1.03.03 - Metodologia e Técnicas da Computação.

UM ESTUDO COMPARATIVO DE MÉTODOS DE DEEP LEARNING APLICADOS À SEGMENTAÇÃO SEMÂNTICA DE OBSTÁCULOS, ZONAS SEGURAS E NÃO SEGURAS PARA NAVEGAÇÃO A PARTIR DE DADOS RGB-D.

Felipe M. Barbosa[1], Fernando S. Osório[2]
1. Estudante do Instituto de Ciências Matemáticas e de Computação da Univ. de São Paulo (ICMC-USP)
2. Professor do ICMC-USP - Departamento de Sistemas de Computação/Orientador

Resumo
Os sistemas de Visão Computacional, por meio de técnicas como a segmentação semântica dos elementos presentes nas imagens, atuam na interface entre robôs móveis autônomos e o ambiente, fornecendo meios para que os mesmos naveguem de forma segura. Com o advento das Redes Neurais Convolucionais, e a maior acessibilidade a dados 3D, foram propostas novas técnicas de segmentação semântica baseadas em *Deep Learning*. A literatura apresenta estudos comparativos desses modelos, baseando-se principalmente no processamento de imagens 2D. Mesmo os trabalhos que utilizam dados 3D, não consideram em suas análises arquiteturas derivadas de Redes GAN. Dessa forma, o presente projeto objetivou o estudo, implementação e comparação dos modelos FCN, SegNet e Pix2Pix (GAN), aplicados à segmentação semântica de obstáculos, zonas seguras e não seguras à navegação de um robô móvel autônomo, a partir de imagens RGB-D. Os experimentos posicionam o modelo Pix2Pix, baseado em GAN, como o mais adequado à tarefa.

Palavras-chave: Visão Computacional; Inteligência Artificial; Robôs Móveis Autônomos.

Apoio financeiro: CNPq (Programa Institucional de Bolsas de Iniciação Científica- PIBIC).

Trabalho selecionado para a JNIC: Universidade de São Paulo (USP).

Introdução
Veículos autônomos são dispositivos robóticos móveis desenvolvidos visando auxiliar motoristas na condução de veículos, facilitar a mobilidade de pessoas com necessidades especiais e melhorar a segurança no trânsito. Sendo assim, têm importância central no futuro da mobilidade urbana.

Apesar dos avanços feitos nos últimos anos, ainda há casos de acidentes envolvendo tais dispositivos, alguns com graves consequências [1-2].

Os sistemas de Visão Computacional têm um papel essencial na interface entre o robô e o ambiente, fornecendo técnicas para auxílio à percepção de seu entorno e contribuindo para a segurança de motoristas, passageiros e usuários vulneráveis nas vias - pedestres e ciclistas. Uma dessas técnicas é a segmentação semântica, que consiste na classificação da imagem a nível de pixel. Essa característica permite, além de classificar uma dada região da imagem, definir de forma precisa a sua posição e formato.

Os avanços na área de *Deep Learning* fomentaram o desenvolvimento de novos métodos de segmentação semântica, como as FCN (*Fully Convolutional Networks*) [3] e a SegNet [4]. Atualmente, porém, modelos baseados em arquiteturas GAN (*Generative Adversarial Networks*) [5] têm se mostrado poderosos em tarefas de tradução imagem-a-imagem.

O maior acesso a dados 3D é outro fator de destaque nesse contexto. Um exemplo são as imagens RGB-D (RGB + Depth) que, por encapsularem informação de profundidade em sua estrutura de canais de cor, permitem aprimorar métodos originalmente baseados em imagens 2D [6-7].

A literatura atual apresenta diversos estudos comparativos de métodos de segmentação semântica, a maior parte baseada em imagens 2D, e poucos em imagens RGB-D. Contudo, em nenhum dos casos são considerados modelos baseados em arquiteturas GAN.

Assim, o presente projeto tem por objetivo o estudo, implementação e comparação de métodos de *Deep Learning* para a segmentação semântica de obstáculos, zonas seguras e não-seguras em dados RGB-D.

Como principal contribuição tem-se a inclusão de um modelo baseado em arquitetura GAN na análise, denominado Pix2Pix [8]. Busca-se avaliar o desempenho de uma arquitetura nova, poderosa e versátil em tarefas de tradução imagem-a-imagem, frente a modelos amplamente aceitos e utilizados como referência na literatura – FCN e SegNet.

Metodologia
Inicialmente, foi realizado o estudo da literatura atual na área. Observou-se que os modelos FCN e SegNet são amplamente utilizados como referência e que a literatura carece de análises considerando modelos baseados em arquiteturas GAN.

