**UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS**

Raul de Queiroz Mendes

**Aprendizagem Profunda Aplicada a Estimação Monocular
de Profundidade: Uma Abordagem Baseada em Atenção
e Complementação**

São Carlos

2021

Raul de Queiroz Mendes

**Aprendizagem Profunda Aplicada a Estimação Monocular
de Profundidade: Uma Abordagem Baseada em Atenção
e Complementação**

Dissertação de Mestrado apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo, para obtenção do título de Mestre em Ciências - Programa de Pós-Graduação em Engenharia Elétrica.

Área de concentração: Sistemas Dinâmicos

Orientador: Prof. Dr. Valdir Grassi Junior

São Carlos

2021

Trata-se da versão corrigida da dissertação. A versão original se encontra disponível na EESC/USP que aloja o programa de Pós-Graduação de Engenharia Elétrica.

AUTORIZO A REPRODUÇÃO TOTAL OU PARCIAL DESTE TRABALHO,
POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS
DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da
EESC/USP com os dados inseridos pelo(a) autor(a).

D538a de Queiroz Mendes, Raul
Aprendizagem profunda aplicada a estimação
monocular de profundidade: uma abordagem baseada em
atenção e complementação / Raul de Queiroz Mendes;
orientador Valdir Grassi Junior . São Carlos, 2021.

Dissertação (Mestrado) - Programa de
Pós-Graduação em Engenharia Elétrica e Área de
Concentração em Sistemas Dinâmicos -- Escola de
Engenharia de São Carlos da Universidade de São Paulo,
2021.

1. SIDE. 2. Aprendizagem Profunda. 3. Visão
Computacional. I. Título.

FOLHA DE JULGAMENTO

Candidato: Bacharel **RAUL DE QUEIROZ MENDES**.

Título da dissertação: "Aprendizagem Profunda Aplicada a Estimção Monocular de Profundidade: Uma Abordagem Baseada em Atenção e Complementação".

Data da defesa: 30/07/2021.

Comissão Julgadora

Resultado

Prof. Dr. **Valdir Grassi Junior**
(Orientador)

(Escola de Engenharia de São Carlos – EESC/USP)

Aprovado

Prof. Dr. **Fernando Santos Osório**

(Instituto de Ciências Matemáticas e de Computação/ICMC-USP)

Aprovado

Profa. Dra. **Esther Luna Colombini**

(Universidade de Campinas/UNICAMP)

Aprovado

Coordenador do Programa de Pós-Graduação em Engenharia Elétrica:

Prof. Dr. **João Bosco Augusto London Junior**

Presidente da Comissão de Pós-Graduação:

Prof. Titular **Murilo Araujo Romero**

À minha família que amo ad aeternum. Aos meus parentes falecidos (in memoriam).

AGRADECIMENTOS

À Universidade de São Paulo (USP) pelos cursos, professores e funcionários de excelência, pela infraestrutura de ponta e pelos programas de pós-graduação que estão sempre na vanguarda do fomento científico brasileiro. À Escola de Engenharia de São Carlos (EESC) que disponibiliza espaços como salas de aula, bibliotecas e restaurantes universitários para seus alunos. Em especial, ao Departamento de Engenharia Elétrica e de Computação (SEL) que dispõe de laboratórios e equipamentos avançados, os quais servem aos pesquisadores do país.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq - concessão 465755/2014-3), à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Capes - código financeiro 001 e concessão 88887.339070/2019-00) e à Fundação de Amparo a Pesquisa do Estado de São Paulo (FAPESP - concessão 2014/50851-0) pelo financiamento deste trabalho e pelo constante apoio a pesquisa brasileira em todas as esferas do conhecimento.

Ao meu orientador Prof. Dr. Valdir Grassi Junior por incentivar e auxiliar meu desenvolvimento na carreira científica e por me guiar desde o começo do mestrado até a finalização deste projeto. Aos pesquisadores Me. Eduardo Godinho Ribero e Me. Nicolas dos Santos Rosa pelas mentorias que possibilitaram a minha evolução como profissional e como pessoa.

Ao Laboratório de Sistemas Inteligentes (LASI), ao Centro de Robótica da USP (CRob) e ao Instituto Nacional de Ciência e Tecnologia para Sistemas Autônomos Cooperativos (InSAC), sob o comando do Prof. Dr. Marco Henrique Terra, pelo apoio técnico especializado e pelo suporte de laboratórios e ferramentas de trabalho.

Aos meus pais, Maria Fernanda e Elvis, e à minha irmã, Raquel, pelos esforços, pela resiliência, por sempre acreditarem na minha capacidade, apesar de todos os empecilhos no caminho, e por nunca deixarem de dar respaldo a minha dedicação aos estudos e a busca pelos meus objetivos. À cada pessoa da minha família pelo amor, carinho, preocupação e altruísmo, e à minha companheira Paula Alejandra pela empatia e cumplicidade ao longo desses anos de mestrado.

Aos meus amigos e amigas de laboratório por estarem sempre dispostos a colaborar com experiências e conhecimento, e a fornecer amparo técnico e emocional. Aos meus amigos e amigas pessoais e a todas as pessoas que passaram pela minha vida e transformaram minha maneira de pensar, de forma a ampliar meus horizontes.

“May not machines carry out something which ought to be described as thinking but which is very different from what a man does? This objection is a very strong one, but at least we can say that if, nevertheless, a machine can be constructed to play the imitation game satisfactorily, we need not be troubled by this objection... I believe that at the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted... We may hope that machines will eventually compete with men in all purely intellectual fields. But which are the best ones to start with? Even this is a difficult decision. Many people think that a very abstract activity, like the playing of chess, would be best. It can also be maintained that it is best to provide the machine with the best sense organs that money can buy, and then teach it to understand and speak English. This process could follow the normal teaching of a child. Things would be pointed out and named, etc. Again I do not know what the right answer is, but I think both approaches should be tried. We can only see a short distance ahead, but we can see plenty there that needs to be done.”

(TURING, 1950)

RESUMO

Mendes, R. Q. **Aprendizagem Profunda Aplicada a Estimção Monocular de Profundidade: Uma Abordagem Baseada em Atenção e Complementação.** 2021. 170p. Dissertação (Mestrado) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2021.

Medir a profundidade de imagens é um problema inverso fundamental dentro do campo da Visão Computacional, uma vez que as informações de profundidade são obtidas por meio de imagens 2D, as quais podem ser geradas a partir de infinitas possibilidades de cenas reais observadas. Ademais, tal problema é não bem-posto, pois os valores estimados de profundidade são fortemente dependentes das imagens de espaços do mundo real adquiridas. Outras importantes tarefas da área de Robótica recorrem a medidas de profundidade, como a Localização e Mapeamento Simultâneos (SLAM) e *Structure from Motion* (SfM). Beneficiando-se do progresso de Redes Neurais Convolucionais Profundas (DCNNs) para explorar características estruturais e informações espaciais de imagens, a Estimção de Profundidade a partir de Uma Única Imagem (SIDE) é frequentemente destacada em meios de inovação científica e tecnológica, já que este conceito proporciona vantagens relacionadas ao seu baixo custo de implementação, menores restrições de uso e robustez a condições ambientais. No contexto de veículos autônomos, as DCNNs otimizam a tarefa de SIDE através da predição de mapas com dados precisos de profundidade, os quais são indispensáveis durante o processo de navegação autônoma em locais distintos. No entanto, essas redes geralmente são treinadas em mapas de profundidade esparsos e ruidosos, gerados por varreduras de *Light Detection and Ranging laser* (LiDAR) ou *structured-light* e *time-of-flight devices* (Kinect), e são executadas com alto custo computacional, exigindo Unidades de Processamento Gráfico (GPUs) de alto desempenho. Sendo assim, este trabalho propõe uma nova arquitetura de DCNN supervisionada e funções de custo baseadas em atenção para solucionar problemas de SIDE. Inova-se também ao incorporar múltiplas técnicas de Visão Computacional, como a utilização de algoritmos de densificação e informações adicionais de semântica, de profundidade e de normais de superfície ao treinamento de tal DCNN. O método introduzido neste trabalho tem foco em aplicações de veículos autônomos em ambientes internos e externos.

