Segmentação de imagens de rua utilizando redes neurais convolucionais: um estudo comparativo

1st Hugo Cardoso
Centro de Informática
Universidade Federal de Pernambuco
Recife, Brasil
hac@cin.ufpe.br

2nd Julio Sobral Filho
Centro de Informática
Universidade Federal de Pernambuco
Recife, Brasil
jcasf@cin.ufpe.br

3rd Rodrigo Guimarães Filho
Centro de Informática
Universidade Federal de Pernambuco
Recife, Brasi
rrgf@cin.ufpe.br

***Abstract*—A segmentação de imagens de ruas se tornou um ponto crucial na pesquisa e desenvolvimento de aplicações de visão computacional para soluções que utilizam veículos autônomos. Nesse sentido, o presente trabalho se propõe a realizar um estudo comparativo entre diferentes modelos de redes neurais convolucionais com finalidade de apreciar qual possui melhor desempenho na função de segmentar imagens de ruas.**

Palavras-chave—Convolutional Neural Networks, CNNs, Segmentation, Street Images

I. Introdução

Aplicações autônomas que utilizam processamento e segmentação de imagens em tempo real vem ganhando cada vez mais notoriedade no mercado consumidor. Com isso, a pesquisa e desenvolvimento de soluções que atacam esse problema vem ganhando cada vez mais notoriedade e importância na literatura. A utilização desses sistemas de captura de imagens, processamento e tomada de decisões vem ganhando destaque, com desenvolvimento de carros totalmente autônomos por parte de empresas como a Tesla, e robôs autônomos por parte de empresas como a Boston Dynamics. Além disso, surge a finalidade de realizar a tarefa de segmentação de imagens em diferentes interfaces de hardware e em meios distintos, isto é, redes que devem estar embarcadas e em dispositivos móveis podem ter características e poder computacional distinto, quando comparado a redes que operam em sistemas computacionais robustos, o que reforça a necessidade de traçarmos comparativos com diferentes redes neurais convolucionais que possuem arquiteturas distintas para atacar essas diferentes finalidades.

II. Objetivo

O objetivo deste trabalho é realizar um estudo comparativo entre as redes neurais convolucionais DeepLabV3_resnet101, DeepLabV3_mobilenet_v3_large e U-Net, para segmentação semântica de imagens urbanas utilizando o Cityscapes Dataset. Este dataset é amplamente reconhecido por sua alta qualidade e detalhamento nas anotações de cenas urbanas, contendo diversas categorias como estradas, calçadas, veículos e pedestres.

III. Justificativa

Com esta abordagem, espera-se comparar os diferentes resultados para as diferentes arquiteturas, observando as nuances das redes e as finalidades para as quais cada uma foi desenvolvida, traçando observações e paralelos.

IV. Metodologia

A. Dataset

Os dados foram obtidos a partir do Cityscapes Dataset [1], um dataset formado a partir de registros de vídeos feitos por ruas da Alemanha, que inclui um total de 3.474 imagens de resolução 256 x 512 pixels divididas em dois conjuntos: treinamento e validação. Essas imagens são anotadas em 19 classes, que representam diversos elementos urbanos, incluindo estrada, calçada, construção, vegetação, carro, pedestre, ciclista, entre outros. Cada imagem fornecida possui em seu lado esquerdo a representação original da imagem e em seu lado direito a representação desejada da segmentação. A Figura 1 mostra um exemplo de imagem fornecido pelo dataset.