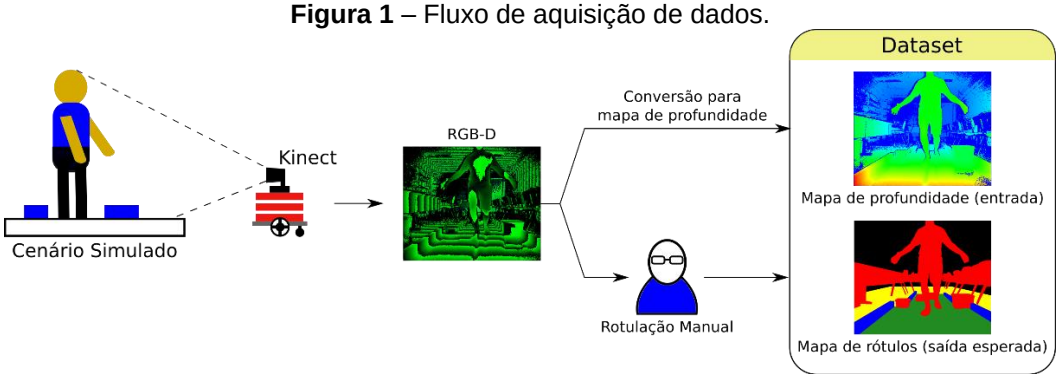
O presente projeto, além implementar os modelos FCN e SegNet, inclui nas análises o modelo Pix2Pix (*Conditional GAN*) que, em estudos anteriores, se mostrou eficiente em tarefas de segmentação [9].

A metodologia experimental consistiu nas etapas de criação da base de dados, implementação/treinamento dos modelos e avaliação dos resultados (seção seguinte).

Ao invés de utilizar uma base de dados RGB-D publicamente disponível, os modelos foram treinados e avaliados em uma base de dados criada para o escopo do projeto. Isso foi feito com o objetivo de simular, em escala (considerando um robô móvel autônomo de médio porte), elementos presentes no ambiente de navegação de um veículo autônomo, como a via navegável, via não navegável, bordas e obstáculos.

Para esse fim, foram criados cenários em ambiente *indoor* (Laboratório do Espaço Maker – EngComp USP São Carlos) com diversas configurações de vias, posições e formatos dos obstáculos. As 562 imagens RGB-D resultantes, coletadas com um sensor Kinect V2, seguiram dois fluxos de processamento, um referente à sua rotulação e o outro à sua conversão em mapas de profundidade (Figura 1).

Figura 1 – Fluxo de aquisição de dados.



Fonte: Elaborada pelo autor.

O processo de rotulação, realizado com a ferramenta LabelMe [10], foi necessário para fornecer aos modelos, além da entrada, a saída esperada (mapa de rótulos) - aprendizado supervisionado.

Do conjunto de dados original, 415 imagens foram reservadas para treinamento, 47 para validação e 100 para teste dos modelos. Tais dados passaram, ainda, por transformações como translações, inversões e rotações – processo de *data augmentation*, para ampliação do número de exemplos.

A configuração da máquina utilizada no desenvolvimento é descrita na Tabela 1. As tecnologias utilizadas foram Python [11], OpenCV [12], Tensorflow [13] e Keras [14].

Tabela 1 – Especificações da máquina utilizada no desenvolvimento.

Modelo	Acer Nitro 5
Processador	Intel Core i5-8300H
Memória RAM	8GB
GPU	NVIDIA GeForce 1050
Sistema Operacional	Windows 10

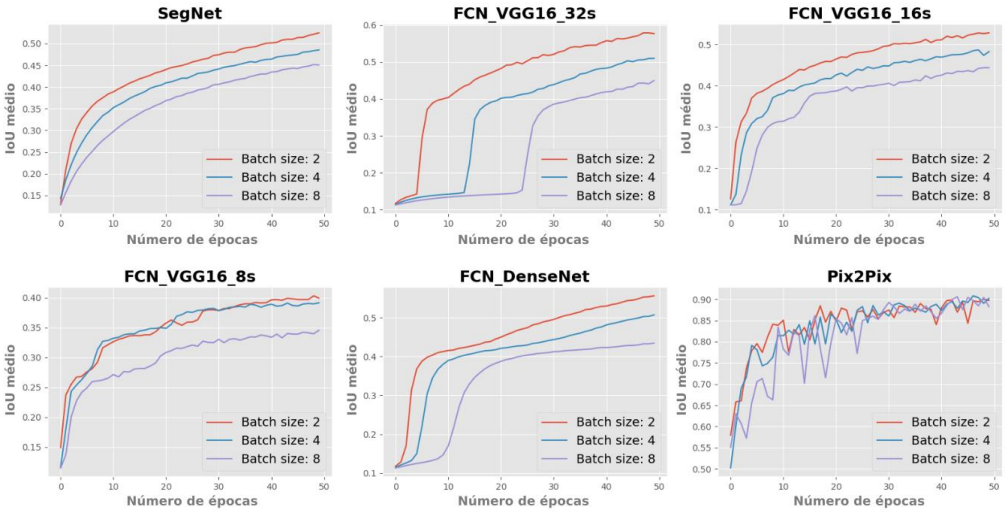
Fonte: Elaborada pelo autor.

