Na década de 1990, a "segmentação e reconhecimento de objetos" distinguiu ainda mais objetos semânticos de todas as classes do plano de fundo e, pode ser visto, como um problema de segmentação de imagem de duas classes (EDELMAN; POGGIO, 1989). Como a partição completa de objetos em primeiro plano do fundo é muito desafiadora, um problema de segmentação de imagem de duas classes relaxado: a detecção de objetos de janela deslizante, foi proposto para particionar objetos com caixas delimitadoras (VIOLA; JONES, 2001). Encontrar onde estão os objetos nas cenas, com excelentes algoritmos de segmentação de imagem de duas classes, é fundamental. No entanto, a segmentação de imagem de duas classes não pode dizer quais são esses objetos segmentados. Como resultado, o sentido genérico de reconhecimento de objetos (ou detecção) foi gradualmente estendido para rotulagem de imagens multi-classes (CARBONETTO; FREITAS; BARNARD, 2004), ou seja, segmentação semântica no sentido atual, para dizer onde e quais os objetos na cena.

A segmentação semântica desempenha um papel importante na compreensão da imagem e essencial para tarefas de análise de imagem, atribuindo um rótulo a cada píxel, também conhecido como classificação ao nível de píxel. Desta forma, muitas aplicações reais se beneficiam dessa tarefa, através da visão computacional e inteligência artificial, por exemplo, a condução autônoma; navegação robótica; inspeção industrial; sensoriamento remoto; em ciências cognitivas e computacionais com a detecção de objetos de saliência; em ciências da agricultura; na moda com a categorização de artigos de vestuário; em ciências médicas com a análise de imagens médicas; e, etc. (JI et al., 2019).

As primeiras abordagens usadas para segmentação semântica eram Texton Forests (MULLANI; DANDAVATE, 2019), enquanto as técnicas de aprendizado profundo permitiam uma segmentação precisa e muito mais rápida (LATEEF; RUICHEK, 2019a) A segmentação semântica requer classificação de imagens, detecção de objetos e localização de limites. A figura 15 é um exemplo de detecção de objetos, envolvendo caixa delimitadora e classificação de cada píxel em diferentes classes (carro, estrada, céu, vegetação, terreno, etc.). As informações semânticas ao nível de píxel ajudam os sistemas inteligentes a compreender posições espaciais ou fazer julgamentos importantes. Nesse contexto, a segmentação semântica se distingue de outras tarefas comuns de visão computacional. Por exemplo, a classificação de objetos requer que uma imagem inteira seja anotada com um ou mais rótulos semânticos. Em relação à detecção de objetos, o sistema precisa saber onde os objetos alvos estão localizados na cena (LATEEF; RUICHEK, 2019a).

Figura 15 – Um exemplo de diferentes tarefas de visão.



Fonte: (KIRILLOV et al., 2019)

Inúmeros métodos de segmentação foram propostos antes da era do *Deep Learning*, como os métodos baseados em equações diferenciais parciais. Com dados de treinamento suficientes, a estratégia de aprendizado supervisionado consegue estender muito a capacidade de um modelo de segmentação, como a *Random forest* e a gramática visual aplicada na compreensão de cenas naturais (GAO et al., 2016). O surgimento da técnica de *Deep Learning* tem promovido muito a pesquisa de segmentação semântica. Por exemplo, Long, Shelhamer e Darrell (2015a) propuseram a pioneira Rede Totalmente Convolucional (RTC), que aumentou drasticamente a precisão da segmentação. A RTC abriu o caminho para a segmentação semântica baseada em *Deep Learning*. Até o momento, vários novos métodos baseados em aprendizado profundo foram propostos, baseados em diferentes roteiros técnicos e direcionados a diferentes aplicações. Em comparação com os métodos tradicionais, os métodos baseados em *Deep Learning* mostraram uma melhoria notável na eficácia. Quase todos os desempenhos de última geração de conjuntos

de dados públicos foram alcançados por métodos de *Deep Learning* (WADHWA et al., 2018).

Gunde e Shirgave (2018) introduziram principalmente os métodos tradicionais de segmentação semântica baseados em aprendizado, como os métodos baseados na máquina de vetores de suporte e árvore de decisão. Em Zhao et al. (2017), os autores notaram o surgimento de métodos de segmentação semântica baseados em *Deep Learning*, como abordagens baseadas em proposta de região em RTC. Geng, Zhou e Cao (2018), concentraram-se no desafio de segmentação semântica PASCAL VOC 2012 e analisaram os métodos relacionados, bem como seus resultados. Guo et al. (2018), introduziram de forma abrangente os métodos baseados em proposta de região e RTC, bem como os métodos baseados em supervisão fraca. Lateef e Ruichek (2019b), forneceram uma introdução mais abrangente a este campo, incluindo as estruturas de rede, os conjuntos de dados e métricas comumente usados, os métodos de última geração e algumas possíveis direções futuras.

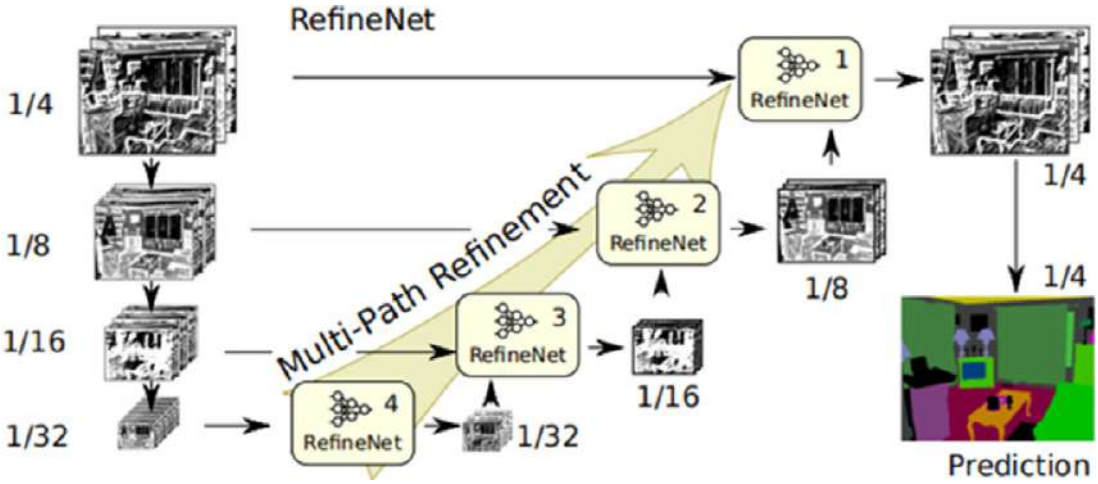
A Rede Totalmente Convolucional (RTC) (LONG; SHELHAMER; DARRELL, 2015a), é pioneira no aprimoramento de recursos por meio da estratégia de conexão de salto. Ao aplicar a conexão de salto entre os recursos de previsão e os recursos da camada intermediária, a resolução final é aumentada de 1/32 para 1/8 e a precisão é aprimorada em aproximadamente 3% em MIoU. Isso prova que alavancar os recursos de camadas rasas e profundas é benéfico para melhorar a precisão da segmentação semântica. Drozdzal et al. (2016), demonstra ainda a importância da conexão de salto na segmentação da imagem.

Ronneberger, Fischer e Brox (2015), propuseram a arquitetura de encoder-decoder simétrica chamada U-Net. Diferente do RTC, a U-Net aproveita totalmente os recursos de cada camada usando conexões de salto densas. Os recursos de cada camada na parte do encoder são conectados às camadas simétricas na parte do decoder. A U-Net atraiu muita atenção da comunidade de análise de imagens médicas (ZHOU et al., 2018). Por exemplo, para resolver a limitação de ignorar a informação espacial ao longo da dimensão z em um modelo baseado em 2D, You et al. (2018) estendem operando 2D-DenseUNet e 3D-DenseUNet de forma cooperativa. Este método visa apreender as características intracorte e intercorte, respectivamente, e fundi-las através do bloco de fusão.

O U-Net também foi estendido e aplicado em outras aplicações. Para a tarefa de segmentação de imagem natural, estende em U-Nets empilhadas. Hariharan et al. (2015), propuseram a representação em hipercoluna, que usa os recursos concatenados de diferentes camadas no pipeline da RCN para fazer a inferência final. Recentemente, o RefineNet (LIN et al., 2017), foi proposto para alavancar ainda mais recursos complementares, no qual um processo de refinamento de múltiplos caminhos é construído, e os detalhes espaciais dos mapas de recursos são aprimorados gradativamente, conforme mostrado na Figura 16.

No entanto, o RefineNet é relativamente caro computacionalmente, como observado em (NEKRASOV; SHEN; REID, 2018). Para extrair características semânticas de alto nível, mantendo detalhes espaciais, Pohlen et al. (2017), separa toda a rede em dois sub-fluxos, ou seja,

Figura 16 – Arquitetura do RefineNet.



Fonte: (LIN et al., 2017)

o fluxo de pooling e o fluxo residual. O fluxo de pooling visa extrair semântica de alto nível (baixa resolução). O fluxo residual visa manter os detalhes (alta resolução) e cooperar com os recursos aprendidos pelo fluxo de pool. Zhang, Wang e Liu (2018). propuseram a estratégia de aprimoramento bilateral, ou seja, um processo de aprimoramento mútuo entre características de baixo e alto nível. Podemos observar que os métodos baseados em aprimoramentos de recursos abrangem principalmente o projeto de uma rede para conectar e fundir diferentes tipos de recursos.

Na segmentação semântica, vários métodos têm alcançado resultados promissores usando redes neurais profundas. Em geral, ao alimentar imagens suficientes e seus mapas de rotulagem píxel a píxel como dados de treinamento, uma rede neural profunda é treinada para aprender um mapeamento entre um rótulo semântico e suas aparências visuais diversificadas. O processo de aprendizagem preenche gradualmente a inconsistência entre a semântica de alto nível e os recursos de baixo nível, tornando a rede cada vez mais consciente de vários conceitos semânticos (VERGARI; MAURO; ESPOSITO, 2018).

A rede VGG foi proposta por Simonyan e Zisserman (2014), do Grupo de Geometria Visual da Universidade de Oxford. Existem diferentes versões para redes VGG conforme o número da camada, como VGG-13, VGG-16 e VGG-19. A principal contribuição das redes VGG é que elas abriram o caminho para projetar estruturas mais profundas para melhor desempenho. O VGG tem sido adotado como a espinha dorsal de vários modelos de segmentação semântica (BADRINARAYANAN; KENDALL; CIPOLLA, 2017). ReNet (VISIN et al., 2015) se destaca pela maneira como substituiu as camadas convolucionais por redes neurais recorrentes multidirecionais, fornecendo uma forma alternativa de construir a arquitetura da rede. O método representativo de segmentação semântica baseado em ReNet é o ReSeg.

A rede residual profunda (ResNet) (HE et al., 2016), permite com sucesso uma rede

muito mais profunda e alcança melhor desempenho em várias tarefas de visão. Sua principal contribuição está na modelagem da representação residual na estrutura da rede RCN, resolvendo a dificuldade de treinar uma estrutura de rede muito profunda. Diferente da estratégia tradicional que torna uma rede mais profunda ou mais ampla, o DenseNet (HUANG et al., 2017) conecta todas as camadas entre si. Sua vantagem está nos seguintes aspectos: 1) menos parâmetros; 2) mais reutilização de recursos e; 3) um melhor processo de treinamento que alivia o problema do gradiente de fuga e degeneração do modelo.