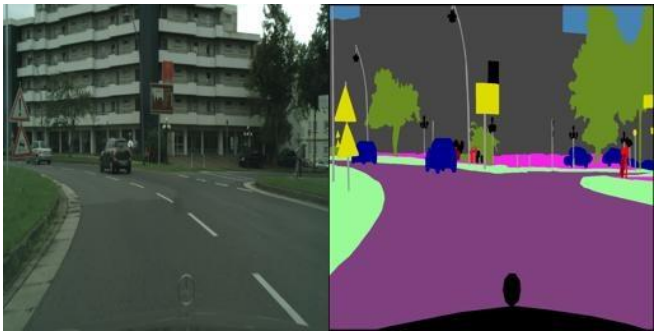


Fig. 1 - Exemplo de imagem fornecido pelo dataset

B. Divisão do Conjunto de Dados

Como foi dito anteriormente, o Cityscape Dataset foi dividido em dois conjuntos principais: treinamento e validação. A distribuição dos dados é a seguinte:

- **Conjunto de treinamento:** Composto por 2975 imagens, representando aproximadamente 85% do total de imagens do dataset. Esse conjunto será utilizado apenas para treinar os modelos de segmentação, permitindo que eles aprendam a partir dos exemplos fornecidos.
- **Conjunto de validação:** Contém 500 imagens, equivalente a 15% do total de imagens. O conjunto de validação é utilizado para avaliar a performance dos modelos.

C. Arquitetura da DeepLabV3_resnet101

A DeepLabV3 [2] minimiza o problema da perda de resolução espacial (spatial resolution loss) ao incorporar

camadas de convolução dilatada, que ampliam o campo de visão da rede sem aumentar o número de parâmetros. Além disso, a DeepLabV3 utiliza um módulo chamado ASPP (Atrous Spatial Pyramid Pooling) que combina features extraídas em múltiplas escalas, permitindo uma melhor captura de contextos variados presentes em cenas urbanas complexas. Uma representação visual da dilatação do kernel de convolução pode ser visto a seguir na Figura 2.

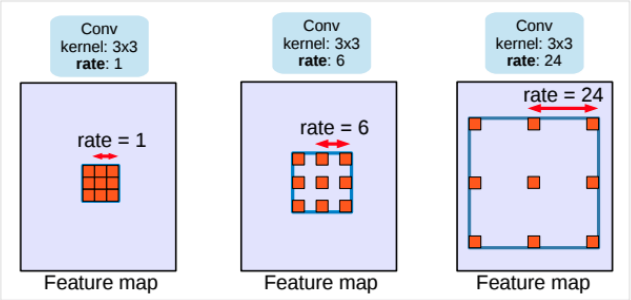


Fig. 2 - Representação visual da dilatação do kernel

A arquitetura da DeepLabV3 é composta por várias camadas convolucionais organizadas em módulos que aplicam convoluções dilatadas com diferentes taxas de dilatação. Um backbone popular para a DeepLabV3 é a ResNet-101, que possui 101 camadas e é usada para extrair features de alta qualidade das imagens de entrada. A ResNet-101, quando utilizada como backbone, também se beneficia dos blocos residuais que minimizam o problema do desvanecimento do gradiente, permitindo o treinamento de redes mais profundas. A Figura 3 traz o diagrama da arquitetura da DeepLabV3.

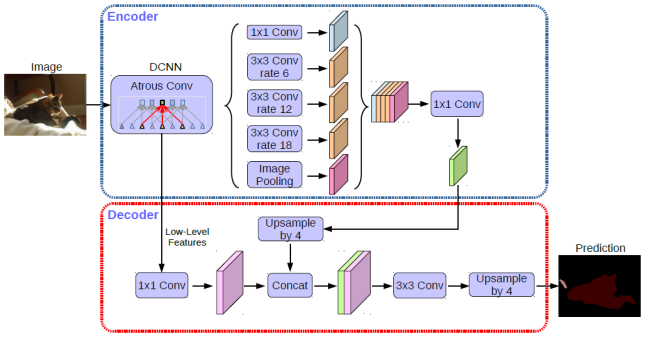


Fig. 3 - Diagrama da arquitetura da DeepLabV3

O módulo ASPP é essencial para a DeepLabV3, pois realiza pooling em múltiplas escalas, combinando as informações para produzir mapas de features robustos. Isso é crucial para a segmentação precisa em diversas condições urbanas presentes no Cityscapes Dataset. A arquitetura também inclui camadas de batch normalization e funções de ativação ELU (Exponential Linear Unit) para melhorar a estabilidade do treinamento e acelerar a convergência.