A ideia inicial era avaliar o desempenho dos modelos com e sem o uso de transferência de aprendizado. Porém, por não ter sido encontrada uma implementação do modelo Pix2Pix com pesos pré-calibrados, todos os modelos foram treinados de forma completa para garantir uma comparação justa.

A avaliação dos modelos foi feita com base no valor médio da Intersecção sobre a União (IoU médio), número de parâmetros, eficiência, tempo de inferência e inspeção visual.

Como etapa prévia às análises, foi avaliada a influência do parâmetro *batch size* no aprendizado neural, com relação à evolução do valor de IoU médio. Foram treinadas, por 50 épocas e com otimizador SGD (parâmetros padrão), variantes com 2, 4 e 8 imagens por *batch* (Figura 2). As variantes treinadas com *batch size* 2 (melhor desempenho) foram utilizadas na etapa de testes - resultados discutidos na seção seguinte.

Figura 2 – Evolução do aprendizado neural, com relação ao IoU médio, para diferentes valores de *batch size*.



Fonte: Elaborada pelo autor.

Resultados e Discussão

A Intersecção sobre União mede o nível de semelhança entre a segmentação gerada e a esperada. Seu valor, para uma dada classe, pode ser obtido pela divisão do número de pixels classificados corretamente (intersecção) pelo número de pixels da união das regiões pertencentes àquela classe em ambas as imagens. Como consideramos mais de uma classe no problema, tomamos o valor IoU médio. A Tabela 2 exibe a média e o desvio padrão do valor IoU médio para as segmentações geradas no conjunto de teste.

Tabela 2 - IoU médio e tempo de inferência por modelo.

Modelo	IoU Médio		Tempo de Inferência
	Média	Desvio Padrão	
Pix2Pix	**0.90**	0.073	0.183
FCN VGG16 32s	0.61	0.101	**0.049**
FCN VGG16 16s	0.58	0.091	0.055
SegNet	0.53	0.113	0.067
FCN DenseNet	0.52	0.082	0.054
FCN VGG16s 8s	0.42	0.062	0.059

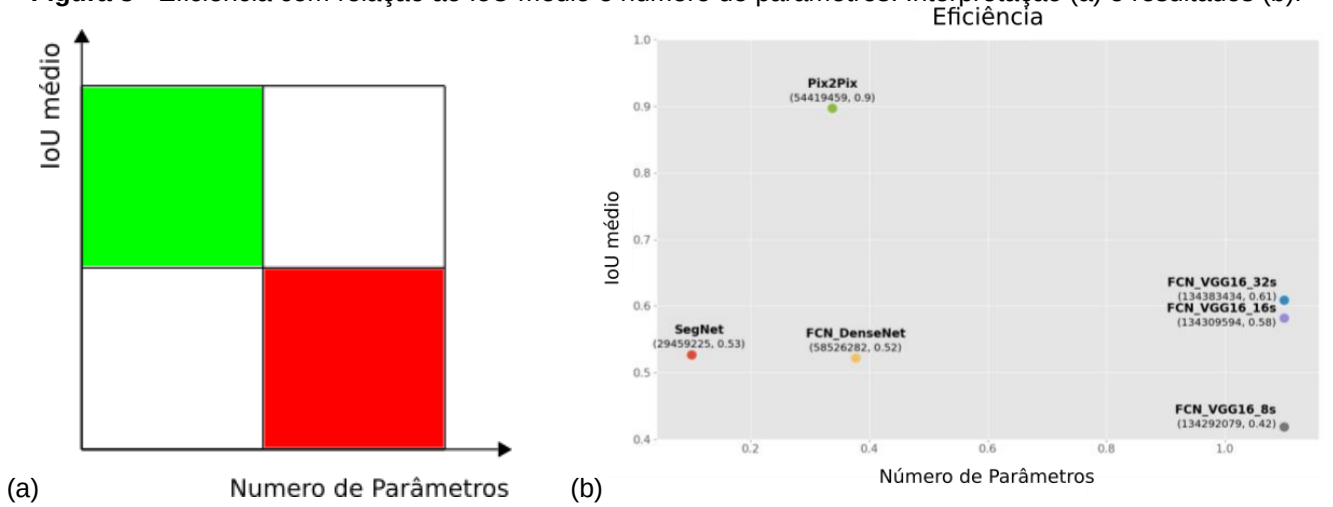
Fonte: Elaborada pelo autor.