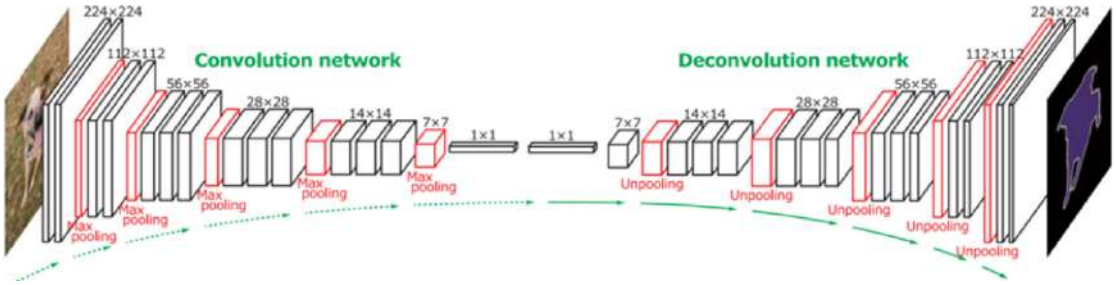
Visando melhorar o desempenho da rede preservando a complexidade da rede, o ResNeXt (XIE et al., 2017) se destaca em sua arquitetura homogênea, multi-ramificações, que possui apenas alguns hiper parâmetros a serem configurados. Os métodos de segmentação semântica DShortcut e ExFuseNet utilizam ResNeXt como sua backbone. Entretanto, é importante projetar redes para equilibrar a precisão e os custos computacionais. Nesse contexto, várias redes leves foram projetadas. A MobileNetV1 (HOWARD et al., 2017) introduz a convolução em profundidade, que alcança uma grande melhoria na eficiência. No desafio de classificação ImageNet, atinge 70,6% de precisão com parâmetros de 4,2M. Abordando a limitação do MobileNetV1, o MobileNetV2 (SANDLER et al., 2018), é baseado na estrutura residual invertida. MobileNetV3 (HOWARD et al., 2019), alcança melhor desempenho com ainda menos parâmetros através da incorporação do mecanismo de atenção. Métodos de segmentação semântica baseados em MobileNetV1 e MobileNetV2 são potencialmente úteis para aplicações de tempo real. O primeiro método de segmentação semântica baseado em deconvolução foi proposto por Noh, Hong e Han (2015) e chama-se DeconvNet. Conforme a Figura 17, o DeconvNet é baseado na arquitetura simétrica codificador-decodificador.

Na parte do codificador, as características semânticas são extraídas gradativamente enquanto a resolução é menor devido ao agrupamento máximo. Além disso, o método armazena as localizações dos valores máximos, chamados índices de pooling, nas janelas deslizantes durante o processo de pooling (ZEILER; FERGUS, 2014). No componente decodificador, o operador de unpooling utiliza os índices salvos para aumentar a resolução do mapa de recursos de baixa resolução em um mapa de recursos de alta resolução. Então, a deconvolução adota os filtros treináveis para reproduzir os mapas de características densas. A Figura 17 fornece ainda a comparação entre pooling e unpooling ou convolução e deconvolução.

2.3.1 DeepLab

A tarefa da segmentação semântica consiste em atribuir um rótulo de classificação para cada píxel na imagem (SHANMUGAMANI, 2018). Para atribuir estes rótulos de classificação aos píxeis, torna-se necessário implementar recursos precisos de identificação dos contornos dos objetos separando-os em segmentos distintos. Esta metodologia, definida para construção deste tipo de recurso, faz com que a arquitetura do modelo seja mais rigorosa do que a própria detecção de objetos com caixas delimitadoras (e.g., YOLO - do inglês “You Only Look Once”)

Figura 17 – A arquitetura do DeconvNet.



Fonte: (NOH; HONG; HAN, 2015)

(LATEEF; RUICHEK, 2019c).

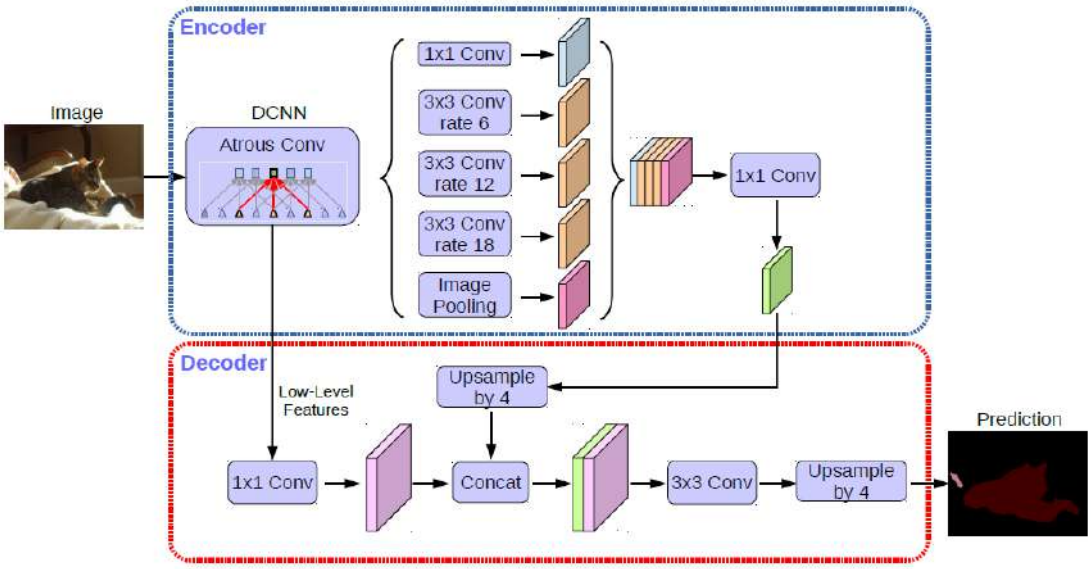
Chen et al. (2015) do Google, propuseram um modelo de rede neural convolucional profunda chamado DeepLab. Em vez de usar a deconvolução, eles propuseram a convolução Atrous. O algoritmo Atrous foi originalmente desenvolvido por Holschneider et al. (1990) para calcular a transformada wavelet não decimada. A arquitetura do DeepLab é semelhante à de Long, Shelhamer e Darrell (2015b), com algumas modificações, como converter camadas totalmente conectadas em camadas convolucionais, usando passo de 8 píxeis, pulando a sub-amostragem após as duas últimas camadas de agrupamento, modificando filtros convolucionais nas camadas, aumentando o comprimento das últimas três camadas convolucionais em 2x e a primeira camada totalmente conectada em 4x, introduzindo zeros.

O método proposto é combinado com campos aleatórios condicionais totalmente conectados, conseguindo produzir previsões semanticamente precisas e mapas de segmentação detalhados de forma eficiente. Yu e Koltun (2015), desenvolveram um projeto de módulo de rede neural convolucional para predição densa usando convoluções dilatadas para combinar informações contextuais multiescala sem perder resolução e analisando imagens redimensionadas para segmentação semântica. Este módulo pode ser conectado a arquiteturas existentes em qualquer resolução. A Figura 18 mostra um exemplo de convolução de dilatação com diferentes taxas de dilatação, que definem o espaçamento entre os valores em um kernel (LATEEF; RUICHEK, 2019a).

2.4 SEGMENTAÇÃO DE PISTA

O sistema de assistente de condução de segurança (ADAS), parte integrante do sistema de transporte inteligente, tem ganhado popularidade devido ao seu contínuo desenvolvimento e aplicação. Ele oferece serviços essenciais, como frenagem de emergência, decisões de condução auxiliar e alerta de emergência, garantindo a estabilidade e segurança do veículo. Isso ajuda a minimizar perdas econômicas e lesões em acidentes de trânsito. Um componente vital do ADAS é o sistema de alerta de saída de faixa (LDWS), que tem sido objeto de crescente atenção, conforme mencionado por Chen e Boukerche (2020).

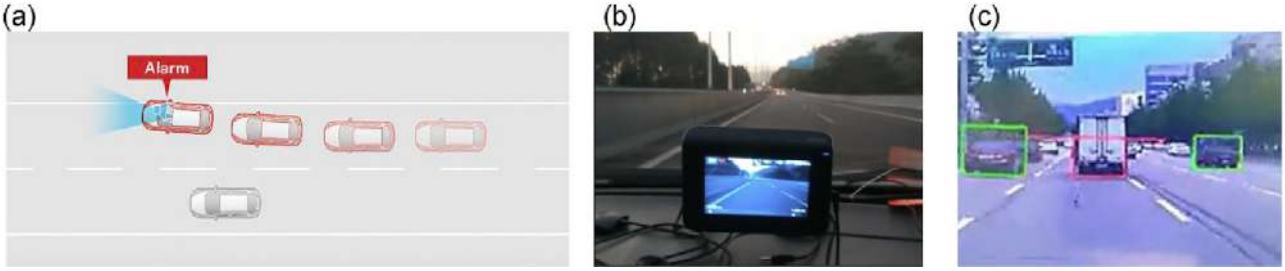
Figura 18 – DeepLab V3 e DeepLab v3+.



Fonte: (LATEEF; RUICHEK, 2019a)

O LDWS é um componente vital do ADAS, agindo como um sistema de segurança que alerta o motorista sobre desvios iminentes ou ocorridos da pista. No entanto, ele não pode controlar ativamente o veículo para evitar esses desvios. Utilizando informações da estrada captadas pelos sensores do veículo, o LDWS analisa o estado do veículo, o limite e tempo de aviso definidos, determinando se há tendência de desvio da faixa atual, conforme explicado por Lu, Cai e Li (2010). Quando o veículo condutor está prestes a desviar ou já desviou da pista e a luz de direção não está ligada, o LDWS emite um alerta ao motorista por meio de sinais auditivos, táteis ou visuais, como descrito por (LIANG, 2017). Ele engloba detecção de linha de pista, ajuste de linha, decisão de partida e liberação de aviso como parte de suas funcionalidades, a figura 19 explica o funcionamento do sistema.

Figura 19 – Demonstração do diagrama do sistema de alerta de saída de faixa e sistema real. (a) Diagrama de aviso de saída de faixa. (b) Monitor em um carro. (c) Aviso baseado no veículo dianteiro.

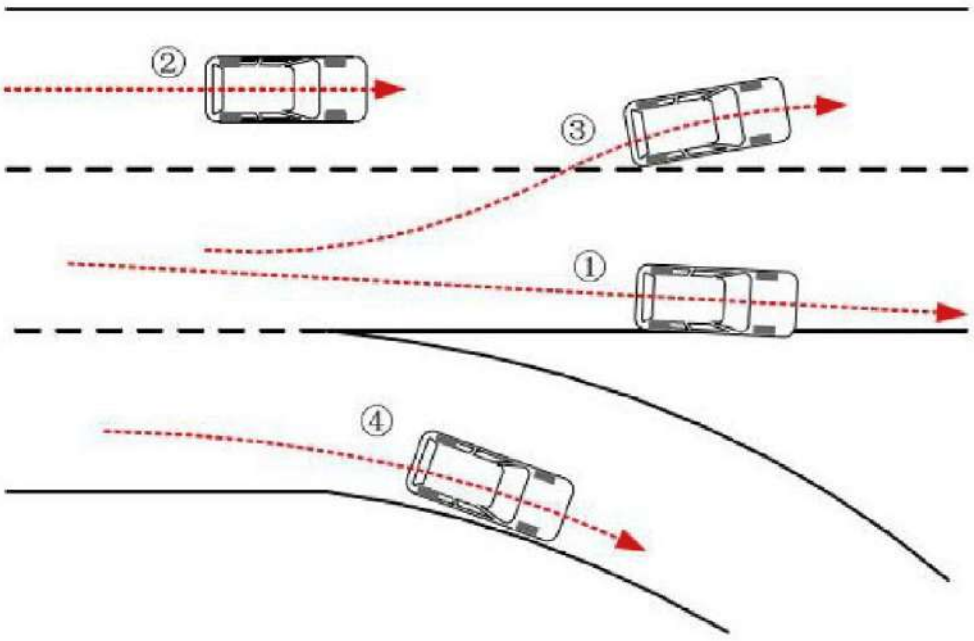


Fonte: (LU; CAI; LI, 2010)

Assim como as características acima do LDWS, esta área de pesquisa ainda tem muito o que fazer. Nas rodovias normais ou nas rodovias com alto grau de estrutura, o estado dos veículos

pode ser basicamente dividido em quatro situações (LIU, 2014), conforme mostra a Figura 20. A primeira é a saída inconsciente da pista. Neste caso, devido à desatenção do motorista, faz com que o veículo gire lenta e gradativamente para uma faixa lateral e, em seguida, se aproxime do limite da faixa, o que eventualmente leva o veículo a sair da faixa original. A segunda é que o veículo continua a circular na pista e mantém o seguimento normal. Neste caso, o veículo fica basicamente paralelo à borda das pistas, e leva muito tempo para o veículo se aproximar e cruzar a linha da pista. A terceira é que o motorista muda conscientemente de faixa. Neste caso, conforme as regras de trânsito, os motoristas devem ligar as piscas correspondentes para indicar a intenção de mudar de faixa para os veículos circundantes. A quarta é conduzir os veículos dentro ou fora da via expressa. Nesse caso, a curvatura da estrada geralmente é grande e os motoristas geralmente se concentram em desacelerar (LIU, 2014).