Palavras-chave: SIDE, Aprendizagem Profunda, Visão Computacional.

ABSTRACT

Mendes, R. Q. **Deep Learning Applied to Monocular Depth Estimation: An Approach Based on Attention and Complementation.** 2021. 170p. Dissertação (Mestrado) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2021.

Measuring the depth of images is a fundamental inverse problem within the field of Computer Vision since the depth information is obtained through 2D images, which can be generated from infinite possibilities of observed real scenes. Furthermore, this problem is ill-posed, since the estimated depth values are strongly dependent on the acquired images of real-world spaces. Other important tasks in the area of Robotics use depth measurements, such as Simultaneous Localization and Mapping (SLAM) and Structure from Motion (SfM). Benefiting from the progress of Deep Convolutional Neural Networks (DCNNs) to explore structural features and spatial image information, Single Image Depth Estimation (SIDE) is often highlighted in means of scientific and technological innovation, as such a concept provides advantages related to its low-cost implementation, fewer restrictions on use and robustness to environmental conditions. In the context of autonomous vehicles, DCNNs optimize the task of SIDE by predicting maps with accurate depth data, which are indispensable during the autonomous navigation process in different locations. However, these networks are usually trained on sparse and noisy depth maps, generated by scans of Light Detection and Ranging laser (LiDAR) or structured-light and time-of-flight devices (Kinect), and are executed with high computational cost, requiring powerful Graphic Processing Units (GPUs). Therefore, this work proposes a new supervised DCNN architecture and attention-based loss functions to solve SIDE problems. It is also innovative by incorporating multiple Computer Vision techniques, such as the use of densification algorithms and additional information on semantics, depth and surface normals to the training of such DCNN. The method introduced in this work focuses on autonomous vehicles applications in indoor and outdoor environments.

Keywords: SIDE, Computer Vision, Deep Learning.

LISTA DE FIGURAS

Figura 1 – Representação da rede Perceptron	40
Figura 2 – Principais funções de ativação utilizadas em ANNs	41
Figura 3 – Ilustração da rede Perceptron multicamadas	42
Figura 4 – Subdivisões do campo da Inteligência Artificial	44
Figura 5 – Operação de convolução discreta	46
Figura 6 – Exemplo de topologia FCN (SegNet)	47
Figura 7 – Exemplo de topologia CNN-FC (AlexNet)	48
Figura 8 – Comportamento da função ReLU	48
Figura 9 – Funções de ativação para CNNs	49
Figura 10 – Funcionamento da Subamostragem	50
Figura 11 – Aplicação da operação de convolução em uma imagem sem preenchimento	50
Figura 12 – Aplicação da operação de convolução em uma imagem com preenchi- mento semelhante	51
Figura 13 – Representação do funcionamento da técnica de desligamento durante as fases de treinamento e de teste	52
Figura 14 – Resultado da implementação da técnica de desligamento	52
Figura 15 – Métodos de normalização de mapas de características	53
Figura 16 – Geometria Epipolar	60
Figura 17 – Retificação de um par estéreo	61
Figura 18 – Modelo 3D gerado a partir dos rótulos geométricos da imagem	63
Figura 19 – <i>Layout</i> da superfície da imagem desenvolvido por meio de classes geo- métricas	63
Figura 20 – Vetores de características de <i>patches</i> na imagem	64
Figura 21 – Classificador de profundidade e semântica	65
Figura 22 – Técnica de estimação de profundidade que utiliza o algoritmo SIFT (LOWE, 2004) e informações temporais extraídas de vídeos	65
Figura 23 – Estrutura fundamental da ResNet	67
Figura 24 – Estrutura fundamental da DenseNet	68
Figura 25 – Arquitetura constituída de uma CNN de escala grossa global e uma CNN de escala fina local	70
Figura 26 – Blocos de <i>up-projection</i>	70
Figura 27 – Modelo de SIDE baseado em classificação que utiliza CRF	71
Figura 28 – Método de regressão ordinal para estimar profundidade de maneira monocular	72
Figura 29 – Arquitetura de rede FCN com um módulo de fusão multi-escala	73
Figura 30 – CNN composta por camadas de orientação planar local	73

Figura 31 – Arquitetura de rede Transformer com detalhamento da atenção ao produto ponto em escala	75
Figura 32 – <i>Baseline</i> da CNN proposta	92
Figura 33 – Estrutura de CNN introduzida para a tarefa de preensão robótica . . .	93
Figura 34 – Modelo fundamental de DenseNet	94
Figura 35 – <i>Framework</i> para análise de informações adicionais de profundidade . .	98
Figura 36 – <i>Framework</i> para análise de informações adicionais de semântica	99
Figura 37 – <i>Framework</i> para análise de informações adicionais de normais de superfície	100
Figura 38 – Etapas de treinamento do modelo proposto	106
Figura 39 – Exemplos de par imagem RGB e mapa de profundidade do <i>dataset</i> KITTI Depth	107
Figura 40 – Exemplos de imagens RGB, mapas de profundidade, mapas de semântica e de normais de superfície encontrados no <i>dataset</i> NYU Depth V2 . . .	108
Figura 41 – Detalhes da <i>Baseline</i> proposta	112
Figura 42 – Comparação qualitativa entre as predições da melhor abordagem desenvolvida (DSN †) e de métodos auto-supervisionados recentes que abordam a tarefa de SIDE. As imagens de entrada e o <i>ground truth</i> pertencem ao conjunto de dados KITTI Depth (UHRIG et al., 2017). Os mapas de profundidade de referência são interpolados para fins de visualização. Devido a discrepância das paletas de cores utilizadas, a comparação entre as imagens deve ser realizada considerando-se a geometria e o <i>layout</i> das cenas.	121
Figura 43 – Diferenças entre os mapas de profundidade retornados pela melhor configuração da DSN (DSN †) e por abordagens de SIDE no estado-da-arte com <i>pipelines</i> de treinamento supervisionados. As imagens RGB de entrada compõem o <i>dataset</i> KITTI Depth (UHRIG et al., 2017) . .	122
Figura 44 – Análise qualitativa das nuvens de pontos reconstruídas a partir das predições da DSN. DSN † indica a configuração da DSN que produz às métricas gerais mais precisas. A entrada RGB e as nuvens de pontos de referência com dados de profundidade são retiradas do conjunto de dados KITTI Depth (UHRIG et al., 2017). No entanto, as informações das normais da superfície de referência são geradas com o auxílio do algoritmo introduzido por Fouhey, Gupta e Hebert (2013)	122
Figura 45 – Evolução dos erros e da precisão das predições da DSN de acordo com o número de amostras esparsas de entrada. Utilizou-se o <i>ground truth</i> do <i>dataset</i> KITTI Depth (UHRIG et al., 2017) para apresentar os resultados	124

Figura 46 – Avaliação qualitativa do <i>framework</i> de densificação de profundidade com base no <i>ground truth</i> ruidoso e esparsos do <i>dataset</i> KITTI Depth (UHRIG et al., 2017). Legenda: Entrada - imagem RGB de entrada; GT - <i>ground truth</i> ; d - dados de entrada esparsos	125
Figura 47 – Comparação entre os erros e a precisão obtidos pelo <i>framework</i> de densificação de profundidade (sem o módulo de normais de superfície), quando ele é apresentado a dados esparsos do <i>ground truth</i> e aos mapas de profundidade produzidos pela Monodepth2 (GODARD et al., 2019)	125
Figura 48 – Resultados qualitativos das estimações de trajetória da câmera obtidas nas sequências 00, 02, 03, 05, 07, 08 e 10 do KITTI Odometry (GEIGER; LENZ; URTASUN, 2012) pela CNN-SVO + BA em conjunto com a DSN e o módulo de normais de superfície. As trajetórias pretas pontilhadas e azuis sólidas referem-se ao <i>ground truth</i> e as predições respectivamente	129
Figura 49 – Perfil de <i>scale drift</i> dos resultados de <i>tracking</i> gerados pela CNN-SVO + BA + DSN (com o módulo de normais de superfície). As sequências são retiradas do conjunto de dados KITTI Odometry (GEIGER; LENZ; URTASUN, 2012)	131
Figura 50 – Comparação entre as predições de profundidade de cenas <i>indoor</i> do <i>pipeline</i> de SIDE proposto e de outros métodos supervisionados no estado-da-arte. A abordagem DSN † está relacionada à configuração da DSN que produz os resultados gerais mais precisos. As imagens de entrada e o <i>ground truth</i> pertencem ao <i>dataset</i> NYU Depth V2 (SILBERMAN et al., 2012). O <i>ground truth</i> é interpolado para fins de visualização	135
Figura 51 – Avaliação detalhada das predições do método introduzido e daquelas recuperadas por abordagens supervisionadas recentes. DSN † refere-se à melhor configuração da DSN e as entradas RGB compõem o conjunto de dados NYU Depth V2 (SILBERMAN et al., 2012)	137
Figura 52 – Comparação qualitativa dos mapas de normais de superfície retornados pelo <i>framework</i> desenvolvido e por técnicas no estado-da-arte. DSN † corresponde à melhor configuração da DSN. As entradas RGB fazem parte do <i>dataset</i> NYU Depth V2 (SILBERMAN et al., 2012). Determinou-se as imagens do <i>ground truth</i> através do método de Fouhey, Gupta e Hebert (2013)	138