D. Arquitetura da DeepLabV3_mobilenet_v3_large

A rede DeepLabV3_mobilenet_v3_large [3], que por simplicidade chamaremos simplesmente de mobilenet, possui algumas semelhanças com a arquitetura apresentada anteriormente. Ambas possuem como características a dilatação realizada no kernel para ampliar o campo de visão do modelo e apreciar mais detalhes da imagem. Entretanto, como diferencial a arquitetura da mobilenet possui uma otimização para ser executada em em CPUs de aparelhos celulares. O ganho real com o desenvolvimento deste tipo de rede é a capacidade que que dados podem ser avaliados nos próprios dispositivos móveis, sem que seja necessário o envio desses dados para um servidor e posterior obtenção de resposta, além de prover um melhor desempenho com relação à eficiência energética e consumo de bateria. Na Figura 4 pode-se observar as características da arquitetura da mobilenet.

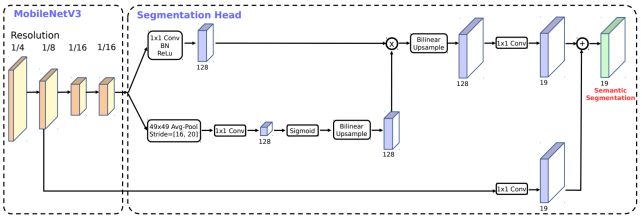


Fig. 4 - Arquitetura da mobile net

E. Arquitetura da U-Net

A arquitetura U-Net [4] pode se tornar uma boa escolha para tarefas de segmentação semântica devido às suas capacidades em segmentar imagens com alta precisão, mesmo quando os dados de treinamento são limitados. Esse modelo consegue resolver esse tipo de problema utilizando uma arquitetura encoder-decoder simétrica, a qual preserva informações detalhadas das imagens enquanto melhora a generalização.

Como dito antes, a arquitetura U-Net é separada em duas partes: encoder e decoder. O encoder é responsável por extrair features das imagens através de sucessivas camadas de convolução e pooling, que reduzem a resolução da imagem, porém aumenta a profundidade das features. Já no decoder, fica responsável por reconstruir a imagem segmentada, trazendo ela para a resolução original através de camadas de upsampling e convoluções adicionais.

Outro aspecto importante da U-Net é o uso de skip connections entre as camadas de encoder e decoder. Essas conexões garantem que informações de alta resolução das camadas iniciais sejam diretamente utilizadas nas camadas correspondentes de decoder, assim melhorando a segmentação.

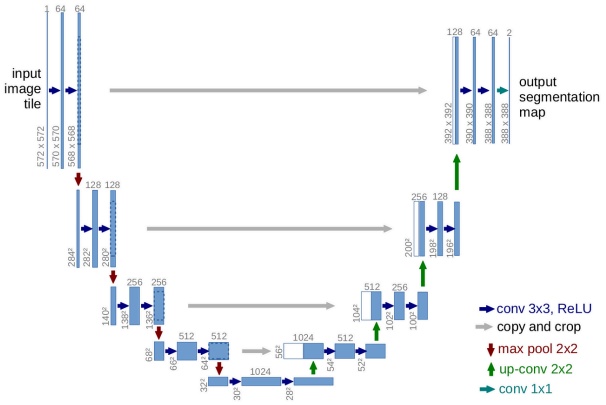


Fig. 5 - Arquitetura da U-Net

F. Avaliação dos Modelos

Para realizarmos a avaliação dos modelos e, consequentemente, fazer a comparação entre eles, utilizaremos diferentes métricas:

- **Training Loss:** Reflete o quão bem o modelo está aprendendo os dados de treinamento. Quando o valor de training loss está baixo, indica que o modelo está se ajustando bem aos dados. Porém, caso o training loss continue a abaixar e o validation loss comece a aumentar, isso pode indicar overfitting no modelo.
- **Validation Loss:** Indica o quão bem o modelo está generalizando para dados que não foram vistos durante o treinamento. Um baixo valor de validation loss, sugere que o modelo está conseguindo fazer previsões mais precisas para novos dados.
- **Pixel Accuracy:** Métrica de avaliação em tarefas de segmentação que indica a porcentagem de pixels classificados corretamente. É uma métrica interessante para calcular o desempenho geral de um modelo em tarefas de segmentação.