Figura 20 – Modelo de ciclo de condução rodoviária do veículo.



Fonte: (LU; CAI; LI, 2010)

O LDWS emite alerta ao motorista quando detecta desvios gradualmente da faixa, proporcionando um tempo de reação maior, reduzindo acidentes significativamente de saída de faixa (ZHANG et al., 2016). Além disso, ele corrige o hábito de não usar sinais de mudança de direção, prevenindo desatenção devido à condução prolongada ou fadiga, mediante avisos sonoros, vibrações e imagens. Estes alertas reduzem a fadiga ao dirigir, melhorando assim a segurança do veículo (ZHANG; WANG; LIU, 2018).

O LDWS é essencial para prevenir acidentes de saída de faixa e possui dois subsistemas de alerta: longitudinal, que detecta saídas devido à alta velocidade ou falta de controle, e transversal, que monitora desatenção ou abandono da operação de direção pelos motoristas (FAN, 2018; JUNG; KELBER, 2005).

A detecção de pista inicialmente envolve processamento de imagem tradicional e algoritmos de visão computacional. Embora métodos modernos, como aprendizado profundo, estejam substituindo gradualmente as abordagens tradicionais, estas últimas ainda contêm ideias e detalhes valiosos (CHEN et al., 2020). Os algoritmos tradicionais de detecção de linha de pista envolvem etapas, começando com a aquisição da imagem, onde a escolha da câmera e resolução é crucial. A condução automática é atualmente um foco central tanto na pesquisa acadêmica quanto na indústria de visão computacional e tecnologia robótica (SHIRKE; UDAYAKUMAR, 2019).

Para alcançar a condução totalmente automática, é essencial ter sensores e módulos de controle, sendo a detecção de pista por meio de câmeras um método crucial. Esse sistema não apenas posiciona o veículo corretamente na pista, mas também é fundamental para o planejamento futuro da rota ou saída da pista. A detecção precisa da pista via câmera é um fator-chave para a condução autônoma total (LIU, 2014).

O segundo passo é decidir qual região é a região de detecção de pista válida para evitar ruídos e complexos de processamento de imagens. Geralmente, a região válida pode ser apresentada como na Figura 21, mas em alguns casos, no entanto, a região válida pode ser variável na condução (YU; WU; SHEN, 2017), por exemplo, quando o veículo está indo de uma estrada de pista única para uma estrada de várias pistas, a região pode precisar ser mais amplo; ao dirigir da luz do dia para a noite, o comprimento e a largura da região podem ser alterados (LIN; HAN; HAHN, 2010). De qualquer forma, essa variação será alterada com base nas mudanças nas condições da estrada e na iluminação da estrada.

Na terceira etapa, a imagem na região válida passa por pré-processamento, como suavização e nitidez, para melhorar sua qualidade para extração de características da linha de pista. O desenvolvimento do algoritmo deve levar em conta as características específicas da imagem, como em casos de imagens noturnas, de neblina ou de sombra, exigindo algoritmos de aprimoramento diferentes. Por exemplo, cinco algoritmos podem ser usados para aprimorar uma imagem vaga: transformação exponencial aumenta o contraste das linhas de pista; transformação logarítmica melhora a claridade da imagem, mas as linhas de pista continuam vagas; equalização da imagem é melhor, mas a parte superior ainda é vaga; o algoritmo baseado em canal escuro melhora cores e contraste, mas as linhas de pista podem não estar claras; e o algoritmo baseado em retinex aumenta o contraste, mas pode não realçar bem as cores (KORTLI et al., 2017; TSAI; LIN; GUO, 2019).

Em casos de distorção na lente da câmera, como distorção radial e tangencial, é essencial corrigi-la antes da detecção da pista para uma detecção de cena mais precisa. A correção da geometria do mapeamento de coordenadas entre a imagem e a câmera proporciona uma conversão mais precisa, exceto para lente grande angular e olho de peixe. Além disso, para facilitar a detecção da linha de pista, é realizada uma transformação da imagem, como a transformação em mapeamento de perspectiva inversa (IPM). Esta transformação tem vantagens, reduzindo

Figura 21 – Divisão regional da imagem da pista em geral.

Fonte: (LIN; HAN; HAHN, 2010)

cálculos, embora perca algumas informações, e facilitando a extração de linhas paralelas na visão aérea (WANG; QI; MA, 2014; CHOI; PARK; YOE, 2018).

Na quarta etapa, ocorre a extração de características da imagem para detectar as linhas da pista. Os algoritmos de extração de características usam a forma da linha da pista, o gradiente de píxel e as características de cor na imagem. Primeiro, a imagem é convertida para escala de cinza e, em seguida, são extraídas informações sobre a região da pista ou as bordas. Esses algoritmos podem ser classificados como baseados em similaridade ou descontinuidade (JOSHY; JOSE, 2014).

3 METODOLOGIA

O estudo se concentra na revisão da literatura, configuração do ambiente de desenvolvimento, experimentos em conjuntos de dados específicos e técnicas de aumento de dados para treinamento de modelos de aprendizagem profunda.

3.1 Levantamento Bibliográfico

Neste trabalho, realizamos um levantamento bibliográfico de livros e portais de universidades prestigiadas para encontrar a melhor forma de realizar este trabalho.

Os sites de busca mais utilizados em pesquisas bibliográficas são o SciELO e o Google Acadêmico. O intervalo de pesquisa é de 22 anos, e são pesquisados trabalhos acadêmicos entre 2000 e 2022. As principais referências utilizadas são:

- ALVAREZ, J. M.; LECUN, Y.; GEVERS, T.; LOPEZ, A. M. Semantic road segmentation via multi-scale ensembles of learned features. Proceedings of the European Conference on Computer Vision, 2012.
- CHEN, L.; PAPANDREOU, G.; KOKKINOS, I.; MURPHY, K.; YUILLE, A. L. Semantic image segmentation with deep convolutional nets and fully connected CRFs. arXiv:1412.7062, 2015.
- CHEN, L. et al. Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587, 2017.
- CHEN, L.; ZHU, Y.; PAPANDREOU, G.; SCHROFF, F.; ADAM, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. ECCV, 2018.
- CHOLLET, F. Xception: Deep Learning with Depthwise Separable Convolutions. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Julho de 2017.
- HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.
- HOLSCHNEIDER, M.; KRONLAND-MARTINET, R.; MORLET, J.; TCHAMITCHIAN, P. A real–time algorithm for signal analysis with the help of the wavelet transform. Wavelets, 1990.
- LATEEF, F.; RUICHEK, Y. Survey on semantic segmentation using deep learning techniques. Neurocomputing, vol. 338, pp. 321–348, Abril de 2019.

- LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015.

- WANG, P.; CHEN, P.; YUAN, Y.; LIU, D.; HUANG, Z.; HOU, X.; COTTRELL, G. W. Understanding convolution for semantic segmentation. 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 2018.

3.2 Implementação

Todo projeto que envolve design de software envolve a análise do problema e a minimização de erros de implementação por meio de pesquisas. Neste trabalho, utilizamos a rede **DeepLabv3**, uma arquitetura avançada de rede neural convolucional (CNN) projetada por engenheiros do Google, foi selecionada. Considerando sua eficiência computacional, facilita o processo de treinamento, e sua estrutura modular permite adaptação a diversos tipos de dados. Com seus recursos multiescala, alcança excelente precisão de segmentação em vários bancos de dados de referência, como PASCAL VOC e Cityscapes.

A arquitetura segue a seguinte estrutura:

- **Convolução Atrous:** A convolução atrosa (ou dilatada) é uma técnica que permite aumentar o campo receptivo (ou seja, a área da imagem "vista"por cada operação de convolução) sem aumentar o número de parâmetros ou a quantidade de cálculo necessária. Isso é feito inserindo espaços entre pixels em filtros de convolução padrão, permitindo que o modelo capture uma gama mais ampla de informações sem perder resolução.

- **Atrous Spatial Pyramid Pooling (ASPP):** O ASPP desempenha um papel fundamental na arquitetura DeepLabv3 ao possibilitar que a rede colete informações contextuais em diferentes escalas. A estratégia consiste na aplicação de convoluções atrous com taxas de dilatação diversas de maneira simultânea, seguida pela fusão dos resultados. Esse processo auxilia o modelo a capturar tanto os detalhes minuciosos quanto os contextos abrangentes das imagens, o que é essencial para obter uma segmentação precisa.

- **Otimização do Campo de Recepção:** O DeepLabv3 aprimora eficientemente o campo de recepção para capturar contexto em diversas escalas. Ao modificar as taxas de dilatação no ASPP e nas convoluções atrous, o modelo pode ser personalizado para variados tamanhos de imagem e necessidades de detalhamento, o que resulta em uma melhoria na precisão da segmentação.

- **Resposta à Diversidade de Escalas:** Uma das grandes vantagens do DeepLabv3 é sua capacidade de lidar com objetos e características de diferentes tamanhos em uma imagem. Isso é possível devido à união de várias taxas de dilatação no ASPP, o que proporciona uma

análise minuciosa sem a obrigatoriedade de ajustar o tamanho da imagem ou modificar a estrutura da rede neural.

- **Eficiência e Precisão:** Mesmo sendo complexa, a arquitetura foi desenvolvida para garantir uma eficiência computacional eficaz. Usando convoluções atrofiadas de forma inteligente e adotando a estratégia de pooling espacial, é possível alcançar alta precisão na segmentação sem necessidade de aumentar significativamente os recursos computacionais.

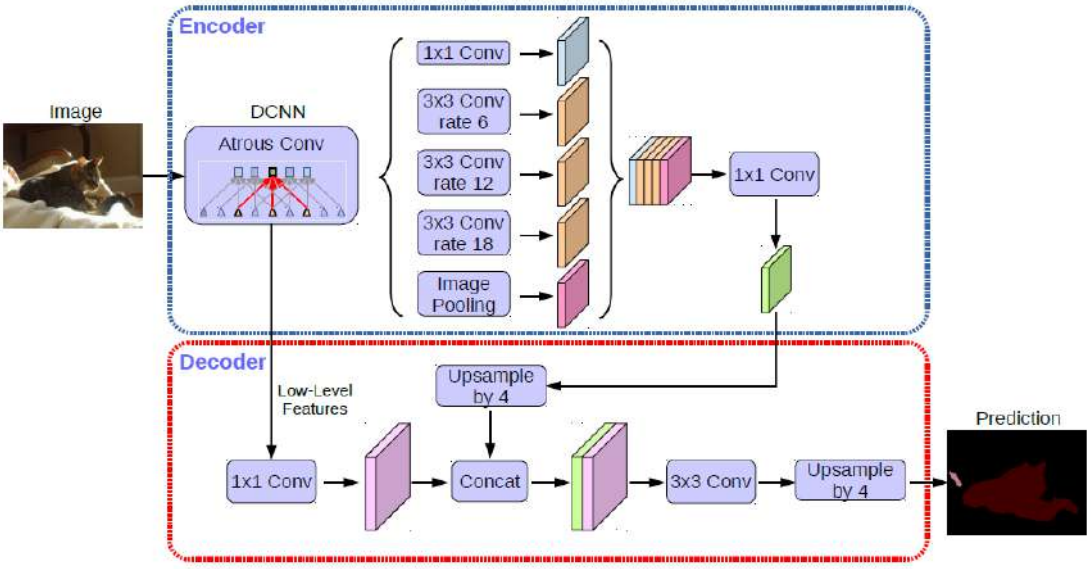


Figura 22 – Arquitetura DeepLabV3+

Fonte: Image Courtesy: DeepLabV3+ [Chen et al.]

