Aplicando KittiSeg da MultiNet com dados da IARA

Vinicius Cardoso
Departamento de Informática
Universidade Federal do Espírito Santo (UFES)
Vitória – ES, Brasil

Lucas Oliveira
Departamento de Informática
Universidade Federal do Espírito Santo (UFES)
Vitória – ES, Brasil

***Abstract*—O problema de detecção de pista e lanes tem se popularizado com a evolução do desenvolvimento de veículos autônomos e também devido a revolução na área de *deep learning*, iniciada com a introdução da AlexNet em 2012. Neste artigo, foi utilizada uma arquitetura capaz de realizar a segmentação de pista usando imagem monocular de forma eficiente utilizanda a rede neural profunda MultiNet. Os datasets utilizados foram colhidos pela Inteligent Autonomous Robotic Automobile (IARA) e os resultados foram promissores, mostrando que a capacidade de generalização da rede em imagens não conhecidas.**

Keywords—segmentação de pista; deep learning; redes neurais convolucionais; carros autônomos;

I. INTRODUÇÃO

O problema de detecção de pista e lanes tem se popularizado com a evolução do desenvolvimento de veículos autônomos e também devido a revolução na área de *deep learning*, iniciada com a introdução da AlexNet em 2012 [2]. Desde então, diversas novas abordagens têm surgido a cada dia e cada vez mais precisas. Em parte, isso se dá por causa do aumento da disponibilidade de dados, do aumento do poder computacional e do desenvolvimento de novos algoritmos.

No artigo estudado foi proposta uma arquitetura de rede que pode executar classificação, detecção e segmentação semântica de forma eficiente e simultânea (menos de 100 ms para realizar uma inferência). A eficácia da rede foi verificada no *Kitti Vision Benchmark* [3] e provou que seu desempenho na segmentação de pista era o estado da arte.

Para a realização dos experimentos utilizamos imagens extraídas de *logs* gerados pela IARA (*Intelligent Autonomous Robotic Automobile*), o carro autônomo desenvolvido pelo Laboratório de Computação de Alto Desempenho (LCAD) da UFES.

II. TRABALHOS CORRELATOS

Nesta seção descreveremos alguns dos trabalhos analisados que foram submetidos ao *Kitti Vision Benchmark* [3], na categoria *Road/Lane Detection*.

A. Up Convolutional Network [Up-Conv-Poly]

Este artigo aborda o problema da segmentação de pistas em imagens RGB convencionais, explorando avanços recentes na segmentação semântica através de redes neurais convolutivas (CNNs). As redes de segmentação são muito grandes e consomem uma quantidade de recursos computacionais considerável, o que torna a execução em tempo real complicada. Para tornar esta técnica aplicável à robótica, os autores propuseram vários refinamentos de arquitetura que proporcionam um melhor *trade-off* entre a qualidade da segmentação e o tempo de execução. Isto é conseguido por um novo mapeamento entre classes e filtros no lado da expansão da rede. A rede é treinada de ponta a ponta e produz predições da estrada / pista na resolução de entrada original em cerca de 50ms. [5]

A rede apresenta boa precisão, no conjunto de dados KITTI para a segmentação de pista. Mas, apesar de o código ter sido disponibilizado não conseguimos utilizar a rede ou replicar o experimento.

B. Deep FCN with Random Data Augmentation for Enhanced Generalization in Road Detection [DEEP-DIG]

Neste artigo, um sistema Deep Learning para detecção de estradas é proposto usando a rede *ResNet-101* com uma arquitetura completamente convolucional e várias etapas de *upscaling* para interpolação de imagem. É demonstrado que os significativos ganhos de generalização no processo de aprendizagem são alcançados com a geração de dados aleatórios de treinamento usando várias transformações geométricas e mudanças de *pixelwise*, tais como transformações de perspectiva, espelhamento, corte de imagem, distorções, desfocagem, ruído e mudanças de cor. Além disso, este artigo mostra que o uso de uma estratégia de *upscaling* de 4 passos oferece ótimos resultados de aprendizagem em comparação com outras técnicas similares que realizam o *upscaling* de dados com base em camadas superficiais com escassa representação dos dados da imagem. O sistema é treinado e testado com dados KITTI e, além disso, também é testado em imagens gravadas no Campus da Universidade de Alcala (Espanha). A melhora alcançada após o aumento de dados e a realização de variantes de treinamento é realmente encorajadora, mostrando o caminho a seguir para a generalização de aprendizagem aprimorada de sistemas de detecção de estradas com vista à implantação real em carros autônomos. [6]