É importante informar que não fomos capazes de utilizar outras métricas que foram propostas anteriormente, como Iou e mIoU, devido à ausência das informações sobre as classes dos objetos nas imagens do dataset.

G. Treinamento dos Modelos

Para o treinamento dos modelos citados anteriormente, cada modelo foi submetido a uma etapa de treinamento que consistiu de vinte e cinco épocas sobre o dataset de treinamento. Todos os modelos foram submetidos à mesma divisão do dataset de treinamento e de teste, não sendo o dataset de treinamento embaralhado após cada treinamento, garantindo que cada etapa de treino será igual para cada rede. Após isso o modelo treinado foi submetido a uma análise sobre o dataset de teste, onde pode-se avaliar o Mean Square Error sobre o dataset de validação, bem como o Average Pixel Accuracy do modelo sobre o dataset de validação, além disso, o valor de MSE do modelo sobre o dataset de treino foi levado em consideração. Para todos os modelos foi utilizado a função de custo Mean Square Error (MSE) e foi utilizado o algoritmo ADAM para a otimização, utilizando o valor de 0.01 de learning rate para todos os modelos. O treinamento foi realizado em batches de vinte imagens por batch para todos os modelos.

H. Modificações na arquitetura

As redes DeepLabV3_resnet101 e a DeepLabV3_mobilenet_v3_large sofreram modificações na arquitetura para a utilização na aplicação específica. Ambas as redes fornecem como saída um tensor com vinte e um canais, os quais podem ser mapeados para um pool de classes, cada classe correspondendo a um valor de pixel desejado para classificação, assim realizando a segmentação da imagem, adotando o valor da classe com maior probabilidade de representar aquele pixel na máscara final. Entretanto, por não possuímos os valores exatos dos pixels de cada classe da máscara, optamos por modificar a saída da rede. Foi adicionado uma camada convolucional final à rede, que realiza a conversão da saída de vinte e um canais para três canais, cada canal representando um valor correspondente a uma das três cores principais: vermelho, azul e verde (RGB). Dessa forma, a rede fornece valores intermediários entre as classes desejadas pela saída referenciada, o que torna o treinamento não tão eficiente quando comparado com o uso direto dos valores de classe desejados pela máscara.

Já na rede U-Net, foi implementado uma arquitetura básica do modelo, sem a presença de adaptações relevantes para a tarefa de segmentação semântica.

V. Resultados

A. DeepLabV3_resnet101

Abaixo pode-se observar, na Figura 6, alguns resultados de máscaras obtidas, a partir das respectivas imagens de entrada e máscaras desejadas como saída:

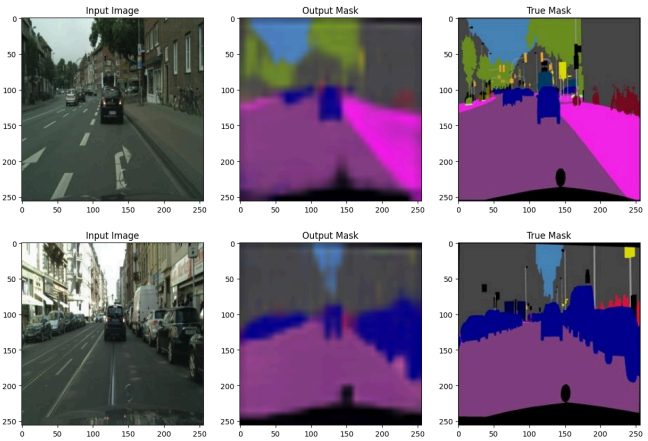


Fig. 6 - Exemplos de máscaras obtidas pela DeepLabV3_resnet101

Como podemos observar, as máscaras obtidas conseguem capturar bem objetos grandes e que possuem contornos bem definidos na imagem, como carros, vegetação , a rua, o capô do carro que está realizando as imagens e o céu. Entretanto, objetos como placas, semáforos e pessoas não são bem capturados nos casos em que não aparecem com nitidez nas imagens, de tal forma que o modelo acaba classificando estes objetos em classes que possuem uma maior ocorrência nas imagens. Essa característica poderia ser reduzida caso a rede pudesse realizar o treinamento por maiores períodos de tempo, ou caso usássemos as categorias pré-definidas de pixels nas máscaras desejadas como saída.

Quando falamos de resultado quantitativos para esta rede, ela obteve os seguintes resultados para as métricas escolhidas para comparação entre as redes:

- Train Loss: 0.013087
- Validation Loss: 0.041533
- Average Pixel Accuracy: 16.32%

B. DeepLavV3_mobilenet_v3_large

A seguir pode-se observar, na Figura 7, alguns resultados de máscaras obtidas, a partir das respectivas imagens de entrada e máscaras desejadas como saída:

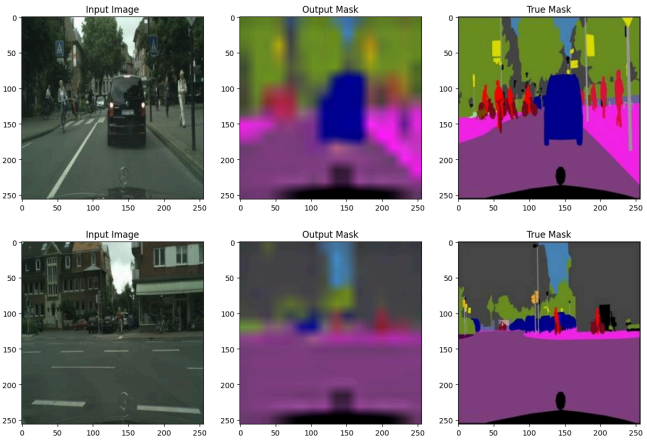


Fig. 7 - Exemplos de máscaras obtidas pela DeepLabV3_mobilenet_v3_large

Podemos observar que a rede conseguiu também, como observado no exemplo anterior, que a rede foi capaz de capturar de bem objetos maiores e que possuem cores e contornos bem definidos nas imagens. Também foi possível observar que esta rede, diferentemente da anterior, conseguiu capturar de melhor forma as classes que são menores nas imagens, e que não são tão comuns nas imagens, como pessoas, placas e semáforos. Esse fato pode ser entendido pelo motivo de que a rede mobilenet possui menos parâmetros quando comparada à resnset_101, o que provavelmente proporciona uma convergência mais rápida quando comparada à outra rede. Para que a resnet_101 conseguisse trazer essa capacidade de identificar de melhor maneira as demais classes na imagem, seria necessário fornecer mais tempo ao seu treinamento.

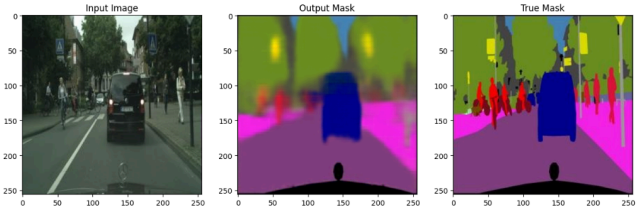
Outro ponto que pode ser observado é o de que a mobilenet, também pelo fato de ser uma rede menor quando comparada com a resnet_101, ele não traz uma nitidez tão boa nos contornos dos objetos nas máscaras quando comparada com a resnet_101, isso pode ser identificado visualmente quando observamos o contorno do capô do carro que está retirando as imagens entre as duas redes, isso também se verifica nas classes maiores como carros, edifícios, ruas e vegetação.