O desenvolvimento perpassa por múltiplas fases, tais como incorporação de bibliotecas, o processamento de dados, a definição de conjuntos de dados e carregadores de dados, além da configuração e avaliação do modelo durante o treinamento.

3.3 Ambiente de Desenvolvimento

Nesse trabalho utilizamos as seguintes tecnologias:

- `os` - Biblioteca padrão do Python para interação com o sistema operacional.
- `cv2 (OpenCV)` - Biblioteca de visão computacional e machine learning.
- `numpy` - Biblioteca para computação científica com suporte a arrays e matrizes.
- `pandas` - Biblioteca para manipulação e análise de dados.

- `random` - Módulo que implementa geradores de números pseudoaleatórios para várias distribuições.
- `tqdm` - Biblioteca para barras de progresso em loops.
- `seaborn` - Biblioteca de visualização de dados baseada em matplotlib.
- `matplotlib.pyplot` - Módulo para criação de gráficos estáticos, animados e interativos.
- `torch` - Pacote da PyTorch para tensores e operações de autograd.
- `torch.nn` - Módulo da PyTorch para construir redes neurais.
- `torch.utils.data.DataLoader` - Facilitador para carregamento de dados em redes neurais.
- `albumentations` - Biblioteca rápida de aumento de imagem para deep learning.
- `Linguagem:` - Python

3.4 Experimentação

Neste estudo, investigamos o desempenho de redes neurais sob diferentes configurações de treinamento utilizando os conjuntos de dados KITTI e UESB. As estratégias de treinamento avaliadas incluíram treinamento regular, treinamento exclusivo usando conjuntos de dados específicos e refinamento com e sem aumento de dados. Para todos os testes, 100 épocas foram utilizadas para treinamento.

Os experimentos foram organizados em quatro categorias principais:

1. **Rede Treinada com o dataset do KITTI (REDE 1)**: O conjunto de dados KITTI, amplamente utilizado em aplicações de visão computacional para carros autônomos, foi usado para treinar a primeira rede neural. Este conjunto de dados é caracterizado pela diversidade e riqueza de cenários automotivos do mundo real e fornece uma base sólida para aprender modelos de detecção e reconhecimento.
2. **Treinamento da Rede 2 exclusivamente com o dataset da UESB:** A segunda rede foi treinada utilizando apenas dados da Universidade Estadual do Sudoeste da Bahia (UESB), permitindo avaliar a capacidade de aprendizado e generalização do modelo usando um conjunto de dados mais limitado e uma variedade de recursos associados ao KITTI.
3. **Ajuste Fino da Rede 1 com dados da UESB (Rede 3):** Neste experimento, a Rede 1 foi melhorada e pré-treinada no conjunto de dados KITTI usando dados UESB. O objetivo desta abordagem é explorar o conhecimento adquirido pela rede sobre os conjuntos de

dados mais amplos e adaptá-lo para um bom desempenho conforme as especificidades dos dados da UESB.

4. **Ajuste Fino da Rede 1 com dados da UESB e Data Augmentation (Rede 4):** No experimento final, melhoramos a Rede 1 com uma estratégia de aumento de dados passo a passo usando transformações de imagem do conjunto de dados da UESB. As operações de zoom incluem corte aleatório, projeção vertical e horizontal, rotação aleatória de 90 graus, distorção de grade, transformação elástica, ajustes de brilho e contraste e muito mais. O objetivo deste método é enriquecer o conjunto de treinamento e melhorar a generalização do modelo.

Para todos os conjuntos de dados, classificamos 80% para treinamento, 10% para validação e 10% para teste. Para o conjunto UESB com aumento de dados, o treinamento foi aumentado com duas operações de aumento para cada imagem, melhorando a variabilidade e robustez dos dados de treinamento. Em contraste, os conjuntos de validação e teste foram mantidos inalterados para avaliar consistentemente a capacidade de generalização do modelo. Cada imagem no conjunto de dados possui um rótulo correspondente. Esta é uma imagem binária que representa a área (neste caso uma pista) que você deseja identificar. Neste contexto, podemos pensar nesses campos como classes. No nosso caso, a classe é definida pela própria pista e pelo ambiente que a rodeia. Essa segmentação binária permite que o modelo aprenda a distinguir regiões de pista das regiões vizinhas, o que é essencial para detectar e analisar pista em uma imagem.

Figura 23 – Imagem original e seu rotúlo.

Fonte: Kitti Dataset

A imagem à direita é a imagem original e a imagem à esquerda é o rótulo correspondente. O tutorial usa um total de 50 imagens com seus rótulos correspondentes. Vamos continuar o processo criando uma função para preparar o conjunto de dados. Nesta fase, devem ser realizadas operações na imagem para completar as bordas, pois a rede espera que o tamanho da largura e da altura seja divisível por 16. Portanto, múltiplas operações foram utilizadas para preenchimento. conforme a Figura 24.

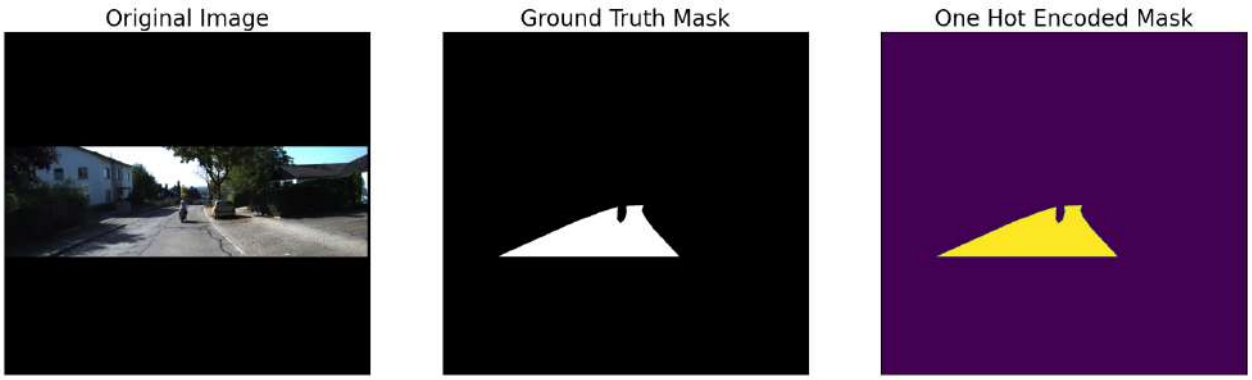


Figura 24 – Arquitetura DeepLabV3+

Fonte: Kitti Dataset

Os resultados desses experimentos fornecem informações valiosas sobre a eficácia de diferentes estratégias de treinamento em situações específicas de visão computacional. A comparação de modelos treinados exclusivamente em diferentes conjuntos de dados e modelos ajustados com e sem aumento pode identificar as melhores práticas para o desenvolvimento de sistemas de visão computacional robustos e eficientes, especialmente para aplicações automotivas.

Para rede de treinamento 2. Adquirimos imagens abrangendo todo o campus com uma câmera montada em uma motocicleta no campus da UESB em Vitória da Conquista. Em seguida, retiramos o quadro e isolamos também a região de interesse. É importante ressaltar que as imagens foram tiradas em um ângulo com visão clara do céu, portanto houve ruído durante o movimento. A imagem abaixo mostra a imagem original.

Figura 25 – imagem original compus UESB vitória da Conquista.

OpenCV é usado para restringir estrategicamente regiões de interesse calculadas para obter resultados satisfatórios. A imagem abaixo mostra o resultado obtido após o recorte da imagem.

Figura 26 – Imagem, campus UESB pós Processamento.

O modelo foi treinado usando uma rede DeepLabV3+ com pesos ResNet101 e adições artificiais de dados, uma função de perda especial (DiceLoss) para capturar a eficiência da segmentação e um otimizador padrão (Adam) para ajustar os pesos para estimar a unidade na interseção. As métricas (IoU) são usadas para monitorar o desempenho do modelo.

3.5 Resultados Quantitativos

Devido a essas limitações, os resultados obtidos são insatisfatórios e podem ser considerados uma etapa preliminar do treinamento. Usar mais épocas de treinamento, conjuntos de dados mais abrangentes e técnicas adicionais, como aprendizagem por transferência, pode melhorar o desempenho do modelo.

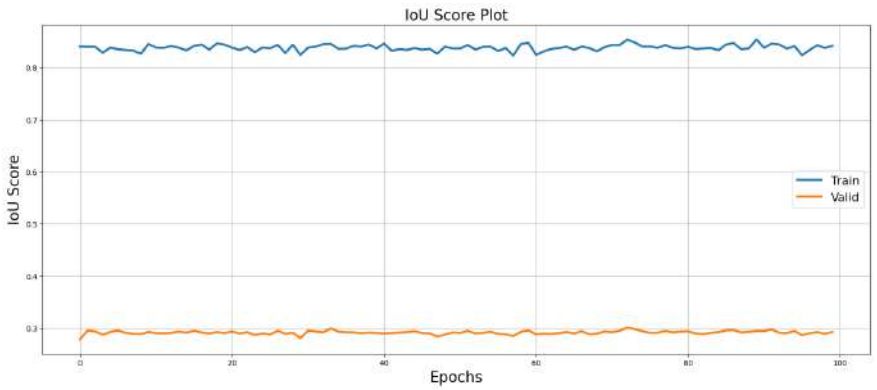


Figura 27 – Informações sobre a perda de dados e a métrica IoU (Intersection over Union), Rede 01

Fonte: Kitti Dataset

A curva de perda de dados mostra a mudança na perda durante o período de treinamento. A perda é uma medida da discrepância entre as previsões do modelo e os rótulos reais. O ideal é que a perda diminua durante o treinamento, indicando que o modelo está aprendendo a segmentar corretamente as classes de interesse. A métrica IoU é uma métrica de avaliação amplamente utilizada na segmentação semântica. Mede a sobreposição entre a segmentação prevista pelo modelo e os rótulos reais. Quanto maior o valor de IoU, melhor será a sobreposição das regiões segmentadas. Você pode avaliar o desempenho do seu modelo durante o treinamento observando a curva de perda de dados e as métricas de IoU. Esperamos que a perda diminua gradualmente e a métrica IoU aumente, indicando que a segmentação semântica continua a melhorar. A análise desses gráficos é importante para compreender o progresso do treinamento e identificar possíveis problemas, como overfitting (quando o modelo superajusta o conjunto de treinamento e não generaliza bem para novos dados).