O modelo Pix2Pix apresentou os melhores resultados, com desempenho cerca de 30% superior à segunda melhor configuração. Essa diferença pode ser atribuída a questões relacionadas à sua arquitetura de rede, mais recente, versátil e com maior poder computacional.

No tocante ao tempo de inferência (Tabela 2), o modelo FCN VGG16 32s foi superior. Nesse critério, o modelo Pix2Pix teve o pior desempenho, sendo cerca de 4 vezes mais lento que o modelo FCN VGG16 32s.

A análise da eficiência foi feita com base no valor médio de Intersecção sobre União e no número de parâmetros dos modelos. De forma resumida, quanto maior o IoU médio e menor o número de parâmetros do modelo, mas eficiente ele é na tarefa considerada – região verde da Figura 3a. De forma equivalente, um baixo IoU médio e um alto número de parâmetros caracterizam modelos pouco eficientes – região vermelha da Figura 3a. A Figura 3b apresenta os resultados referentes à análise da eficiência dos modelos.

Figura 3 - Eficiência com relação ao IoU médio e número de parâmetros. Interpretação (a) e resultados (b).

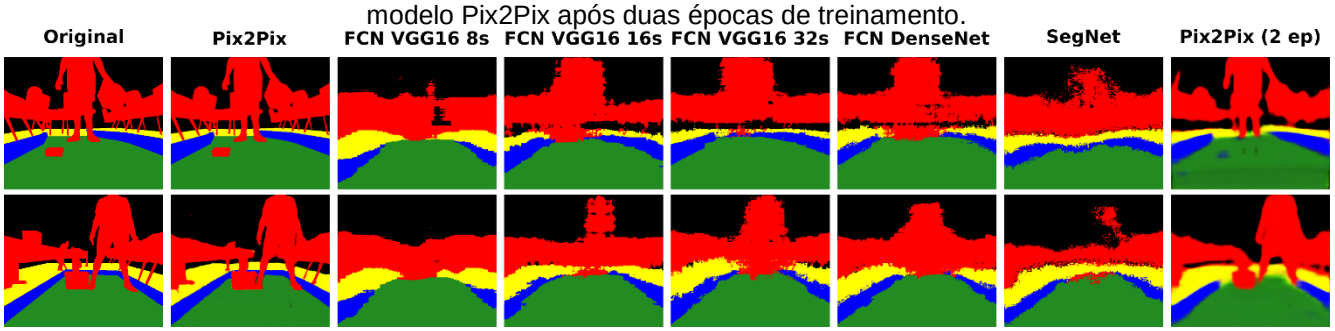


Fonte: Elaborada pelo autor.

Podemos observar que o modelo Pix2Pix, apesar de apresentar um número de parâmetros cerca de 45% superior ao modelo SegNet, tem um IoU médio 37% superior ao mesmo. Dessa forma, o modelo Pix2Pix foi selecionado como aquele com melhor desempenho segundo o critério de eficiência.

Finalmente, foi realizada a inspeção visual dos resultados (Figura 4). Apesar de ser um critério subjetivo, é o que dá a melhor intuição do desempenho dos modelos, por permitir avaliar visualmente a qualidade das segmentações; além disso, permite validar os resultados obtidos nas análises objetivas.

Figura 4 – Inspeção visual das segmentações. A última coluna corresponde aos resultados gerados pelo modelo Pix2Pix após duas épocas de treinamento.



Fonte: Elaborada pelo autor.

De forma geral, todos os modelos conseguiram segmentar as regiões navegáveis, bordas e regiões não navegáveis. Contudo, apenas o modelo Pix2Pix delineou bem os contornos detalhados dos obstáculos. Assim, a inspeção visual valida os resultados anteriores, que apontam o modelo Pix2Pix como o mais preciso.

Um fato interessante é que o modelo Pix2Pix, mesmo nas primeiras épocas de treinamento, já apresentava resultados muito superiores ao desempenho final dos demais modelos.

Em [8] o desempenho do modelo Pix2Pix em tarefas de segmentação é dito insatisfatório, pois foi projetado gerar saídas com alto nível de detalhe, o que não é o caso dos mapas de rótulos. Contudo, a análise realizada deixa clara sua superioridade frente aos demais modelos, referências em segmentação semântica.