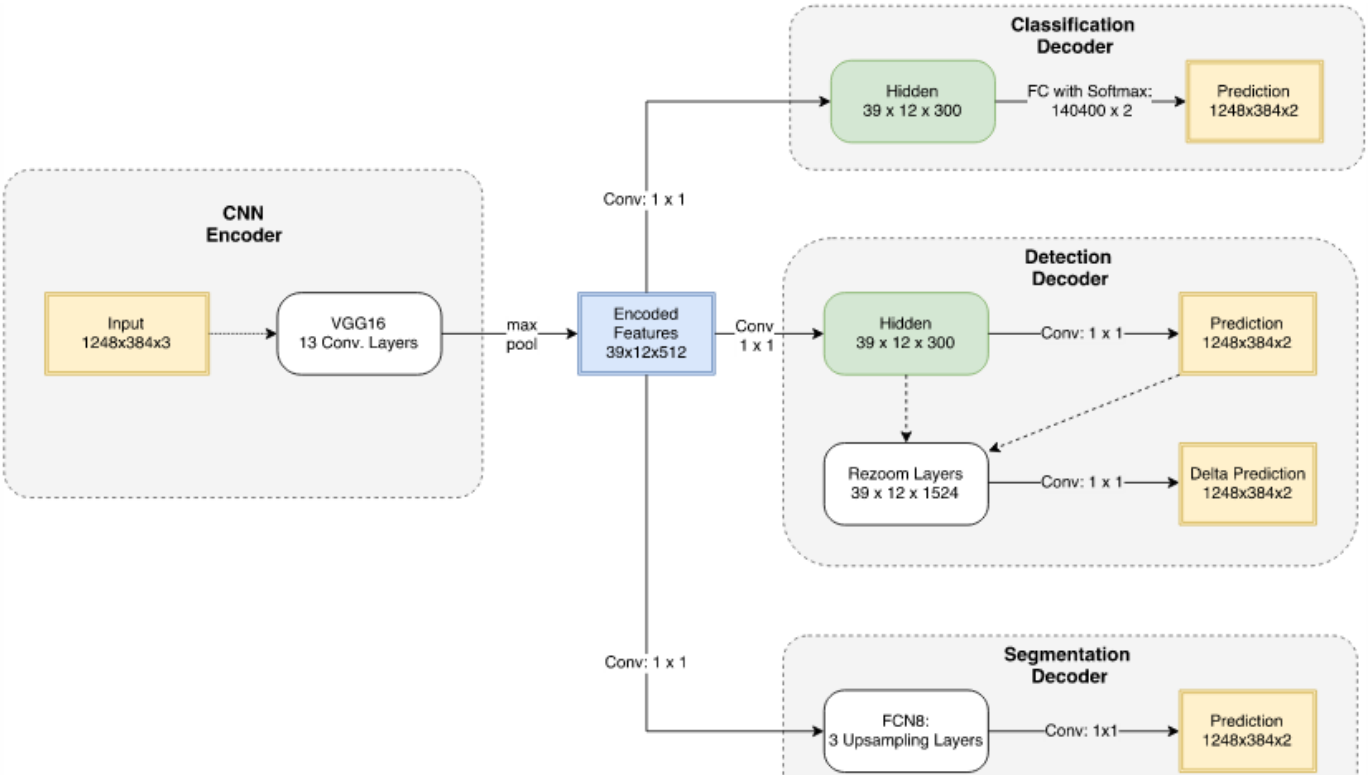
C. Multipurpose Deep Decoder Deconvolution Network [MultiNet]