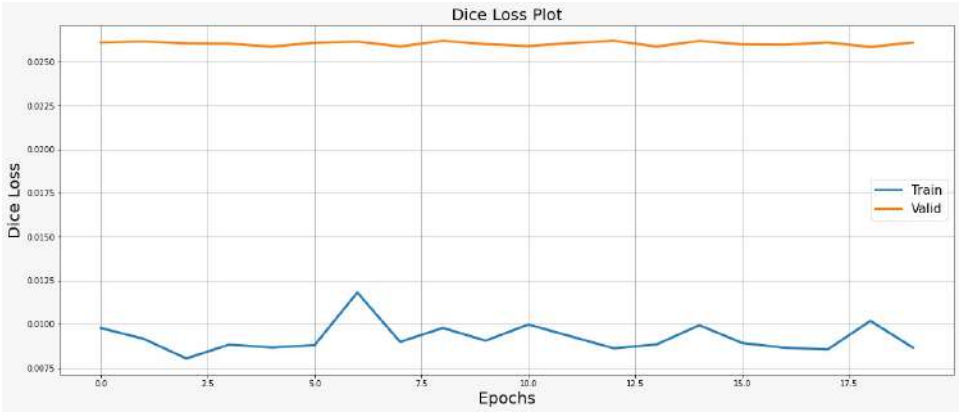


Figura 28 – Após o ajuste fino da rede, conseguimos observar uma melhora expressiva

Fonte: Kitti Dataset

3.6 Discussão

Os resultados do treinamento não alcançaram desempenho satisfatório no conjunto de dados Kitti, conforme mostrado na Figura 29 devido a algumas limitações durante o treinamento. Primeiro, o modelo é treinado do zero. Ou seja, os pesos iniciais são definidos aleatoriamente. Mais períodos de treinamento podem ser necessários para que o modelo aprenda recursos relacionados à segmentação semântica das classes de interesse (pista e ambiente).

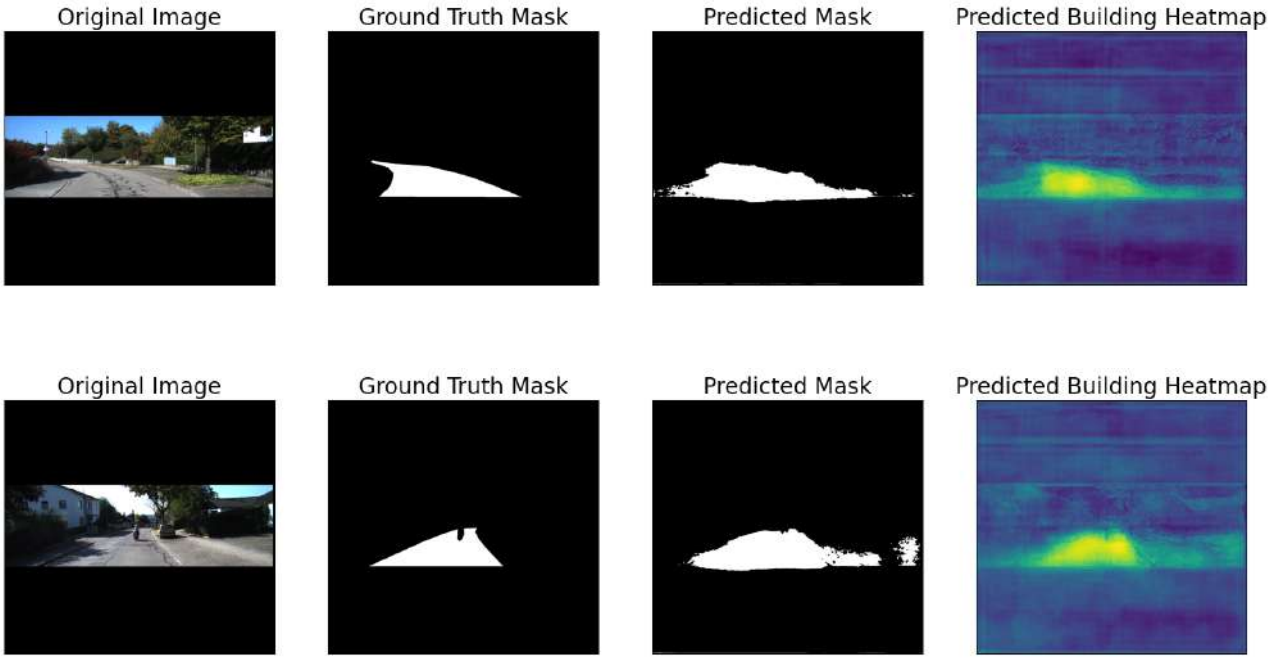


Figura 29 – Treinamento com dados da Kitti Dataset.

Além disso, foram realizadas apenas 100 sessões de treinamento, o que pode não ser suficiente para que o modelo atinja sua capacidade máxima de aprendizagem. Em geral, treinar uma rede neural para uma tarefa complexa, como a segmentação semântica, pode exigir mais tempo para que o modelo capture padrões e representações de dados mais precisos.

Outro fator limitante é o tamanho limitado do conjunto de imagens utilizado para treinamento. Conjuntos de dados maiores e mais diversificados tendem a produzir modelos mais poderosos com melhor generalização. Com um conjunto limitado de imagens, o modelo pode ter dificuldades para aprender uma representação suficientemente abrangente para realizar a segmentação com precisão em uma variedade de condições e perspectivas. Devido a essas limitações, os resultados obtidos são insatisfatórios e podem ser considerados um treinamento preliminar. Usar mais épocas de treinamento, conjuntos de dados mais abrangentes e técnicas adicionais, como aprendizagem por transferência, pode melhorar o desempenho do modelo.

Os resultados mostrados na figura 30 mostram os resultados otimizados da imagem original, máscara e rede treinada. Ao analisar a máscara pretendida, percebe-se que os contornos são mais nítidos em comparação aos rótulos. Isso destaca a obtenção de excelentes resultados durante a avaliação de desempenho do modelo.

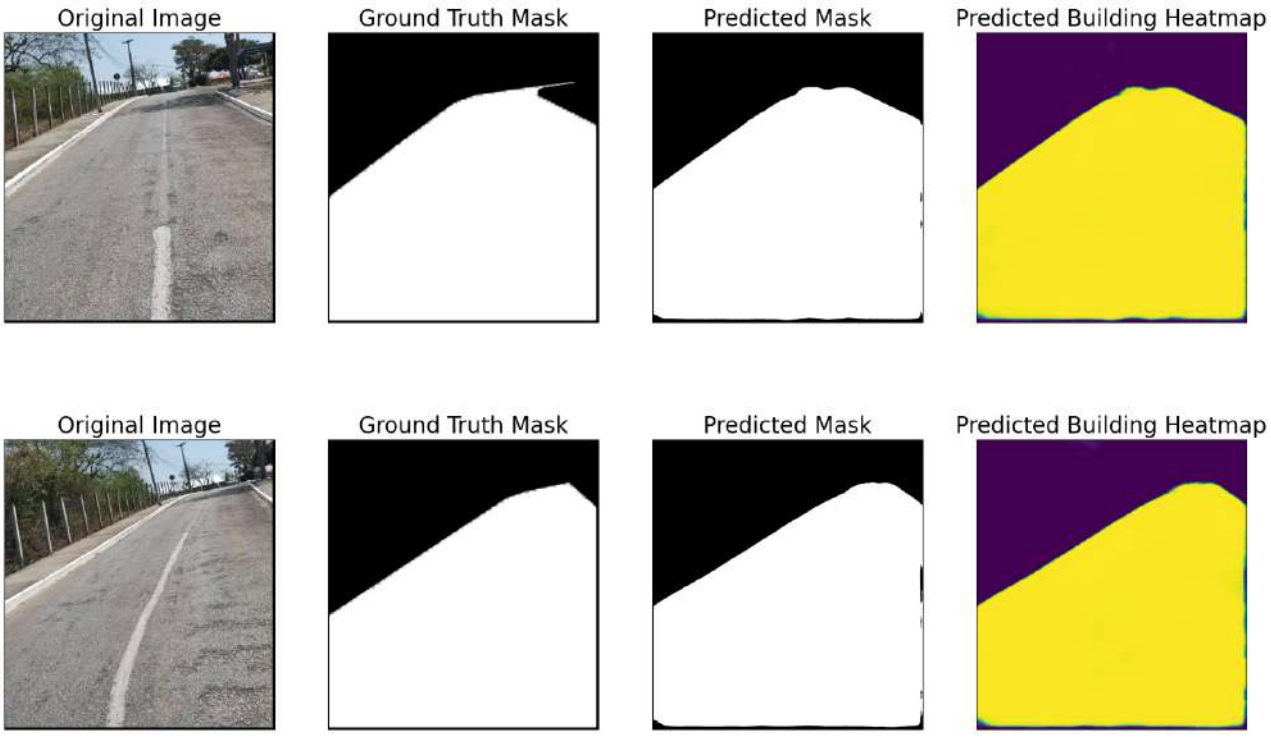


Figura 30 – Treinamento com dados da UESB utilizando pesos da ResNet101

A terceira rede foi treinada por meio do ajuste fino (treinamento de transferência) da REDE 1 utilizando dados da UESB.

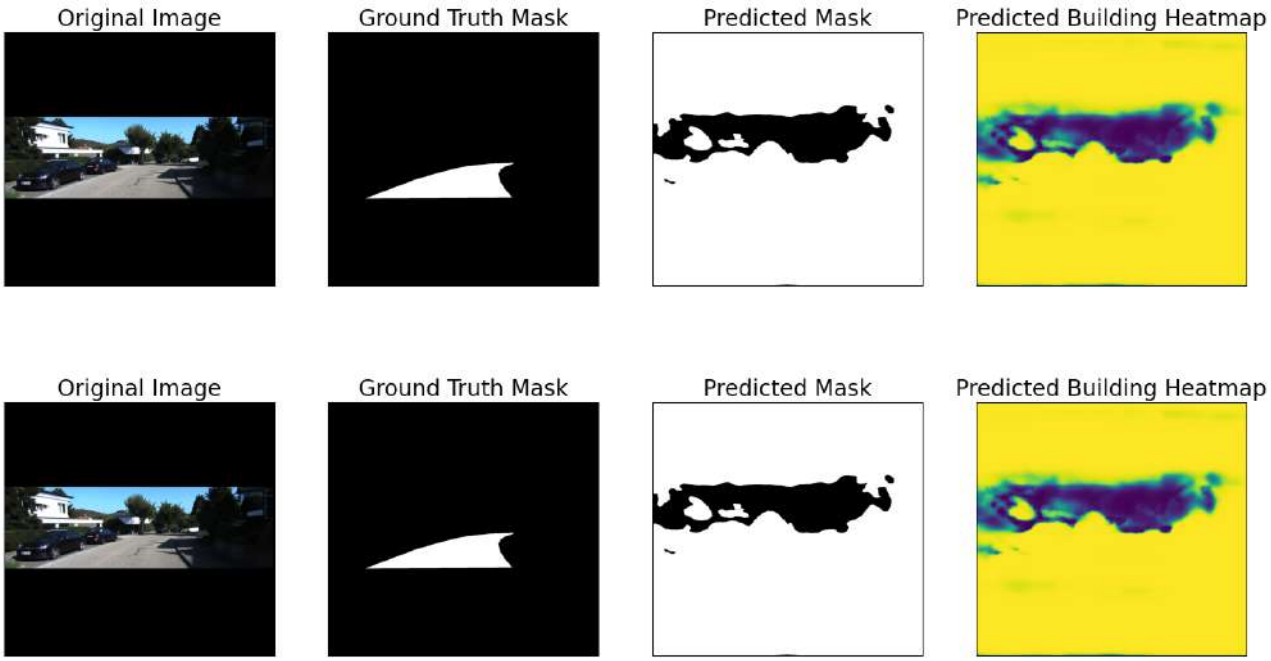
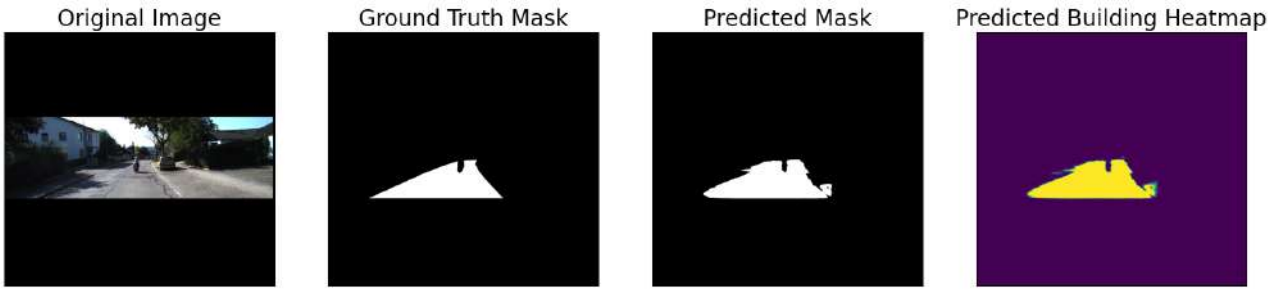


Figura 31 – Treinamento com dados da UESB utilizando pesos da ResNet101

Neste experimento, realizamos o carregamento do modelo treinado com os dados do KITTI para treinar a nossa rede na UESB. Infelizmente, não alcançamos resultados satisfatórios; ao contrário, observamos uma degradação em relação aos resultados anteriores.