Por fim, o tamanho reduzido da base de dados gerou a preocupação com a ocorrência de *overfitting* no modelo Pix2Pix. Porém, dada a qualidade apresentada já nas primeiras épocas de treinamento, o uso de cenas inéditas (conjunto de teste) não degradar seu desempenho e, ainda, a dificuldade de aprendizado dos demais modelos, atribuiu-se o desempenho obtido a avanços arquiteturais apresentados pelo modelo.

Conclusões

Os resultados obtidos posicionam o modelo Pix2Pix como aquele que melhor atendeu aos critérios avaliados. Ele apresenta o melhor valor de IoU médio, maior eficiência e melhor qualidade de segmentações. Sendo assim, é o mais adequado para uso em um veículo autônomo, em escala.

Cabe observar, contudo, que o tempo de inferência é um fator crucial em sistemas autônomos, com impacto direto na velocidade da tomada de decisão. Quanto menor o tempo de inferência, mais tempo há para o planejamento e atuação no sentido da prevenção de situações de risco no trânsito. Dessa forma, a diminuição do tempo de inferência do modelo Pix2Pix seria uma contribuição futura de grande importância.

O presente projeto teve por objetivo avaliar o desempenho dos modelos em ambiente *indoor* controlado. Em aplicações reais, contudo, o sensor Kinect utilizado sofreria influência dos níveis de iluminação e de possíveis oclusões - poeira ou neblina, por exemplo. Além disso, seria necessário um maior alcance e melhor qualidade dos dados capturados. Um exemplo de sensor que mitiga as limitações anteriores, sob a pena de um custo elevado, é o LIDAR, muito utilizado em projetos de veículos autônomos.

Feitas as devidas considerações, a principal contribuição do presente projeto foi demonstrar quão poderosos, versáteis e eficientes são os modelos baseados em arquiteturas GAN, podendo ter papel central em futuros avanços relacionados à visão computacional, não só em veículos autônomos, mas em quaisquer áreas em que a segmentação semântica possa ser aplicada.

Referências bibliográficas

[1] Phil McCausland. 2019. Self-driving Uber car that hit and killed woman did not recognize that pedestrians jaywalk – NBC. Disponível em: https://nbcnews.to/3ojSC1U

[2] BBC News. 2020. Tesla Autopilot crash driver 'was playing video game' – BBC. Disponível em: https://bbc.in/3qqPYcz

[3] Jonathan Long, Evan Shelhamer, Trevor Darrell. 2015. Fully Convolutional Networks for Semantic Segmentation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 7-12, 2015 Boston, MA. IEEE, 3431-3440. https://doi.org/10.1109/CVPR.2015.7298965

[4] Vijay Badrinarayanan, Alex Kendall and Roberto Cipolla. 2016. SegNet: A Deep Convolutional Encoder Decoder Architecture for Image Segmentation. Disponível em: https://arxiv.org/abs/1511.00561 (v3)

[5] Ian J. Goodfellow et al. 2014. Generative Adversarial Nets. arXiv:1406.2661. Disponível em: https://arxiv.org/abs/1406.2661

[6] F. Fooladgar and S. Kasaei. 2020. A survey on indoor RGB D semantic segmentation: from hand crafted features to deep convolutional neural networks. Multimedia Tools and App. 79. DOI: 10.1007/s11042 019 7684 3

[7] Seichter et al. 2020. Efficient RGB D Semantic Segmentation for Indoor Scene Analysis. Disponível em: https://arxiv.org/abs/2011.06961

[8] Phillip Isola et al. 2018. Image to Image Translation with Conditional Adversarial Networks. Disponível em: https://arxiv.org/abs/1611.07004 (v3)

[9] BARBOSA, F. M.; OSÓRIO, F. S. Análise de Dados 3D visando a detecção de Zonas Seguras/Não Seguras para Navegação de Robôs Móveis Autônomos. Universidade de São Paulo. São Carlos, p. 1-30. 2019.

[10] Kentaro Wada. 2016. labelme: Image Polygonal Annotation with Python. Disponível em: https://bit.ly/3l3brWy

[11] Python. 2020. Welcome to Python.org. Disponível em: https://www.python.org/

[12] OpenCV. 2020. OpenCV. Disponível em: https://opencv.org/

[13] Tensorflow. 2020. Uma plataforma completa de código aberto para machine learning. Disponível em: https://www.tensorflow.org/

[14] Tensorflow Core. 2020. Keras. Disponível em: https://bit.ly/37tUdvo