Embora a maioria das abordagens de segmentação semântica tenha se concentrado em melhorar o desempenho, neste artigo os autores argumentam que os tempos computacionais são muito importantes para permitir aplicações em tempo real, como a condução autônoma. Com este objetivo,

foi apresentado uma abordagem para classificação, detecção e segmentação semântica de forma conjunta através de uma arquitetura unificada onde o codificador é compartilhado entre as três tarefas. Nesta abordagem, o sistema pode ser treinado de treinado ponta a ponta e funciona bem no conjunto de dados KITTI. Segundo os autores esta abordagem é extremamente eficiente, levando menos de 100 ms para executar todas as tarefas. [1]

III. METODOLOGIA

A rede utilizada foi a MultiNet [1]. Conforme podemos ver na Figura 1, na arquitetura proposta há um codificador comum para as três tarefas que podem ser executadas pela rede e cada tarefa possui seu próprio decodificador. A tarefa do codificador é processar uma imagem e extrair *features* que contenham todas as informações necessárias para realização da segmentação, detecção e classificação da imagem. O codificador da MultiNet consiste nas primeiras 13 camadas da rede VGG16 [7], que são aplicadas de forma totalmente convolucional à imagem produzindo um tensor de tamanho 39x12x512. Esta é a saída da 5ª camada de *pooling*, denominada *pool5* na implementação do VGG.



O **decodificador de classificação** é projetado para aproveitar o codificador. Para esse objetivo, é aplicada uma convolução 1x1, seguida de uma camada totalmente conectada e uma camada softmax para produzir as probabilidades de classe final.

O **decodificador de detecção**, é projetado para ser um sistema de detecção baseado em regressão. A abordagem é inspirada por ReInspect [8], Yolo [9] e Overfeat [10]. Além do pipeline de regressão padrão, foi incluída uma abordagem de ROI *pooling*, que permite que a rede utilize recursos em uma resolução maior. O primeiro passo é produzir uma estimativa aproximada dos *bounding boxes*. Para este objetivo, as *features* passam por uma camada convolutiva 1x1 com 500 filtros, produzindo um tensor de tamanho 39x12x500. Este tensor é processado com outra camada convolucional 1x1 que produz 6 canais na resolução 39x12. Os dois primeiros canais desse tensor formam uma segmentação grosseira da imagem. Seus valores representam a confiança de que um objeto de interesse está presente naquele local específico na grade 39x12. Os últimos quatro canais representam as coordenadas de uma caixa delimitadora na área em torno dessa célula.

O **decodificador de segmentação** segue a arquitetura FCN (*Fully Convolutional Networks*). Dado o codificador, transformamos as camadas completamente conectadas (FC) restantes da arquitetura VGG em camadas convolutivas 1x1 para produzir uma segmentação de baixa resolução de tamanho 39x12. Seguem-se três camadas de convolução transpostas para realizar a amostragem. As camadas s*kip* são utilizadas para extrair *features* de alta resolução das camadas inferiores. Essas *features* são processadas por uma camada de convolução 1x1 e, em seguida, adicionadas aos resultados parcialmente amostrados.

IV. METODOLOGIA EXPERIMENTAL

Para avaliar a rede MultiNet na tarefa de segmentação de pista nos dados da IARA, apenas o **decodificador de segmentação** foi utilizado. No código fonte disponibilizado pelos autores em https://github.com/MarvinTeichmann/MultiNet essa parte está separado no módulo KittiSeg https://github.com/MarvinTeichmann/KittiSeg.

Experimentos foram realizados utilizando imagens da capturadas pelas câmeras da IARA: *Bumblebee* e *ZED*, e compararemos o desempenho do módulo KittiSeg da MultiNet em ambas. Os detalhes dos datasets, descrição dos experimentos e configuração utilizada são detalhados a seguir.