Após o ajustes finos nas camadas do modelo, observamos um notável avanço nos resultados. Este processo de refinamento é crucial para adaptar o modelo às nuances específicas dos dados e melhorar sua capacidade de realizar tarefas específicas com maior precisão, a imagem abaixo ilustra o resultado obtido. Figura 32.



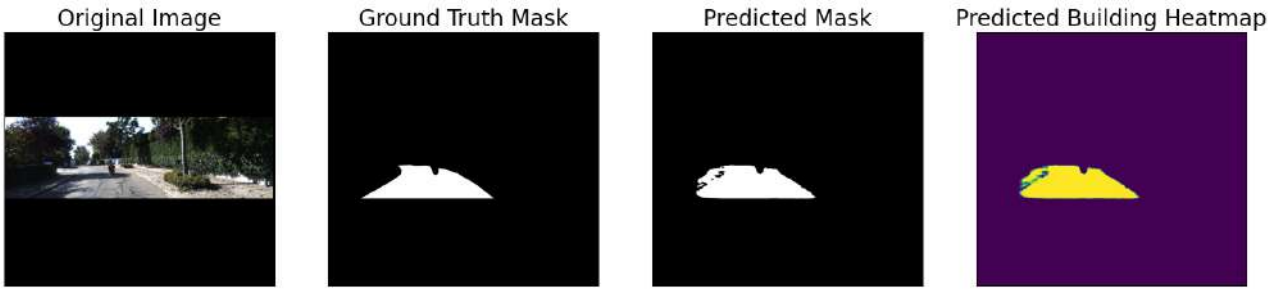


Figura 32 – Treinamento com ajuste fino na rede

A última rede avaliada foi feito um Ajuste Fino (Transfer Learning) da REDE 1, utilizando os dados da UESB + data augmentation das imagens. Os resultados não foram tão promissores. Figura 33.

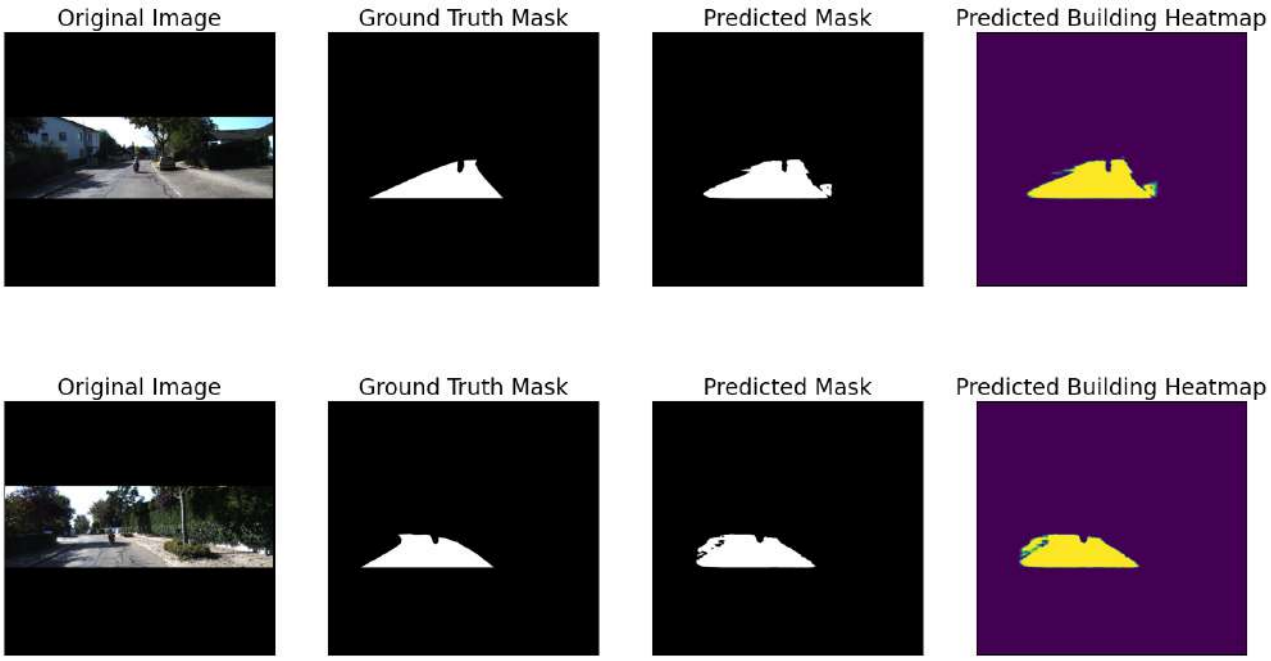
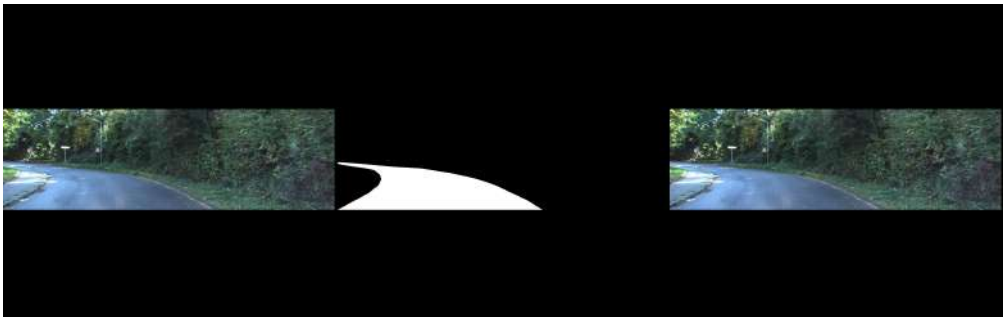
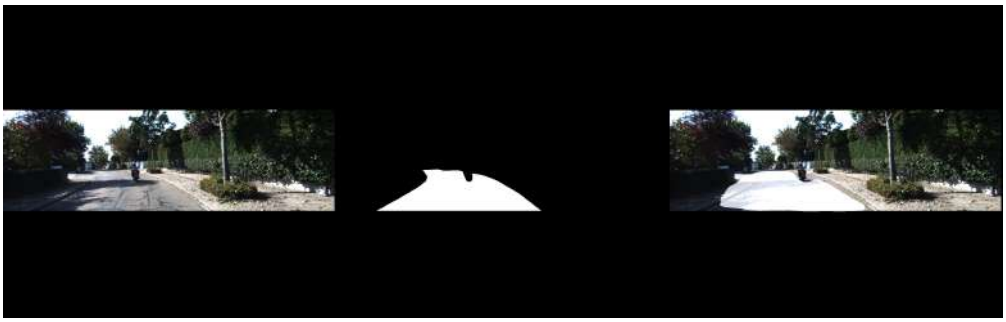


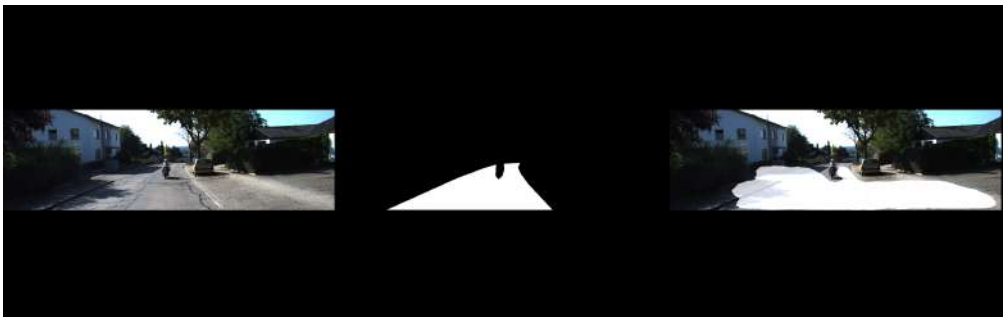
Figura 33 – Treinamento com dados da UESB utilizando pesos da ResNet101

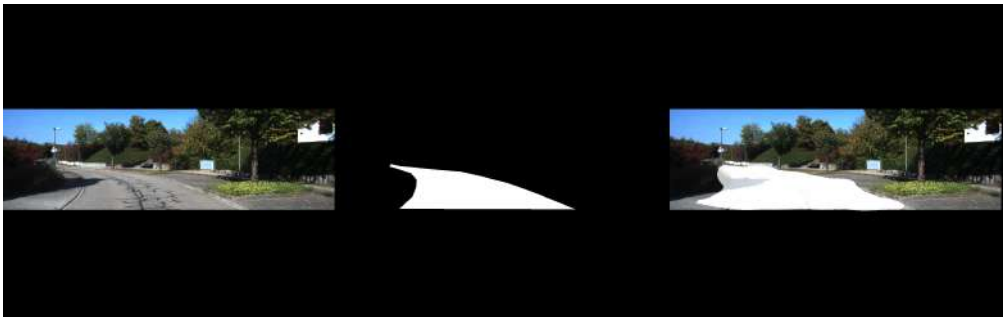
3.7 Análise Qualitativa

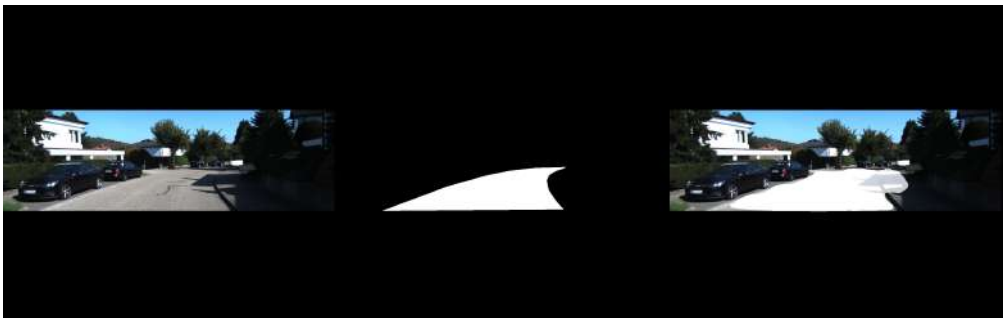
Resultados de segmentação semântica obtidos com os dados do Kitti Dataset.