Figura 53 – Avaliação qualitativa das nuvens de pontos reconstruídas a partir das predições mais precisas da DSN. As imagens de entrada e o <i>ground truth</i> de profundidade compõem o <i>dataset</i> NYU Depth V2 (SILBERMAN et al., 2012). No entanto, gerou-se os dados de normais de superfície de referência com o auxílio do algoritmo desenvolvido por Fouhey, Gupta e Hebert (2013)	139
Figura 54 – Evolução das métricas de densificação de profundidade obtidas pela DSN. Empregou-se os mapas de referência de profundidade do conjunto de dados NYU Depth V2 (SILBERMAN et al., 2012) para traçar os gráficos	140
Figura 55 – Comparação entre os mapas de profundidade densos retornados pela melhor configuração da CNN de densificação de profundidade (DSN †) e por outros métodos na literatura. Têm-se também os mapas de normais de superfície estimados pela DSN †. O NYU Depth V2 (SILBERMAN et al., 2012) é empregado por todas as abordagens. Legenda: RGB - imagem de entrada; 500d - dados de entrada esparsos; GT - <i>ground truth</i>	141

LISTA DE TABELAS

Tabela 1	– Especificações de trabalhos no estado-da-arte da tarefa de SIDE. O símbolo $\dagger$ indica o número total de parâmetros treináveis do <i>encoder</i> , enquanto $\ddagger$ refere-se ao tamanho total da estrutura de rede	89
Tabela 2	– Composição dos modelos propostos para serem aplicados como <i>backbone</i> na etapa de extração de características da <i>baseline</i> . Para todos os modelos, k indica a taxa de crescimento dos filtros, θ é a compressão dos filtros e t é o número total de parâmetros	95
Tabela 3	– Dimensões de entrada e saída das camadas da CNN proposta para imagens dos <i>datasets outdoor</i> e <i>indoor</i>	97
Tabela 4	– Comparação de métodos de densificação de profundidade no estado-da-arte de acordo com as métricas do KITTI Benchmark (UHRIG et al., 2017) ($iMAE = \frac{1}{MAE}$ e $iRMSE = \frac{1}{RMSE}$)	113
Tabela 5	– Análise dos graus de densificação dos <i>datasets</i> utilizados de acordo com o número médio e a porcentagem média de pixels válidos. A resolução 375×1242 é típica para o KITTI Depth (UHRIG et al., 2017) e 352×704 é o tamanho da imagem que é alimentada no <i>framework</i>	114
Tabela 6	– Avaliação do método proposto considerando variações na função de custo e no <i>dataset</i> utilizado durante o treinamento. A DenseNet-121 é fixada como a rede de extração de características no <i>encoder</i> da DSN .	115
Tabela 7	– Comparação dos resultados obtidos pelo <i>framework</i> proposto com diferentes redes de extração de características no <i>dataset</i> KITTI Depth (UHRIG et al., 2017). O Modelo 1, o Modelo 2 e o Modelo 3 referem-se a DenseSIDENet com os <i>encoders</i> da Tabela 2. A MobileNetV2-50 e a MobileNetV2-140 são versões da MobileNetV2 com multiplicadores de profundidade 1, 4 e 0,5 respectivamente. A função de custo BerHu é empregada em todos os experimentos e a coluna Mapa indica o número total de mapas de características de entrada para o estágio de decodificação da DSN	116
Tabela 8	– Comportamento da DSN quando sujeita a modificações em sua estrutura e processo de treinamento. A DSN 3 é a CNN <i>baseline</i> com o Modelo 3 como codificador. O Pyramid ¹ e o Pyramid ² são módulos em forma de pirâmide implementados no decodificador. CS e Sem referem-se ao pré-treinamento com as disparidades do conjunto de dados do Cityscapes (CORDTS et al., 2016) e com os mapas semânticos do KITTI (ALHAIJA et al., 2018) e do Cityscapes (CORDTS et al., 2016) respectivamente. A expressão $\mathcal{L}_{BerHu}^+$ é igual à $\mathcal{L}_{BerHu}$, porém com $\kappa = 4$	118

Tabela 9 – Avaliação da DSN com o módulo de normais de superfície no <i>dataset</i> KITTI Depth (UHRIG et al., 2017). A configuração DSN 3 + Pyramid ¹ + CS + SN, bem como o detector de bordas Sobel e a função de custo $\mathcal{L}_{BerHu}^+$, é padrão para todos os experimentos. SN significa que o treinamento é realizado com o <i>ground truth</i> adicional de normais de superfície. A coluna Blur indica a presença ou ausência de uma convolução com kernel gaussiano. As constantes ψ_1 e ψ_2 formam a função de custo $2,5D$ e são empregadas nas etapas de pré-treinamento e treinamento respectivamente	119
Tabela 10 – Desempenho da FCN proposta e de outros trabalhos na literatura de SIDE de acordo com as métricas do KITTI Depth (UHRIG et al., 2017). Legenda: Cal. Cam. - câmera calibrada; CM - movimento da câmera; DM - mapa de disparidade; DSN † - configuração da DSN que produz os resultados gerais mais precisos; DSO - odometria esparsa direta; GAN - Rede Generativa Adversarial; GT - <i>ground truth</i> ; MV - vídeo monocular; SGM - correspondência semi-global; SI - imagem única; SL - rótulos semânticos; SN - normais de superfície; SP - par estéreo; Synt. - sintético	120
Tabela 11 – Avaliação do <i>pipeline</i> de densificação de profundidade de acordo com Eigen Split aplicado ao conjunto de dados do KITTI Depth (UHRIG et al., 2017). A configuração DSN 3 + Pyramid ¹ + CS, além da função $\mathcal{L}_{BerHu}^+$, é considerada como padrão para os primeiros experimentos. A configuração DSN 3 + Pyramid ¹ + CS + SN, juntamente com o detector de borda, é analisada posteriormente	123
Tabela 12 – Comparação quantitativa dos resultados do <i>framework</i> proposto e dos resultados de outras técnicas de densificação de profundidade. As métricas são calculadas considerando o Eigen Split e o conjunto de dados KITTI Depth (UHRIG et al., 2017). Aplicou-se o limite de 80 <i>m</i> em todas as análises	124
Tabela 13 – Avaliação das predições da DSN, aplicadas no <i>framework</i> da CNN-SVO, de acordo com a métrica RMSE. Foram utilizadas as primeiras 11 sequências do <i>dataset</i> KITTI Odometry (GEIGER; LENZ; URTASUN, 2012) para gerar os resultados. Legenda: BA - Bundle Adjustment; DSN † - configuração DSN, sem o módulo de normais de superfície, que produz os melhores resultados; DSN ‡ - configuração de DSN, com o módulo de normais de superfície, que produz os resultados mais precisos; DSO - Odometria Esparsa Direta; SVO - Odometria Visual Semi-Direta	127