A. Datasets

Para avaliar a qualidade da KittiSeg nos dados da IARA, dois datasets foram criados: *Bumbleee e ZED*. Os datasets foram separados de forma a contemplar os desafios encontrados pela IARA nas regiões onde trafega, usando as duas câmeras disponíveis (Point Grey Bumblebee XB3 stereo e ZED Stereo Camera). As imagens são gravadas de uma volta onde um motorista conduz o carro por um percurso definido. As regiões escolhidas foram uma Avenida dentro de um perímetro urbano e o anel viário da Universidade Federal do Espiríto Santo – UFES. Na Avenida Fernando Ferrari as pistas são bem delimitadas e com boas marcações, além de múltiplas lanes. Já no anel viário da UFES, possui marcações precárias, e com diferentes tipos de pavimentação e apenas duas faixas na maior parte do percurso.

O dataset da Bumblebee foi gerado extraindo imagens gravadas da câmera Point Grey Bumblebee XB3 stereo instalada no teto da IARA voltada para pista. Neste caso os dados foram da câmera direita, por ser a mais próxima do centro do veículo. O dataset contém imagens das duas regiões: Avenida Fernando Ferrari (Figura 1) e o anel viário da UFES (Figura 2). Esse dataset contém 372 imagens da primeira região e 1383 da segunda, com resolução de 1280x387.

O dataset da ZED foi gerado extraindo imagens gravadas da câmera ZED Stereo instalada no para-brisas da IARA voltada para pista. Apenas a câmera esquerda foi utilizada, por ser a câmera central. As imagens são apenas da região da Avenida Fernando Ferrari (Figura 3). Este dataset é composto de 181 imagens com resolução 1920x580.

Pela posição, ambas as câmeras apresentam na imagem, parte do capô do carro. Para melhor comparação, as imagens foram cortadas para remover o capô e para manter o ratio da base KITTY sem perder o trecho da pista.

Além da posição onde estão instaladas na IARA, as câmeras utilizadas apresentam diferenças em brilho, contraste, campo de visão, entre outros. Os dois datasets foram criados buscando encontrar a melhor configuração para aplicação da rede neural.

Figura 1:Dataset Bumblebee Aveninda Fernando Ferrari

Figura 2: Dataset Bumblebee Anel viário da UFES

Figura 3: Dataset ZED Avenida Fernando Ferrari

B. Experimentos

Para verificar a qualidade da KittiSeg, nos diferentes datasets, foram variados os valores do *thresholding* utilizado

pela rede para determinar o nível de confiança mínimo para considerar um pixel como pista ou não. Inicialmente uma pequena parte dos dados foram separados e os valores de *thresholding* foram variados até o valor mínimo de probabilidade inferido pela rede, neste caso, 0.0001. Encontrado o limite mínimo, os valores foram variados de 0.5 até 0.0001 de em um fator de 5 e depois de 2, (0.5, 0.1, 0.05, 0.01...), para identificar os pontos de maior variação dos da qualidade dos resultados. Com essa informação foram realizados 4 experimentos utilizando todo o conjunto de imagem dos datasets variando os *thresholding*. Os valores escolhidos foram:

Experimento	Tresholding
Experimento 1	0.5
Experimento 2	0.1
Experimento 3	0.01
Experimento 4	0.001

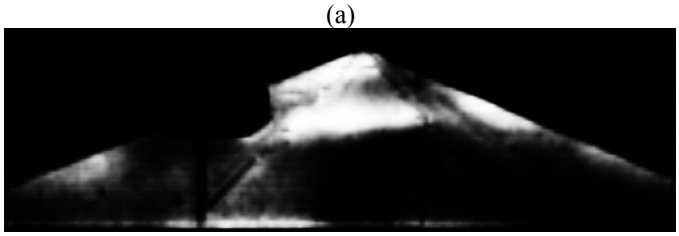
C. *Configuração utilizada*

Para a realização dos experimentos utilizamos um computador Dell Precision R5500 com sistema operacional Ubuntu 14.04.4 LTS, TensorFlow 1.0, CUDA 8.0, cudnn 6, 12GB de memória RAM, processador Intel® Xeon(R) CPU E5606 @ 2.13GHz x 8 e placa de vídeo Titan X (Pascal) 12GB.