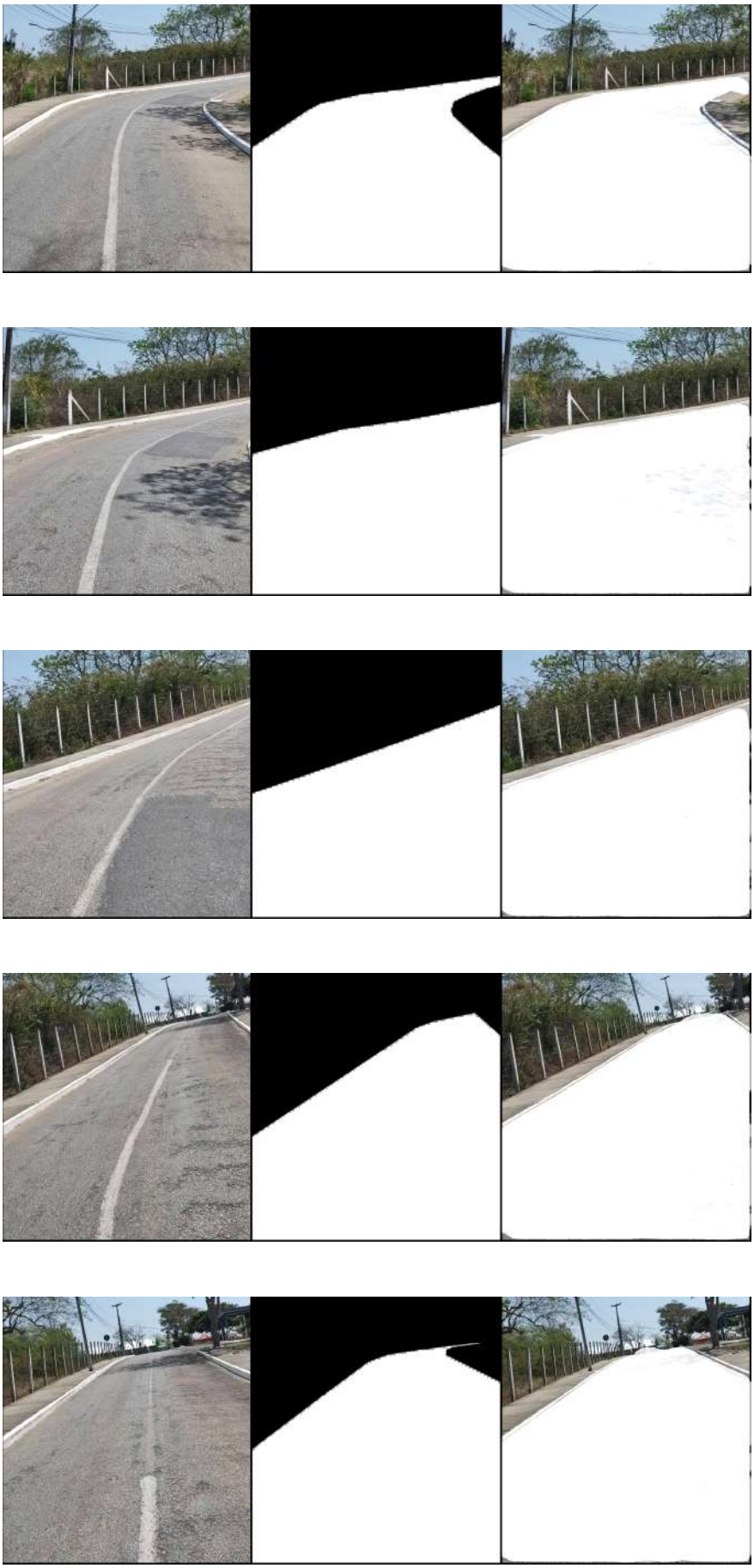








Resultado dados da UESB.



3.8 Considerações Finais

A consideração final deste trabalho destaca a importância da segmentação semântica na compreensão profunda de imagens, que é de interesse crescente na visão computacional. A implementação e avaliação do modelo DeepLabv3 demonstra a eficácia desta abordagem em distinguir e classificar diferentes objetos em imagens com precisão significativa. Experimentos mostram que o DeepLabv3 melhorou significativamente a precisão da segmentação semântica em comparação com modelos anteriores por meio da combinação de arquitetura e tecnologias avançadas, como módulos Atrous Convolution e Atrous Spatial Pyramid Pooling (ASPP).

No entanto, apesar do progresso significativo, esta pesquisa também tem limitações, especialmente porque requer abundância de dados anotados para treinamento e a computação intensiva necessária para treinamento e inferência de modelos. Esses aspectos destacam a importância de pesquisas futuras visando eficiência computacional e métodos de aprendizagem semissupervisionados ou não supervisionados que possam reduzir a dependência de dados anotados manualmente.

Trabalhos futuros explorarão técnicas de otimização de modelos e integrarão o DeepLabv3 com novas abordagens, como redes generativas adversárias (GANs) e aprendizado por reforço, para melhorar ainda mais a precisão da segmentação semântica e expandir sua aplicabilidade além de imagens estáticas para ambientes de vídeo e em tempo real. Além disso, investigar a eficácia de diferentes técnicas de pré-processamento e aumento de dados pode fornecer uma maneira valiosa de melhorar a generalização do modelo sob diferentes condições.

Em resumo, este trabalho não apenas demonstra a viabilidade e eficiência do DeepLabv3 para segmentação semântica de imagens, mas também abre caminho para pesquisas futuras explorarem novas áreas de visão computacional e inteligência artificial.

REFERÊNCIAS

ALBAWI, S.; MOHAMMED, T. A.; AL-ZAWI, S. Understanding of a convolutional neural network. In: *ICET2017*. Antalya, Turkey: [s.n.], 2017. p. 6. Citado 2 vezes nas páginas 22 e 23.

BADRINARAYANAN, V.; KENDALL, A.; CIPOLLA, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 39, n. 12, p. 2481–2495, 2017. Citado na página 34.

BENGIO, Y. Learning deep architectures for ai. *Foundations and Trends in Machine Learning*, v. 2, n. 1, p. 1–127, 2009. Citado na página 22.

BRAGA, A. P.; PONCE, C.; LUDERMIR, T. B. *Redes neurais artificiais: teoria e aplicações*. Rio De Janeiro: Ltc Editora, 2007. Citado na página 20.

BRUNA, J. et al. Spectral networks and locally connected networks on graphs. *arXiv*, v. 1312.6203, may 2014. ArXiv:1312.6203 [cs]. Citado na página 30.

CARBONETTO, P.; FREITAS, N. D.; BARNARD, K. A statistical model for general contextual object recognition. In: *Proceedings of the European Conference on Computer Vision*. [S.l.: s.n.], 2004. p. 350–362. Citado na página 31.

CHANG, C.-H. Deep and shallow architecture of multilayer neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, v. 26, n. 10, p. 2477–2486, oct 2015. Citado na página 22.

CHEN, L.-C. et al. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2015. Citado na página 36.

CHEN, W. et al. Lane departure warning systems and lane line detection methods based on image processing and semantic segmentation: A review. *Journal of Traffic and Transportation Engineering (English Edition)*, Elsevier, v. 7, n. 6, p. 748–774, dec 2020. Citado na página 39.

CHEN, Y.; BOUKERCHE, A. A novel lane departure warning system for improving road safety. In: IEEE. *2020 IEEE International Conference on Communications (ICC)*. Dublin, 2020. Citado na página 36.

CHOI, Y.; PARK, J. H.; JUNG, H. Y. Lane detection using labeling based ransac algorithm. *International Journal of Computer and Information Engineering*, v. 12, n. 4, p. 245–248, 2018. Citado 2 vezes nas páginas 8 e 19.

CHOI, Y.; PARK, J. H.; YOE, H. Lane detection using labeling based ransac algorithm. *International Journal of Computer and Information Engineering*, v. 12, n. 4, p. 245–248, 2018. Citado na página 40.

COHEN, T. S.; WELLING, M. Steerable cnns. *arXiv*, v. 1612.08498, dec 2016. ArXiv:1612.08498 [cs, stat]. Citado 2 vezes nas páginas 28 e 29.

CORDTS, M. et al. The cityscapes dataset for semantic urban scene understanding. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 3213–3223. Citado na página 15.

COX, D.; DEAN, T. Neural networks and neuroscience-inspired computer vision. *Current Biology*, v. 24, n. 18, p. R921–R929, sep 2014. Citado na página 31.

DAI, J. et al. Deformable convolutional networks. *arXiv*, v. 1703.06211, jun 2017. ArXiv:1703.06211 [cs]. Citado 2 vezes nas páginas 25 e 26.

Dartmouth College. *Artificial Intelligence (AI) Coined at Dartmouth*. 1956. <https://home.dartmouth.edu/about/artificial-intelligence-ai-coined-dartmouth>. Citado na página 13.

DONG, Z. et al. Codenet: Efficient deployment of input-adaptive object detection on embedded fpgas. *arXiv*, v. 2006.08357, jan 2021. ArXiv:2006.08357 [cs, eess]. Citado na página 27.

DROZDZAL, M. et al. The importance of skip connections in biomedical image segmentation. In: *Deep Learning and Data Labeling for Medical Applications*. [S.l.: s.n.], 2016. p. 179–187. Citado na página 33.

EDELMAN, S.; POGGIO, T. Integrating visual cues for object segmentation and recognition. *Opt. News*, v. 15, n. 5, p. 8–13, 1989. Citado na página 31.

EDUKA.AI. Como a inteligência artificial surgiu e por que foi criada. *Eduka.AI Blog*, 2023. Disponível em: <https://eduka.ai/inteligencia-artificial/>. Citado na página 13.

FAN, B. *Research on Monocular Vision Detection Method of Structured Road Lane Line*. Dissertação (Mestrado) — Hunan University, Changsha, 2018. Citado na página 38.

FILHO, O. M.; NETO, H. V. *Processamento Digital de Imagens*. Rio de Janeiro: Brasport, 1999. Citado na página 18.

FURHT, B.; AKAR, E.; ANDREWS, W. A. *Digital Image Processing: Practical Approach*. Cham: Springer International Publishing, 2018. Citado na página 18.

GAO, Y. et al. Accurate segmentation of ct male pelvic organs via regression-based deformable models and multi-task random forests. *IEEE Transactions on Medical Imaging*, v. 35, n. 6, p. 1532–1543, 2016. Citado na página 32.

GEIGER, A. et al. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, Sage Publications Sage UK: London, England, v. 32, n. 11, p. 1231–1237, 2013. Citado na página 14.

GENG, Q.; ZHOU, Z.; CAO, X. Survey of recent progress in semantic image segmentation with cnns. *Science China Information Sciences*, v. 61, n. 5, p. 051101, 2018. Citado na página 33.

GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. Upper Saddle River, N.J.: Prentice Hall, 2002. Citado na página 20.

GORI, M.; MONFARDINI, G.; SCARSELLI, F. A new model for learning in graph domains. In: *Proceedings of the IEEE International Joint Conference on Neural Networks*. [S.l.: s.n.], 2005. v. 2, p. 729–734. Citado na página 30.

GUNDE, P. S.; SHIRGAVE, S. K. Survey on semantic segmentation. *International Journal of Computer Sciences and Engineering*, v. 6, n. 12, p. 603–606, Dec 2018. 31 dez. 2018. Citado na página 33.

HARIHARAN, B. et al. Hypercolumns for object segmentation and fine-grained localization. In: IEEE. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.], 2015. p. 447–456. Citado na página 33.

HAWKINS, D. M. The problem of overfitting. *ChemInform*, v. 35, n. 19, may 2004. Citado na página 25.

HAYKIN, S. *Redes Neurais: Princípios e Práticas*. 2. ed. São Paulo: Bookman, 2001. 900 p. Citado na página 21.

HAYKIN, S.; ENGEL, P. M. *Redes neurais: princípios e prática*. São Paulo: Artmed, 2007. Citado na página 20.

HE, K. et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2016. p. 770–778. Citado na página 34.

HOLSCHNEIDER, M. et al. A real–time algorithm for signal analysis with the help of the wavelet transform. In: ____. *Wavelets*. [S.l.]: Springer, 1990. p. 286–297. Citado na página 36.

HOWARD, A. G. et al. Searching for mobilenetv3. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. [S.l.: s.n.], 2019. p. 1314–1324. Citado na página 35.

HOWARD, A. G. et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. Citado na página 35.

HUANG, G. et al. Condensenet: An efficient densenet using learned group convolutions. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2018. p. 2752–2761. Citado na página 28.