- Tabela 14 – Desempenho do método de SIDE proposto para diferentes configurações de arquitetura e estratégias de treinamento. A rede de extração de características da DSN 3 é o Modelo 3 desenvolvido. Ambos os módulos da pirâmide, além da função de custo $\mathcal{L}_{BerHu}^+$, são implementados como nas análises envolvendo os conjuntos de dados de ambientes externos. Os termos Sem e Indoor indicam o pré-treinamento da DSN 3 com mapas semânticos e mapas de profundidade extra, respectivamente, fornecidos pelos *datasets* de ambientes internos 130
- Tabela 15 – Análise do *pipeline* que explora pistas de normais de superfície. A configuração DSN 3 + Pyramid¹ + Indoor + SN, juntamente com a função de custo $\mathcal{L}_1^+$, é mantida em todos os experimentos. Legenda: Blur - convolução com kernel gaussiano; Borda - detector de bordas Sobel; DSN 3 - DSN com o Modelo 3 como *encoder*; Indoor - pré-treinamento com *datasets indoor*; ψ_1 e ψ_2 - constantes da função de custo 2,5D utilizada nas etapas de pré-treinamento e treinamento, respectivamente; Pyramid¹ - módulo do *decoder* em formato de pirâmide; SN - *ground truth* de normais de superfície 132
- Tabela 16 – Comparação quantitativa entre o método desenvolvido e outros que abordam a tarefa de estimação monocular de profundidade. Os resultados são avaliados no *dataset* NYU Depth V2 (SILBERMAN et al., 2012). DSN † refere-se à configuração da DSN que fornece os resultados gerais mais precisos 133
- Tabela 17 – Desempenho da CNN desenvolvida, bem como de outras técnicas, no *ground truth* do NYU Depth V2 (SILBERMAN et al., 2012). Legenda: DSN † - configuração da DSN que fornece os resultados gerais mais precisos; DSN ‡ - mesma configuração da DSN †, porém com o filtro Gaussiano; Mask - desconsideração dos pixels inválidos relacionados ao *ground truth* de normais de superfície 134
- Tabela 18 – Métricas do NYU Depth V2 (SILBERMAN et al., 2012) obtidas pelo *pipeline* de densificação de profundidade. O padrão DSN 3 + Pyramid¹ + Indoor, juntamente com a função $\mathcal{L}_1^+$, é selecionado para todos os experimentos que não envolvem o *ground truth* de normais de superfície. DSN 3 + Pyramid¹ + Indoor + SN † é a configuração da DSN com o módulo de normais de superfície que produz os resultados mais precisos 136
- Tabela 19 – Avaliação quantitativa do melhor *framework* de densificação de profundidade (DSN †) e de outras técnicas no estado-da-arte. O NYU Depth V2 (SILBERMAN et al., 2012) é utilizado para calcular todas as métricas 136

Tabela 20 – Análise das predições de normais de superfície da melhor configuração da FCN de densificação de profundidade (DSN †) quando recebe diferentes quantidades de amostras esparsas como entrada. Mask diz respeito a operação de desconsiderar os pixels inválidos correspondentes ao <i>ground truth</i> de normais de superfície	136
--	-----

LISTA DE ABREVIATURAS E SIGLAS

Abs Rel	Erro Relativo Absoluto
ADAM	Estimação de Momento Adaptativa
AdaDelta	Taxa de Aprendizagem Adaptativa
AdaGrad	Gradiente Adaptativo
AI	Inteligência Artificial
ANN	Rede Neural Artificial
ASPP	<i>Atrous Spatial Pyramid Pooling</i>
ATE	Erro Absoluto de Trajetória
B3DO	Berkeley 3-D Object Dataset
BA	<i>Bundle Adjustment</i>
BerHu	Função Inversa de Huber
Blur	Filtro Gaussiano
BRIEF	<i>Robust Independent Elementary Features</i>
Cal. Cam.	Câmera Calibrada
Cap	Limitação de Distância
CM	Movimento da Câmera
CNN	Rede Neural Convolutacional
CNN-FC	Rede Neural Convolutacional com Camadas Totalmente Conectadas
CPU	Unidade de Processamento Central
CRF	<i>Conditional Random Field</i>
CS	Cityscapes
CSPN	Rede Convolutacional de Propagação Espacial
D	Dimensões
DenseNet	Rede Convolutacional Densa

DCNN	Rede Neural Convolucional Profunda
DM	Mapa de Disparidade
DORN	<i>Deep Ordinal Regression Network</i>
ds	Bloco de Subamostragem
DSN	DenseSIDENet
DSN 3	DenseSIDENet com o Modelo 3 como <i>Encoder</i>
DSO	Odometria Esparsa Direta
DT	<i>Delaunay tessellation</i>
EQM	Erro Quadrático Médio
FC	Camadas Totalmente Conectadas
FC-LSTM	Camadas LSTM Totalmente Conectadas
FCN	Rede Totalmente Convolucional
fps	Quadros por Segundo
GAN	Rede Generativa Adversarial
GD	Gradiente Descendente
GPU	Unidade de Processamento Gráfico
GT	<i>Ground Truth</i>
HD	<i>Hard Drive</i>
ILSVRC	<i>Large Scale Visual Recognition Challenge</i>
iRMSE	Inverso da Raíz do Erro Quadrático Médio
it	Iterações
k	Kernel
KITTI	<i>Karlsruhe Institute of Technology and Toyota Technological Institute</i>
KNN	K Vizinhos Mais Próximos
LiDAR	<i>Light Detection and Ranging</i>
log10	Espaço Logarítmico na Base 10

LPG	Camadas de Orientação Planar Local
LSTM	<i>Long Short-Term Memory</i>
M	Milhões
m	metros
MAE	Erro Absoluto Médio
Mask	Máscara de Pixels Inválidos
MC	<i>Mirror Connection</i>
ML	Aprendizagem de Máquinas
MLP	Perceptron Multicamadas
MRF	<i>Markov Random Field</i>
MSE	Erro Quadrático Médio
MV	Vídeo Monocular
NADAM	<i>Nesterov-accelerated Adaptive Moment Estimation</i>
NYU	<i>New York University</i>
ORB	<i>Oriented Fast and Rotated BRIEF</i>
PAP	<i>Pattern-Affinitive Propagation</i>
Pyramid	Módulo de Pirâmide
RADAR	<i>Radio Detection And Ranging</i>
reduc1x1	Redução 1×1
ReLU	Unidade Linear Retificada
ResNet	Rede Neural Residual
RGB	<i>Red, Green, Blue</i>
RGBD	<i>Red, Green, Blue, Depth</i>
RMSE	Raiz do Erro Quadrático Médio
RMSE (log)	Raiz do Erro Quadrático Médio no Espaço Logarítmico
Sem	Mapas Semânticos

SfM	Estrutura de Movimento
SGD	Gradiente Descendente Estocástico
SGM	Correspondência Semi-global
SI	Imagem Única
SIDE	Estimação de Profundidade a partir de Uma Única Imagem
SIFT	<i>Scale Invariant Feature Transform</i>
SILog	Erro de Escala Invariante
SL	Rótulos Semânticos
SLAM	Localização e Mapeamento Simultâneos
SN	Normais de Superfície
SONAR	<i>Sound Navigation and Ranging</i>
SP	Par Estéreo
Sqr Rel	Erro Relativo Quadrático
SSD	<i>Solid State Drive</i>
SUN	<i>Scene Understanding</i>
SURF	<i>Speed Up Robust Feature</i>
SVO	Odometria Visual Semi-direta
Synt.	Sintético
VGG	<i>Visual Geometry Group</i>
ViT	Transformer Visual
VN	<i>Virtual Normals</i>
VO	Odometria Visual