V. Resultados

No dataset da ZED e Bumblebee na Fernando Ferrari, no Experimento 1 verificou-se que a rede tinha um desempenho abaixo do apresentado nos dados da KITTI. Porém, analisando o mapa de confiança fornecido pela rede (Figura 4(b)), foi possível observar, que uma grande parte da pista estava sendo inferida, mesmo que com baixa confiança, a Figura 4(a) é possível verificar a sobreposição em verde onde a confiança da rede está acima do *thresholding* definido (0.5). Então para melhor analisar o desempenho da rede, os outros experimentos foram necessários.

(a)



(b)

Figura 4: Resultado da inferência com threshold a 0.5. (a) Região sobreposta em verde simboliza a saída da rede. (b) Mapa de confiança da rede, onde quanto mais claro, maior a probabilidade do pixel ser da classe pista.

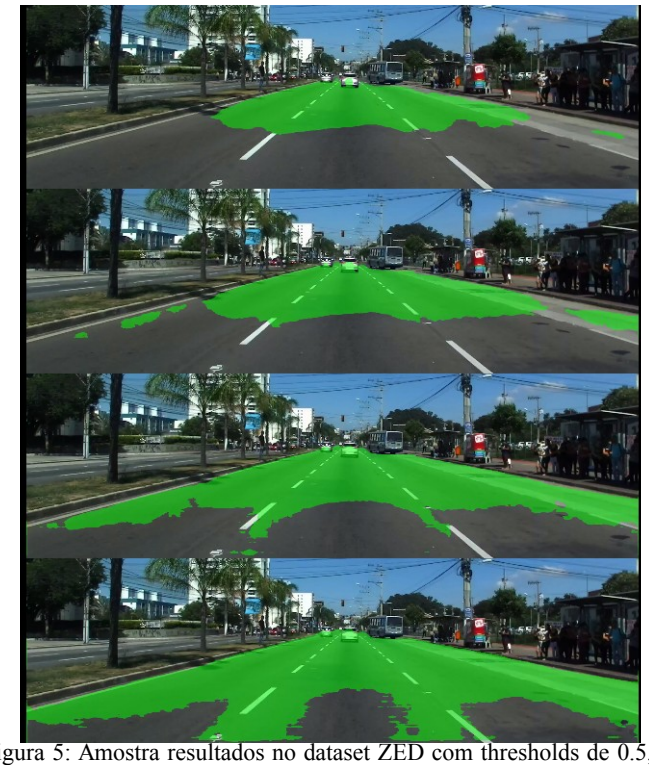


Figura 5: Amostra resultados no dataset ZED com thresholds de 0.5, 0.1, 0.01 e 0.001 respectivamente.

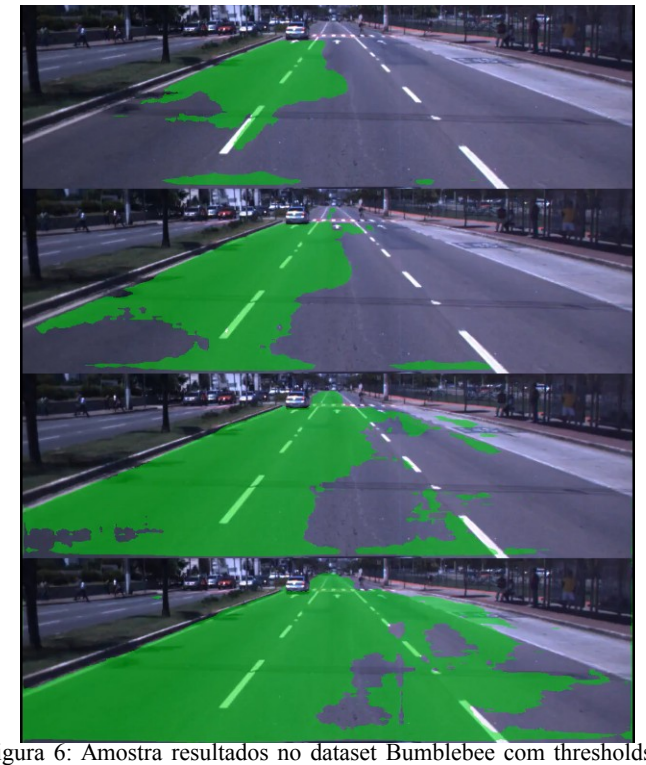


Figura 6: Amostra resultados no dataset Bumblebee com thresholds de 0.5, 0.1, 0.01 e 0.001 respectivamente.