HUANG, G. et al. Densely connected convolutional networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2017. p. 4700–4708. Citado na página 35.

JI, Y. et al. Graph model-based salient object detection using objectness and multiple saliency cues. *Neurocomputing*, v. 323, p. 188–202, jan 2019. Citado na página 31.

JOSHY, N.; JOSE, D. Improved detection and tracking of lane marking using hough transform. *International Journal of Computer Science and Mobile Computing*, v. 3, n. 8, p. 507–513, 2014. Citado na página 40.

JUNG, C. R.; KELBER, C. R. Lane following and lane departure using a linear parabolic model. *Image Vision Computing*, v. 23, n. 13, p. 1192–1202, 2005. Citado na página 38.

KIPF, T. N.; WELLING, M. Semi-supervised classification with graph convolutional networks. *arXiv*, v. 1609.02907, feb 2017. ArXiv:1609.02907 [cs, stat]. Citado na página 30.

KIRILLOV, A. et al. Panoptic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2019. p. 9404–9413. Citado na página 32.

KORTLI, Y. et al. A novel illumination-invariant lane detection system. In: *2nd International Conference on Anti-cybercrime (ICACC)*. Abha: [s.n.], 2017. Citado na página 39.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: *Proceedings of the Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2012. v. 25, p. 1097–1105. Citado na página 27.

Laboratório de Robótica Móvel - ICMC-USP. *Imagens do Projeto Carina 2*. 2024. <http://lrm.icmc.usp.br/web/index.php?n=Port.ProjCarina2Info>. Citado na página 14.

LATEEF, F.; RUICHEK, Y. Survey on semantic segmentation using deep learning techniques. *Neurocomputing*, v. 338, p. 321–348, Apr 2019. Citado 3 vezes nas páginas 32, 36 e 37.

LATEEF, F.; RUICHEK, Y. Survey on semantic segmentation using deep learning techniques. *Neurocomputing*, v. 338, p. 321–348, Apr 2019. Citado na página 33.

LATEEF, F.; RUICHEK, Y. Survey on semantic segmentation using deep learning techniques. *Neurocomputing*, Elsevier, v. 338, p. 321–348, apr 2019. Citado na página 36.

LEE, C.-Y.; GALLAGHER, P. W.; TU, Z. Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. *arXiv*, v. 1509.08985, n. [cs, stat], oct 2015. ArXiv:1509.08985. Citado na página 23.

LI, R. et al. Adaptive graph convolutional neural networks. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. [S.l.: s.n.], 2018. v. 32, n. 1, p. 1–8. Citado na página 30.

LI, Z. et al. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, p. 1–21, 2021. Citado 3 vezes nas páginas 24, 28 e 29.

LI, Z. et al. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, p. 1–21, 2021. Citado na página 26.

LIANG, D. et al. Lane detection: a survey with new results. *Journal of Computer Science and Technology*, Springer, v. 35, p. 493–505, 2020. Citado 2 vezes nas páginas 13 e 14.

LIANG, M. *Research on Key Technologies of Visual Perception of Intelligent Vehicle Driving Environment*. Tese (Doutorado) — Chang'an University, Xi'an, 2017. Citado na página 37.

LIN, G. et al. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2017. p. 1925–1934. Citado 2 vezes nas páginas 33 e 34.

LIN, M.; CHEN, Q.; YAN, S. Network in network. *arXiv:1312.4400 [cs]*, mar 2014. ArXiv:1312.4400. Citado na página 22.

LIN, Q.; HAN, Y.; HAHN, H. Real-time lane departure detection based on extended edge-linking algorithm. In: *Second International Conference on Computer Research and Development*. Kuala Lumpur: [s.n.], 2010. Citado 2 vezes nas páginas 39 e 40.

LIU, G. *Research on Lane Detection and Tracking Algorithm Based on Image*. Tese (Doutorado) — Hunan University, Changsha, 2014. Citado 3 vezes nas páginas 18, 38 e 39.

LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2015. p. 3431–3440. Citado 2 vezes nas páginas 32 e 33.

LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2015. p. 3431–3440. Citado na página 36.

LU, M.; CAI, Z.; LI, Y. Lane line detection method for road area segmentation. *CAAI Transactions on Intelligent Systems*, v. 5, n. 6, p. 505–509, 2010. Citado 2 vezes nas páginas 37 e 38.

MATSUBARA, V. *Carro Autônomo do Google Causa Primeiro Acidente na Califórnia*. 2024. Figura. Disponível em: <http://quatrorodas.abril.com.br/materia/carro-autonomo-do-google-causa-primeiro-acidente-california>. Citado na página 14.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, Springer, v. 5, n. 4, p. 115–133, 1943. Citado 3 vezes nas páginas 8, 20 e 21.

MICHELI, A. Neural network for graphs: A contextual constructive approach. *IEEE Transactions on Neural Networks*, v. 20, n. 3, p. 498–511, 2009. Citado na página 31.

MODELO, A.; EXEMPLO, P. Adapting neural networks for geometric transformation robustness in computer vision. *Journal of Artificial Intelligence Research*, v. 29, n. 1, p. 101–105, 2024. Citado na página 25.

MULLANI, M. N.; DANDAVATE, A. Semantic texton forests for image categorization and segmentation. *International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE)*, v. 8, n. 4, p. 259–262, Apr 2019. 30 abr. 2019. Citado na página 32.

NEKRASOV, V.; SHEN, C.; REID, I. Light-weight refinenet for real-time semantic segmentation. *arXiv preprint arXiv:1810.03272*, 2018. Citado na página 33.

NOH, H.; HONG, S.; HAN, B. Learning deconvolution network for semantic segmentation. In: IEEE. *2015 IEEE International Conference on Computer Vision (ICCV)*. [S.l.], 2015. p. 1520–1528. Citado 2 vezes nas páginas 35 e 36.

OHTA, Y.; KANADE, T.; SAKAI, T. An analysis system for scenes containing objects with substructures. In: *Proceedings of the Fourth International Joint Conference on Pattern Recognition*. [S.l.: s.n.], 1978. p. 752–754. Citado na página 31.

OLIVEIRA, G. L. et al. Efficient and robust deep networks for semantic segmentation. *The International Journal of Robotics Research*, SAGE Publications Sage UK: London, England, v. 37, n. 4-5, p. 472–491, 2018. Citado na página 14.

POHLEN, T. et al. Full-resolution residual networks for semantic segmentation in street scenes. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2017. p. 4151–4160. Citado na página 33.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: NAVAB, N. et al. (Ed.). *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. [S.l.], 2015. (Lecture Notes in Computer Science, v. 9351). Citado na página 33.

SANDLER, M. et al. Mobilenetv2: Inverted residuals and linear bottlenecks. In: IEEE. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.], 2018. p. 4510–4520. Citado na página 35.

SAS. Inteligência artificial: O que é e qual sua importância. *SAS Insights*, 2023. Disponível em: <https://www.sas.com/>. Citado na página 13.

SCARSELLI, F. et al. The graph neural network model. *IEEE Transactions on Neural Networks*, v. 20, n. 1, p. 61–80, 2009. Citado na página 30.

SHAWAL, S.; SHOYAB, M.; BEGUM, S. Fundamentals of digital image processing and basic concept of classification. *International Journal of Chemical and Process Engineering Research*, v. 1, n. 6, p. 98–108, 2014. Citado na página 18.

SHINDE, P. P.; SHAH, S. A review of machine learning and deep learning applications. In: *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*. [S.l.: s.n.], 2018. p. 1–6. Citado na página 21.

SHIRKE, S.; UDAYAKUMAR, R. Lane datasets for lane detection. In: *International Conference on Communication and Signal Processing (ICCSP)*. Kuala Lumpur: [s.n.], 2019. Citado 2 vezes nas páginas 18 e 39.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. Citado na página 34.

SU, R.; LIU, X.; WANG, L. Convolutional neural network bottleneck features for bi-directional generalized variable parameter hmms. In: *2016 IEEE International Conference on Information and Automation (ICIA)*. [S.l.: s.n.], 2016. Citado na página 22.

TANG, J.; LI, S.; LIU, P. A review of lane detection methods based on deep learning. *Pattern Recognition*, Elsevier, v. 111, p. 107623, 2021. Citado na página 13.

TSAI, C.-C.; LIN, C.-Y.; GUO, J.-I. Dark channel prior based video dehazing algorithm with sky preservation and its embedded system realization for adas applications. *Optics Express*, Optical Society of America, v. 27, n. 9, p. 11877–11901, 2019. Citado na página 39.

VERGARI, A.; MAURO, N. D.; ESPOSITO, F. Visualizing and understanding sum-product networks. *Machine Learning*, Springer, v. 108, n. 4, p. 551–573, 2018. Citado na página 34.

VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2001. p. 511–518. Citado na página 31.

VISIN, F. et al. Renet: A recurrent neural network based alternative to convolutional networks. *arXiv preprint arXiv:1505.00393*, 2015. Citado na página 34.

WADHWA, N. et al. Synthetic depth-of-field with a single-camera mobile phone. *ACM Transactions on Graphics*, v. 37, n. 4, p. 64, 2018. Citado na página 33.

WANG, B.; QI, Z.; MA, G. Robust lane recognition for structured road based on monocular vision. *Journal of Beijing Institute of Technology*, v. 23, n. 3, p. 345–351, 2014. Citado na página 40.

WEILER, M.; CESA, G. General e(2)-equivariant steerable cnns. *arXiv*, v. 1911.08251, apr 2021. ArXiv:1911.08251 [cs, eess]. Citado na página 29.

WEILER, M. et al. 3d steerable cnns: Learning rotationally equivariant features in volumetric data. *arXiv*, v. 1807.02547, oct 2018. ArXiv:1807.02547 [cs, stat]. Citado na página 28.

WILLIAMS, M. *The Drive for Autonomous Vehicles: The DARPA Grand Challenge*. 2024. Figura. Disponível em: <https://herox.com/news/159-the-drive-for-autonomous-vehicles-the-darpa-grand>. Citado na página 14.

XIE, S. et al. Aggregated residual transformations for deep neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2017. p. 1492–1500. Citado 3 vezes nas páginas 27, 28 e 35.

YU, F.; KOLTUN, V. Multi-scale context aggregation by dilated convolutions. *arXiv.org*, 2015. Citado 2 vezes nas páginas 25 e 36.

YU, Z.; WU, X.; SHEN, L. Illumination invariant lane detection algorithm based on dynamic region of interest. *Computer Engineering*, v. 43, n. 2, p. 43–47, 2017. Citado na página 39.

ZEILER, M. D.; FERGUS, R. Visualizing and understanding convolutional networks. In: SPRINGER. *Computer Vision – ECCV 2014*. [S.l.], 2014. p. 818–833. Citado na página 35.

ZHANG, L.; WANG, S.; LIU, B. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, v. 8, n. 4, p. e1253, 2018. Citado 3 vezes nas páginas 21, 34 e 38.

ZHANG, Z. et al. Line detection based on hough one-dimensional transform. *Acta Optica Sinica*, v. 36, n. 4, p. 166–173, 2016. 2016b. Citado 2 vezes nas páginas 23 e 38.

ZHANG, Z. et al. Differentiable learning-to-group channels via groupable convolutional neural networks. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. [S.l.: s.n.], 2019. p. 3542–3551. Citado na página 28.

ZHAO, B. et al. A survey on deep learning-based fine-grained object classification and semantic segmentation. *International Journal of Automation and Computing*, v. 14, n. 2, p. 119–135, 2017. Citado na página 33.

ZHOU, Z. et al. Unet++: A nested u-net architecture for medical image segmentation. In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. [S.l.: s.n.], 2018. p. 3–11. Citado na página 33.

ZHU, X. et al. Deformable convnets v2: More deformable, better results. In: *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2019. p. 9308–9316. Citado na página 26.