LISTA DE SÍMBOLOS

D	Conjunto de dados de treinamento
$\mathbb{N}$	Conjunto dos números naturais
$\mathbb{R}$	Conjunto dos números reais
(x_i, y_i)	Conjunto rotulado
N	Número total de amostras de treinamento
$f : X \rightarrow Y$	Função de saída
$(x_1, x_2, \dots, x_n)$	Sinais de entrada de um neurônio artificial
$(w_1, w_2, \dots, w_n)$	Pesos sinápticos
Σ, θ	Combinação linear e limiar de ativação
$u, g(u)$	Potencial de ativação e função de ativação
$y_j(k), d_j(k)$	Sinal de saída e valor desejado de saída de uma rede neural artificial
$E(k)$	Função do erro de uma rede neural artificial
$\frac{1}{p} \sum_{k=1}^p E(k)$	Cálculo do erro quadrático médio
I_j	Sinal de entrada de um neurônio artificial de uma camada intermediária ou final da rede
∇	Operador gradiente
∂	Operador de derivada parcial
$\hat{y}_i$	Sinal de saída de um neurônio artificial de uma camada intermediária da rede
η	Taxa de aprendizagem
$x(a), w(t - a)$	Entrada e função contínua do kernel de uma operação de convolução contínua
$s(t)$	Mapa de características
$S(i, j)$	Convolução discreta
$I(i - m, i - n), K(m, n)$	Imagem discretizada e kernel discreto

$*$	Operador de convolução
$\max(0, x)$	Função de ativação ReLU
$\times$	Operador de multiplicação para escalares, vetores e matrizes
i, k	Tamanho da imagem de entrada e do kernel para a operação de <i>padding</i>
o, p	Convolução e quantidade de preenchimentos com zero para a operação de <i>padding</i>
pw	Probabilidade de desligamento de neurônios artificiais
γ, β	Parâmetros ajustáveis da normalização por lotes
$BN_{\gamma, \beta}(x_i)$	Transformação linear aplicada na normalização por lotes
$l(w)$	Função de custo geral
$\frac{1}{2} \lambda \sum_i w_i^2$	Termo de regularização <i>weight decay</i>
B	Tamanho de um <i>mini-batch</i>
$\nabla L(W; x_j^B)$	Cálculo do gradiente para o algoritmo SGD
α	Constante de ajuste do fator de Momentum
m, g	Estimações de momento de primeira e de segunda ordem do algoritmo de otimização ADAM
$\hat{m}, \hat{g}$	Aplicação de decaimento nos momentos de primeira e de segunda ordem
ϵ	Constante da regra de atualização do algoritmo de otimização ADAM
R_w^c, T_w^c	Matriz de rotação e vetor de translação em formato homogêneo
H_w^c	Matriz de transformação homogênea
$\hat{R}_w^c, \hat{T}_w^c$	Matriz de rotação quadrada 3×3 e vetor de translação 3×1
H_c^w	Matriz de transformação homogênea inversa
f	Distância focal da câmera
(s_x, s_y)	Dimensões horizontal e vertical dos pixels na imagem
(o_x, o_y)	Coordenadas do eixo óptico da câmera projetado na imagem
M_{int}	Matriz de parâmetros intrínsecos
P_w	Ponto no referencial do mundo

p_c, p_{im}	Projeção perspectiva e ponto no referencial da imagem
P_r, P_l	Ponto nos sistemas de coordenadas das câmeras direita e esquerda
p_r, p_l	Projeção do ponto nas câmeras direita e esquerda
T_l^r, R_l^r	<i>Baseline</i> e matriz de rotação do par de câmeras estéreo
S, E	Termo que depende dos parâmetros extrínsecos do par estéreo e matriz essencial
F	Matriz fundamental
e_r, e_l	Epipolos do sistema estéreo
$H_{\ell+1}(x_\ell)$	Função de transformação linear da ResNet
$H_{\ell+1}(x_0, x_1, \dots, x_\ell)$	Função de transformação linear da DenseNet
V, K, Q	Vetores <i>values</i> , <i>keys</i> e <i>queries</i> da rede Transformer
$F(\chi \mid \varphi)$	Função massa de probabilidade associada a variável aleatória χ dada uma probabilidade φ
Π	Operador produtório
d, d^*	Quantidade desejada de amostras e quantidade de dados válidos disponíveis por mapa de profundidade
$\mathcal{L}_{cce}$	Penalidade <i>categorical cross entropy</i>
(y, y^*)	Predição da rede e <i>ground truth</i> de treinamento para semântica e profundidade
$\frac{e^{t_i}}{\sum_j^K e^{t_j}}$	Função de ativação softmax
(v_i, ν_i)	Coordenadas dos pixels em uma imagem
(x_i, y_i, z_i)	Coordenadas de uma nuvem de pontos
ρ_i	Profundidade da cena
(f_x, f_y)	Distâncias focais dos eixos x e y
(s_x, s_y, s_z)	Coordenadas dos vetores normais a uma superfície
$D_{m \times 3}, c_{m \times 1}$	Matriz de coordenadas da nuvem de pontos e vetor com valores constantes
$\hat{s}_{3 \times m}$	Solução de mínimos quadrados

I_x, I_y	Gradiente de uma imagem em x e em y
H_x, H_y	Matrizes de aproximação do gradiente de uma imagem
H, Ψ	Magnitude e direção do gradiente de uma imagem
$G(m, n), I_g(i, j)$	Kernel gaussiano e imagem filtrada
σ	Desvio padrão
$\mathcal{L}_{2.5D}$	Função de custo geométrica 2,5D
ψ	Constante de proporcionalidade geométrica
$\Phi(n, n^*)$	Função que indica o ângulo entre planos
(n, n^*)	Predição da rede e <i>ground truth</i> de treinamento para normais de superfície
$\ n\ _2$	Norma vetorial
$\langle n, n^* \rangle$	Produto interno
$\mathcal{L}_{SILog}, \mathcal{L}_{SILog}^+$	Erro de escala invariante e erro de escala invariante modificado
$\mathcal{L}_{Huber}, \mathcal{L}_{BerHu}$	Funções de custo Huber e BerHu
$\mathcal{L}_1, \mathcal{L}_2$	Erro absoluto médio e erro quadrático médio
$\mathcal{L}_{charbo}$	Penalidade de Charbonnier
$\mathcal{L}_{cosh}$	Penalidade log-cosh
λ, θ	Constantes das funções de erro de escala invariante
h	Residual
$ h $	Módulo do residual
κ	Termo de balanceamento das funções Huber e BerHu
$\max_i$	Função <i>maximum</i>
a, α	Parâmetros de ajuste da penalidade de Charbonnier
$cosh$	Função cosseno hiperbólico
$\mathcal{L}_a$	Função de custo baseada em atenção
γ, ϵ	Termos de atenção e de regularização

$\mathcal{L}_1^+(h), \mathcal{L}_2^+(h)$	Erro absoluto médio e erro quadrático médio em formato de somatória
β_1, β_2	Constantes de regularização para os termos médios do algoritmo de otimização ADAM
T	Quantidade total de pixels com informações de profundidade
δ	Relação de precisão entre pixels
$\tilde{n}$	Limite de diferença angular
ψ_1, ψ_2	Constantes da função de custo geométrica empregadas nas etapas de pré-treinamento e treinamento

SUMÁRIO

1	INTRODUÇÃO	33
1.1	Objetivo	34
1.2	Organização do trabalho	35
2	REFERENCIAL TEÓRICO	37
2.1	Inteligência artificial	37
2.1.1	Aprendizagem de máquinas	38
2.1.2	Redes neurais artificiais	39
2.2	Aprendizagem profunda	43
2.2.1	Redes neurais convolucionais	45
2.2.1.1	Topologias de CNN	46
2.2.1.2	Unidade linear retificada (ReLU)	48
2.2.1.3	Subamostragem (<i>pooling</i>)	49
2.2.1.4	Preenchimento (<i>padding</i>)	49
2.2.1.5	Desligamento (<i>dropout</i>)	51
2.2.1.6	Normalização por lotes (<i>batch normalization</i>)	52
2.2.1.7	Regularização $\mathcal{L}_2$ (<i>weight decay</i>)	53
2.2.2	Algoritmos de otimização	54
2.2.3	<i>Frameworks</i> de desenvolvimento	56
2.3	Estimação de profundidade	56
2.3.1	Estimação monocular de profundidade	62
3	TRABALHOS CORRELATOS	67
3.1	Aprendizagem profunda e estimação monocular de profundidade	67
3.1.1	Aprendizado supervisionado	69
3.1.1.1	Temporalidade	73
3.1.1.2	Atenção	74
3.1.1.3	Profundidade e semântica	76
3.1.1.4	Profundidade e normais de superfície	77
3.1.2	Aprendizado semi-supervisionado	79
3.1.3	Aprendizado auto-supervisionado	80
3.1.3.1	Treinamento com imagens estéreo	80
3.1.3.2	Treinamento com vídeo monocular	82
3.2	Aprendizagem profunda e densificação de profundidade	83
3.2.1	Densificação de profundidade a partir de amostras esparsas	84
3.2.2	Densificação de profundidade guiada	85