A Figura 5 apresenta alguns dos resultados obtidos nos experimento 1 a 4 com o dataset ZED e a Figura 6 resultados no dataset Bumblebee em regiões semelhantes na Fernando Ferrari. No experimento foi possível verificar que as delimitações da pista foram bem inferidas pela rede, e as delimitações da área ocupada pelos carros, quando detectados.

Porém, próximo a IARA a pista não é detectada corretamente e a maior parte da pista está abaixo do thresholding definido do experimento 1. Nas figuras Figura 5 é possível verificar que a rede detecta o recuo de parada de ônibus como parte da pista. Na Figura 5 é possível verificar que um veículo na cor prata, é confundido com parte da pista.

Consideramos que o melhor desempenho da rede fica entre 0.01 e 0.001 nos datasets da ZED e Bumblebee na Fernando Ferrari. Os resultados em todo o dataset pode ser visto nos vídeos: https://youtu.be/_xiEXOvRT4Q e https://youtu.be/cOlB6dKSw-g respectivamente.

No dataset Bumblebee no anel viário da UFES, a rede demonstrou capacidade de separar faixas (Figura 7 e Figura 8). Onde a faixa era continua (mão dupla) a rede detectava apenas a faixa onde a IARA trafegava, como é possível observar na Figura 7. Também foi observado que apenas nos thresholding abaixo de 0.01 que a maior parte da pista foi identificada. Abaixo de 0.01 é possível observar o aumento de falsos positivos. Um vídeo da inferência no dataset completo está disponível em: https://youtu.be/nlM3osCNvXc.

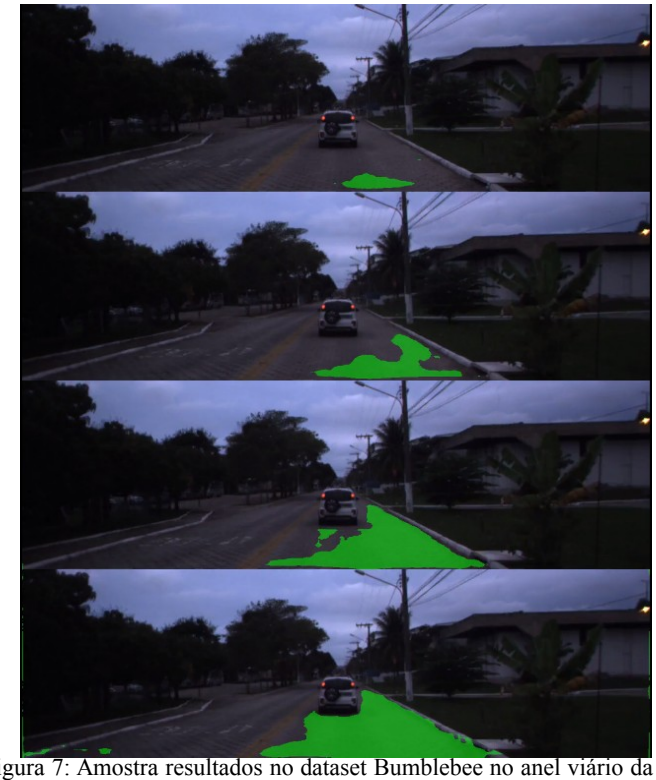


Figura 7: Amostra resultados no dataset Bumblebee no anel viário da UFES com thresholds de 0.5, 0.1, 0.01 e 0.001 respectivamente. Região com pista de paralelepípedo e mão dupla.