Apresentando os resultados quantitativos para esta rede, ela obteve os seguintes resultados para as métricas escolhidas para comparação:

- Train Loss: 0.012687
- Validation Loss: 0.043556
- Average Pixel Accuracy: 16.94%

C. U-Net

Logo abaixo, na Figura 8, podemos observar duas máscaras obtidas como resultados do modelo U-Net:



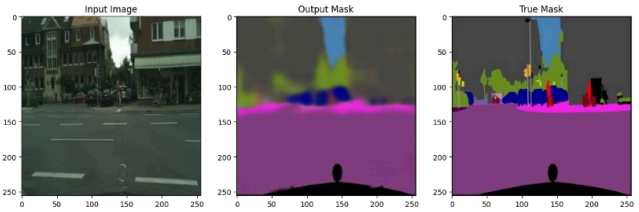


Fig. 8 - Exemplos de máscaras obtidas pela U-Net

Observamos que o modelo U-Net se destaca na identificação de objetos maiores, como carros, ruas, calçadas e árvores, com boa precisão nas bordas. Isso é possível graças às conexões de atalho (skip connections) presentes em sua arquitetura, que ajudam a preservar detalhes importantes durante o processo de segmentação. No entanto, a performance do modelo é menos satisfatória quando se trata de objetos menores, como placas de trânsito e semáforos. Apesar disso, os resultados do U-Net superam os dos outros modelos comparados (DeepLabV3_resnet101 e DeepLavV3_mobilenet_v3_large) devido à maior nitidez das máscaras geradas. Essa vantagem provavelmente decorre da capacidade da U-Net preservar detalhes.

A seguir, apresentamos os resultados quantitativos para a rede, com base nas métricas selecionadas para comparação:

- Train Loss: 0.008181
- Validation Loss: 0.012566
- Pixel Accuracy: 69.52%

Conclusão

A análise comparativa dos modelos DeepLabV3 (com ResNet-101 e MobileNetV3-Large) e U-Net destaca diferenças notáveis em suas capacidades de segmentação semântica. O DeepLabV3 com ResNet-101 demonstra eficiência na identificação de objetos grandes e bem definidos, como carros e vegetação, mas enfrenta dificuldades com objetos menores e menos nítidos, como placas e semáforos. A adição de tempo de treinamento poderia melhorar esses resultados. Por outro lado, o DeepLabV3 com MobileNetV3-Large mostra uma performance superior na detecção de classes menores, embora sua capacidade de capturar detalhes finos seja inferior à da ResNet-101. Em contraste, o modelo U-Net se destaca tanto pela sua excelente nitidez na segmentação de objetos maiores, graças às suas conexões de atalho que preservam detalhes importantes, quanto em sua eficácia para a detecção de objetos menores, quando comparado com os outros dois modelos. Em termos quantitativos, o U-Net supera os outros modelos com a menor perda de treinamento e validação, além de uma maior precisão média dos pixels, revelando-se o melhor modelo geral para a tarefa de segmentação.

Referências

[1] Becker, Dan (2018, abril) Cityscapes Image Pairs obtido em 2024, junho de https://www.kaggle.com/datasets/dansbecker/cityscapes-image-pairs

[2] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation,” in Lecture notes in computer science, 2018, pp. 833–851. doi: 10.1007/978-3-030-01234-2_49

[3] A. Howard et al., “Searching for MobileNetV3,” IEEE/CVF International Conference on Computer Vision (ICCV), Oct. 2019, doi: 10.1109/iccv.2019.00140

[4] O. Ronneberger, P. Fischer, and T. Brox, “U-NET: Convolutional Networks for Biomedical Image Segmentation,” in Lecture notes in computer science, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28