3.3	Considerações finais	86
4	METODOLOGIA	91
4.1	Arquitetura de rede	91
4.1.1	Etapa de codificação da arquitetura de rede	93
4.1.2	Etapa de decodificação da arquitetura de rede	94
4.1.3	<i>Frameworks</i> derivados da arquitetura de rede	97
4.2	Funções de custo	102
4.3	Detalhes de implementação	104
4.4	Bases de dados	105
4.4.1	KITTI Depth	105
4.4.2	NYU Depth V2	107
4.5	Métricas de avaliação	109
5	RESULTADOS	111
5.1	Estudos de ablação em ambientes externos	111
5.1.1	Análises preliminares	111
5.1.2	Estudos estruturais	114
5.1.3	Análises de normais de superfície	117
5.1.4	Estudos sobre densificação de profundidade	122
5.1.5	Resultados de odometria visual	126
5.2	Estudos de ablação em ambientes internos	128
5.2.1	Estudos estruturais	128
5.2.2	Análises de normais de superfície	129
5.2.3	Estudos de densificação de profundidade	134
6	CONCLUSÃO	143
6.1	Publicação	144
	REFERÊNCIAS	145

1 INTRODUÇÃO

Tecnologias de navegação autônoma estão cada vez mais presentes em pesquisas acadêmicas, nas indústrias, na agricultura e, em menor ocorrência, nas ruas e em residências para o auxílio em tarefas cotidianas. No entanto, existem muitos desafios que ainda precisam ser solucionados a fim de tornar os veículos autônomos uma realidade para a população. Um desses desafios está relacionado ao aperfeiçoamento de sistemas de percepção que integram as plataformas robóticas e que são responsáveis pelo mapeamento do ambiente em que o veículo está inserido. Tal percepção ajuda o veículo a ter uma compreensão do espaço 3D e, conseqüentemente, a ter uma noção da distância em que se encontram os objetos presentes na cena, possibilitando assim o desvio autônomo de obstáculos e o tráfego seguro em regiões navegáveis de diversos tipos de cenários como em cidades ou em estradas (MANCINI et al., 2017).

Sendo assim, é imprescindível que as plataformas robóticas tenham a capacidade de explorar pistas de profundidade do ambiente, as quais beneficiam outras tarefas de percepção como: segmentação semântica (EIGEN; FERGUS, 2015), detecção de objetos (CHANG; WETZSTEIN, 2019), estimação de planos (WANG et al., 2016), estimação de contornos de oclusão (RAMAMONJISOA; LEPETIT, 2019), Odometria Visual (VO) (LOO et al., 2019) e Localização e Mapeamento Simultâneos (SLAM) (TATENO et al., 2017). De acordo com Eigen, Puhersch e Fergus (2014), “a estimação de profundidade é um componente importante para entender as relações geométricas dentro de uma cena. Por sua vez, essas relações ajudam a fornecer representações mais ricas de objetos e seu ambiente, muitas vezes levando a melhorias nas tarefas de reconhecimento existentes”.

Sensores RGBD - *Red, Green, Blue, Depth* - (Kinect v1, Kinect v2 (PAGLIARI; PINTO, 2015) e FARO Focus (VASILJEVIC et al., 2019)), *Light Detection and Ranging* (LiDAR), câmeras estéreo, *Radio Detection And Ranging* (RADAR) e *Sound Navigation and Ranging* (SONAR) são tecnologias que desempenham o papel de extrair dados de profundidade do ambiente e, além disso, são amplamente comercializadas. No entanto, existem limitações acerca da utilização das tecnologias apresentadas para a tarefa de percepção de profundidade, visto que: sensores LiDAR são proibitivos em termos de custo por unidade e realizam medições de profundidade esparsas e ruidosas a longas distâncias; sensores RGBD são sensíveis a luz, principalmente em regiões reflexivas, e têm um limite de medição de distância menor quando comparados ao LiDAR; câmeras estéreo geram falhas em regiões refletoras e de baixa textura; e os sensores RADAR e SONAR são, geralmente, auxiliares de outros dispositivos principais de medição de profundidade como o LiDAR (MA; KARAMAN, 2018).

Tendo isso em vista, a utilização de câmeras monoculares, no contexto de percepção

robótica, se torna vantajosa, sabendo-se que elas demandam um espaço de instalação menor, são de baixo custo, podem capturar imagens coloridas e em preto e branco, são robustas a maioria das condições ambientes e são eficientes em termos de consumo de energia (MA; KARAMAN, 2018). Logo, conclui-se que a tarefa de SIDE é importante e desafiadora na área de navegação robótica autônoma, uma vez que veículos autônomos necessitam de informações de profundidade consistentes para navegar em diversos locais.

Recentemente, o avanço de Redes Neurais Convolucionais (CNNs), além da disponibilidade de bases de dados construídas com sensores do tipo *structured-light/time-of-flight/phase-shift* e LiDAR, motivou o surgimento de novos trabalhos no âmbito de estimação monocular de profundidade. No entanto, os trabalhos de SIDE que estão no estado-da-arte utilizam *ground truth* esparsos e ruidosos para treinar redes e não se beneficiam, totalmente, das maneiras com que se pode adicionar informações de profundidade, de semântica, de normais de superfície e até mesmo de atenção para melhorar o desempenho do método em relação a velocidade e a acurácia das predições.

Acrescentando-se a isso, redes profundas aplicadas para a tarefa em questão ainda apresentam erros de estimação que podem chegar a 50cm em ambientes internos e a 4m em ambientes externos (MA; KARAMAN, 2018). Portanto, tem-se que a Estimação de Profundidade a partir de Uma Única Imagem (SIDE) é um tema de extrema importância no contexto de veículos autônomos e também é um ponto aberto a inovações, principalmente devido ao surgimento de novas técnicas de Aprendizagem Profunda, a disponibilidade de bases de dados específicas para esta aplicação e ao aumento de poder computacional impulsionado por Unidades de Processamento Gráfico (GPUs).

1.1 Objetivo

Como um dos objetivos gerais, este trabalho visa introduzir uma arquitetura de Rede Neural Convolucional Profunda (DCNN), totalmente convolucional, supervisionada e com formato codificação-decodificação, aplicada a tarefa de SIDE. Para o treinamento desta rede, foca-se em desenvolver funções de custo baseadas em atenção que se concentram em regiões e estruturas específicas da imagem, uma vez que a distribuição de pixels com informações de profundidade é desigual e que as cenas apresentam fortes pistas geométricas (JIAO et al., 2018).

Em um outro foco, este trabalho objetiva aplicar métodos de Visão Computacional para melhorar os resultados da DCNN com relação a qualidade das predições e a precisão dos resultados. Dentre tais métodos, é possível citar as técnicas de densificação de mapas de profundidade (GANSBEKE et al., 2019), de fusão de informações semânticas (MOUSAVIAN; PIRSIYAVASH; KOŠECKÁ, 2016) e de profundidade (LIAO et al., 2017), normais de superfície (YIN et al., 2019), *Atrous Spatial Pyramid Pooling* (ASPP) (CHEN et al., 2017), *skip-connections* (LEE et al., 2019), *encoders* multi-escala (XU et al., 2017) e

deconvoluções multi-camadas (GARG et al., 2016).

A arquitetura de rede desenvolvida foi treinada e avaliada em ambientes internos por meio do *dataset* NYU Depth V2 (SILBERMAN; FERGUS, 2011) e em ambientes externos por meio do *dataset* KITTI Depth (UHRIG et al., 2017). Assim, como objetivo secundário, este trabalho será comparado com outros que estão no estado-da-arte com o auxílio das métricas que são amplamente difundidas dentro do contexto de estimação monocular de profundidade.