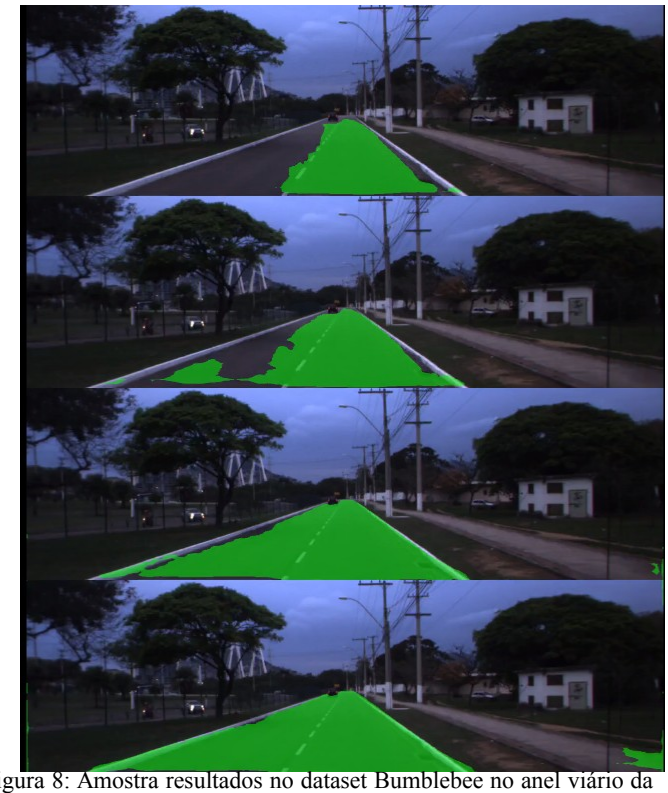


Figura 8: Amostra resultados no dataset Bumblebee no anel viário da UFES com thresholds de 0.5, 0.1, 0.01 e 0.001 respectivamente. Via de mão única com duas pistas.

Analisando todo o percurso, e experimentos, de forma geral, no dataset da Bumblebee, a MultiNet teve um desempenho um pouco melhor. Porém todos os melhores resultados ficaram entre os tresholding de 0.01 e 0.001. Uma comparação entre o melhor resultado da ZED e Bumblebee na Fernando Ferrari pode ser visto no https://youtu.be/zjF5GQnOzig.

A KittiSeg ocupa 11.74GB da GPU, porém a inferência se mostrou suficientemente rápida, em nossos experimentos, em uma média de 140ms.

VI. Conclusão

Neste trabalho a rede Multinet foi avaliada na tarefa de segmentação de pista nos dados da IARA. Para os experimentos foram separados datasets em duas regiões distintas e usando dados de duas câmeras diferentes instaladas na IARA. A rede mostrou um desempenho muito abaixo do resultado apresentado no dataset KITTI. Entretanto, nos experimentos é possível verificar o potencial da abordagem. Para trabalhos futuros fazer um fine-tuning pode gerar melhores resultados.

References

[1] Teichmann, M., Weber, M., Zoellner, M., Cipolla, R., & Urtasun, R. (2016). MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving. arXiv preprint arXiv:1612.07695.

[2] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

[3] Geiger, A., Lenz, P., & Urtasun, R. (2012, June). Are we ready for autonomous driving? the kitti vision benchmark suite. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on (pp. 3354-3361). IEEE.

[4] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[5] Oliveira, G. L., Burgard, W., & Brox, T. (2016, October). Efficient deep methods for monocular road segmentation. In *IEEE/RSJ international conference on intelligent robots and systems (IROS 2016)*.

[6] Munoz-Bulnes, J., Fernandez, C., Parra, I., Fernández-Llorca, D., & Sotelo, M. A. Deep Fully Convolutional Networks with Random Data Augmentation for Enhanced Generalization in Road Detection.

[7] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[8] Ren, M., & Zemel, R. S. End-to-End Instance Segmentation and Counting with Recurrent Attention. arXiv 2016. arXiv preprint arXiv:1605.09410.

[9] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[10] Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229.