1.2 Organização do trabalho

O texto da dissertação está organizado da seguinte maneira: o Capítulo 2 introduz a teoria que envolve as ferramentas e os algoritmos empregados no método desenvolvido. O Capítulo 3 expõe os trabalhos relacionados, principalmente, aqueles que estão no estado-da-arte da literatura correspondente a aplicação deste trabalho. No Capítulo 4, são apresentadas e detalhadas as técnicas e os procedimentos que serão empregados para solucionar o problema. Finalmente, o Capítulo 5 mostra os resultados obtidos e o Capítulo 6 apresenta a conclusão do trabalho, bem como os artigos publicados.

2 REFERENCIAL TEÓRICO

2.1 Inteligência artificial

Em uma concepção sobre inteligência de máquinas, Turing (1950) estabeleceu que “o comportamento inteligente presumivelmente consiste em um afastamento do comportamento completamente disciplinado envolvido na computação, mas um comportamento bastante leve, que não dá origem a um comportamento aleatório ou a *loops* repetitivos inúteis”. Em termos mais diretos, segundo Nilsson e Nilsson (1998) , “a Inteligência Artificial (AI), definida de maneira ampla, preocupa-se com o comportamento inteligente nos artefatos. O comportamento inteligente, por sua vez, envolve percepção, raciocínio, aprendizado, comunicação e atuação em ambientes complexos”.

Tendo isso em vista, de acordo com Genesereth e Nilsson (2012), a Inteligência Artificial, no campo da engenharia, visa o desenvolvimento de máquinas inteligentes a partir dos conceitos e aplicações que as regem. Como exemplos de máquinas inteligentes, é possível citar os sistemas de processamento de linguagem natural, utilizados para compreender trechos de linguagem escrita e falada, sistemas automáticos de consulta especializada, que podem identificar doenças e até mesmo determinar potenciais depósitos de minério, e os sistemas de prova de teorema, cuja função é deduzir teoremas a partir de outros teoremas previamente provados (NILSSON, 2014). Tais sistemas emulam comportamentos inteligentes através de processos computacionais (SCHALKOFF, 1990).

No contexto científico, a AI pode ser descrita como uma teoria da inteligência que compreende o comportamento inteligente observado em pessoas e em outros animais, bem como em máquinas artificialmente desenvolvidas, por meio da formulação de termos e conceitos que ajudam a atingir esse grau de entendimento (GENESERETH; NILSSON, 2012). Por outro lado, segundo (RUSSELL; NORVIG, 2016), as definições de AI podem abordar questões de raciocínio e procedimentos de aquisição de conhecimento, bem como questões comportamentais. Tais definições também podem considerar aspectos da ação humana e conceitos de inteligência. Sendo assim, é possível apresentar os seguintes fragmentos que descrevem AI:

- “A automação de atividades que associamos ao pensamento humano, atividades como tomada de decisão, resolução de problemas, aprendizagem...” (BELLMAN, 1978)
- “O novo e empolgante esforço para fazer os computadores pensarem... máquinas com mentes, em sentido completo e literal.” (HAUGELAND, 1989)

- “O estudo das faculdades mentais através da utilização de modelos computacionais.” (CHARNIAK, 1985)
- “A arte de criar máquinas que executam funções que requerem inteligência quando executadas por pessoas.” (KURZWEIL et al., 1990)
- “O estudo das computações que possibilitam perceber, raciocinar e agir.” (WINSTON, 1992)
- “O ramo da ciência da computação que se preocupa com a automação do comportamento inteligente.” (LUGER; STUBBLEFIELD, 1993)

Portanto, este trabalho aborda o objetivo de AI como engenharia, em que se busca construir mecanismos inteligentes capazes de perceber, representar, estimar e aprender quando estão em contato com o meio.

2.1.1 Aprendizagem de máquinas

A Aprendizagem de Máquinas (ML) é uma subdivisão da AI que está estreitamente relacionada aos campos de mineração de dados e estatística. Em consequência do constante aumento da quantidade e da disponibilidade de dados em diferentes áreas de pesquisa, métodos capazes de extrair padrões e aprender de forma automática são cada vez mais requisitados (MURPHY, 2012). Essas técnicas que analisam dados de forma automática são caracterizadas como métodos de Aprendizagem de Máquinas. De acordo com Murphy (2012), “a aprendizagem de máquinas é um conjunto de métodos que podem detectar padrões em dados automaticamente e, em seguida, usar os padrões descobertos para prever dados futuros ou para realizar outros tipos de tomada de decisão sob incerteza”.

Problemas que envolvem reconhecimento de padrões, no campo da aprendizagem de máquinas, podem ser categorizados em três tipos principais: Supervisionados, Não Supervisionados e Aprendizagem por Reforço. Segundo Murphy (2012), resolver problemas da categoria supervisionada consiste em aprender a relacionar as entradas x as suas respectivas saídas y , uma vez que é dado o conjunto rotulado da Eq. 2.1, onde D corresponde ao conjunto de dados de treinamento e N diz respeito ao número total de amostras de treinamento.

$$D = \{(x_i, y_i)\}_{i=1}^N \quad (2.1)$$

As entradas x_i , compreendidas no espaço de instâncias rotuladas, podem ser tanto vetores D-dimensionais, chamados de atributos, quanto estruturas de dados mais complexas, como vetores de características, imagens e até séries temporais. Por outro lado, as saídas y_i , referentes ao espaço de rótulos, podem ser representadas, na maioria das aplicações,

por variáveis categóricas (em um conjunto finito de categorias $y_i \in 1, \dots, n$ para $n \in \mathbb{N}$), ou por variáveis reais, em que y_i assume valores contínuos ($y_i \in \mathbb{R}$). Quando y_i é categórico, tem-se um problema de classificação ou de reconhecimento de padrões e, quando o y_i é real, tem-se um problema de regressão. Se y_i é real e ordenado, trata-se de um problema de regressão ordinal (MURPHY, 2012).

De acordo com Bishop (2006), o conjunto de dados de treinamento com N amostras é utilizado para ajustar os parâmetros do modelo adaptativo aplicado na resolução do problema supervisionado. Assim, na etapa de treinamento, o algoritmo de aprendizagem de máquinas visa determinar uma função de saída $f : X \rightarrow Y$. Após o treinamento, o modelo pode ser aplicado para o conjunto de teste, no qual se encontram novas variáveis de entrada.

Em problemas Não Supervisionados, apenas a entrada é conhecida, Eq. 2.2, não havendo um conjunto rotulado do tipo entrada-saída para treinar o modelo. Desta forma, esse tipo de problema é solucionado através da determinação de padrões nos dados de entrada, cujas relações não são previamente fornecidas (BISHOP, 2006).

$$D = \{x_i\}_{i=1}^N \quad (2.2)$$

Finalmente, os métodos de Aprendizagem por Reforço consistem na busca pela maximização da recompensa por meio do processo de tomada de ações que levam a estados distintos em um ambiente específico. Tais algoritmos são fundamentados no conceito de tentativa-e-erro, em que é necessário interagir com o meio para encontrar a sequência de ações e estados ótimos (BISHOP, 2006).

No próximo tópico são apresentados os algoritmos de Aprendizagem de Máquinas denominados de Redes Neurais Artificiais (com foco apenas na aprendizagem supervisionada), as quais são conceitos básicos para redes mais complexas como as CNNs.

2.1.2 Redes neurais artificiais

Segundo Silva, Spatti e Flauzino (2010), as Redes Neurais Artificiais (ANNs) são modelos matemáticos implementados computacionalmente que buscam reproduzir o funcionamento do sistema nervoso de seres vivos. As ANNs começaram ser alvo de pesquisas, de uma maneira mais frequente, no começo da década de 90 e ainda são amplamente estudadas em aplicações de: reconhecimento e classificação de padrões, controle de sistemas robóticos e de aparelhos eletrônicos, agrupamento de dados, memórias associativas, entre outros (SILVA; SPATTI; FLAUZINO, 2010). Ademais, quando aplicadas, as ANNs apresentam as seguintes características: adaptação por experiência, capacidade de aprendizado, habilidade de generalização, organização de dados, tolerância a falhas, armazenamento distribuído e facilidade de prototipagem (SILVA; SPATTI; FLAUZINO, 2010).

O sistema nervoso de seres vivos é composto por neurônios que são células capazes de captar, processar e transmitir informações, no formato de impulsos elétricos (gerados por processos físico-químicos), por meio de suas estruturas elementares: dendritos, corpo celular e axônio respectivamente (SILVA; SPATTI; FLAUZINO, 2010). A partir disso, tem-se que o funcionamento dos neurônios artificiais, que são as unidades básicas que processam informação em uma ANN, é baseado na forma como os neurônios biológicos operam, porém de uma maneira mais simplificada. McCulloch e Pitts (1943) foram responsáveis pela elaboração da primeira estrutura de neurônio artificial, o qual, como se pode observar pela Fig. 1, é composto por sinais de entrada $[x_1, x_2, \dots, x_n]$, pesos sinápticos $[w_1, w_2, \dots, w_n]$, combinação linear $[\Sigma]$, limiar de ativação $[\theta]$ e potencial de ativação $[u]$, função de ativação $[g]$ e sinal de saída $[y]$.

Figura 1 – Representação da rede Perceptron

Fonte: Silva, Spatti e Flauzino (2010)

Assim, durante o processo de treinamento, esse tipo de neurônio recebe dados para cada entrada, os quais são ponderados pelos seus respectivos pesos sinápticos. Posteriormente, a soma ponderada das variáveis de entrada é subtraída do limiar de ativação, como se pode observar pela Eq. 2.3. Por fim, o sinal de saída do neurônio é limitado pela aplicação de uma função de ativação nos valores obtidos para o potencial de ativação. As funções de ativação mais utilizadas são apresentadas na Fig. 2.

$$u = \sum_{i=1}^n w_i \cdot x_i - \theta \quad (2.3)$$

$$y = g(u)$$

Quanto à arquitetura, as ANNs clássicas apresentam uma camada de entrada, onde chegam os dados para a rede (não há neurônios), camadas escondidas, onde são extraídas características dos dados de entrada, e uma camada de saída que é constituída pelos neurônios que geram as predições da rede. A Fig. 1 ilustra uma arquitetura de rede do tipo *feedforward* (fluxo de dados ocorre somente em uma direção) com apenas uma camada neural (camada simples). Tal arquitetura pode ser observada nas redes Perceptron

Figura 2 – Principais funções de ativação utilizadas em ANNs

Fonte: Kopiler et al. (2019)

(ROSENBLATT, 1958) e Adaline (WIDROW; HOFF, 1960), as quais são geralmente utilizadas em aplicações de classificação de padrões.

Por outro lado, a Fig. 3 mostra uma arquitetura *feedforward* de camadas múltiplas composta por uma ou mais camadas intermediárias de neurônios. A rede Perceptron multicamadas (MLP) é a arquitetura de camadas múltiplas mais utilizada, uma vez que ela pode ser aplicada para resolver uma maior variedade de problemas, com relação às arquiteturas de camada simples, tais quais: classificação de padrões, aproximação universal de funções, previsão de séries temporais, otimização de sistemas e controle de processos (SILVA; SPATTI; FLAUZINO, 2010). Essa flexibilidade em resolver diversos problemas se dá, principalmente, devido ao maior número de camadas escondidas e ao método de aprendizagem das MLPs, denominado de regra Delta generalizada ou algoritmo de *backpropagation*, introduzido por Rumelhart, Hinton e Williams (1985), o qual é utilizado na fase de treinamento da rede.

O algoritmo de *backpropagation* visa minimizar o erro entre a saída da rede e os valores esperados (Eq. 2.4) por meio do ajuste dos pesos sinápticos e dos limiares de ativação. Na Eq. 2.4, $d_j(k)$ diz respeito ao valor desejado de saída e $y_j(k)$ refere-se a predição da rede gerada pelo j – ésimo neurônio da camada de saída (com n neurônios), para um k – ésimo dado de treinamento. Em um conjunto de treinamento que contém p dados, o algoritmo de retropropagação do erro considera a Eq. 2.5 que representa o erro

Figura 3 – Ilustração da rede Perceptron multicamadas

Fonte: Silva, Spatti e Flauzino (2010)

quadrático médio da Eq. 2.4.

$$E(k) = \frac{1}{2} \sum_{j=1}^n (d_j(k) - y_j(k))^2 \quad (2.4)$$

$$E = EQM(k) = \frac{1}{p} \sum_{k=1}^p E(k) \quad (2.5)$$

Aplicando o gradiente na Eq. 2.5, obtem-se a Eq. 2.6, em que I_j diz respeito ao sinal de entrada do j – ésimo neurônio da camada de saída e $w_{j,i}$ representa o vetor de pesos sinápticos que correlaciona tal neurônio de saída ao i – ésimo neurônio da camada anterior.

$$\nabla E = \frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial I_j} \cdot \frac{\partial I_j}{\partial w_{ji}} \quad (2.6)$$

Os termos que compõem a expressão do gradiente do EQM geram as igualdades expostas nas Eqs. 2.7, 2.8 e 2.9, onde $\hat{y}_i$ equivale ao valor de saída do neurônio da camada anterior.

$$\frac{\partial I_j}{\partial w_{ji}} = \hat{y}_i \quad (2.7)$$

$$\frac{\partial y_j}{\partial I_j} = g'(I_j) \quad (2.8)$$

$$\frac{\partial E}{\partial y_j} = -\frac{1}{p} \sum_{k=1}^p (d_j(k) - y_j(k)) \quad (2.9)$$

Substituindo os resultados de tais equações na Eq. 2.6, pode-se reescrever a expressão do gradiente de acordo com a Eq. 2.10.

$$\nabla E = -\frac{1}{p} \sum_{k=1}^p (d_j(k) - y_j(k)) \cdot g'(I_j) \cdot \hat{y}_i \quad (2.10)$$

Sabendo-se que os parâmetros livres da rede são ajustados de modo a seguir a direção oposta ao vetor gradiente, que aponta para o máximo da função do EQM, tem-se que a Eq. 2.11 descreve a expressão de atualização dos pesos sinápticos, a qual é multiplicada por uma taxa de aprendizagem η . Em formato de algoritmo, a expressão do ajuste dos pesos pode ser reformulada para gerar a Eq. 2.12.

$$\Delta w_{ji} = -\eta \nabla E = \eta \cdot \frac{1}{p} \sum_{k=1}^p (d_j(k) - y_j(k)) \cdot g'(I_j) \cdot \hat{y}_i \quad (2.11)$$

$$w_{ji} \leftarrow w_{ji} + \eta \cdot \delta_j \cdot \hat{y}_i \quad (2.12)$$

$$\delta_j = \frac{1}{p} \sum_{k=1}^p (d_j(k) - y_j(k)) \cdot g'(I_j)$$

Descritos os fundamentos de ML, a próxima seção introduz os conceitos essenciais da área de Aprendizagem Profunda. Dentre tais conceitos, ressaltam-se as CNNs que são as principais ferramentas para a estruturação do método deste e de outros trabalhos na literatura de SIDE.

2.2 Aprendizagem profunda

De acordo com Goodfellow et al. (2016), Aprendizagem Profunda é uma área da Inteligência Artificial, especificamente de Aprendizagem de Máquinas, que objetiva a construção de modelos de redes neurais complexas para extrair representações abstratas de imagens, de níveis baixos até níveis mais altos de abstração, por meio do acúmulo de experiências com os dados. Segundo LeCun, Bengio e Hinton (2015), as redes de Aprendizagem Profunda utilizam a regra Delta generalizada para atualizar seus parâmetros internos, os quais são responsáveis pela formação da hierarquia de representações.

O Diagrama de Venn da Fig. 4 mostra que a Aprendizagem Profunda é uma subdivisão da área de Aprendizagem de Representação, consequentemente de Aprendizagem de Máquinas, a qual é caracterizada por técnicas capazes de computar representações dos dados de entrada e aplicá-las em tarefas de classificação e detecção. Sendo assim, conclui-se que as redes de Aprendizagem Profunda são tipos de métodos de Aprendizagem de Representação, porém com mais níveis de representação referentes às múltiplas camadas neurais de processamento (LECUN; BENGIO; HINTON, 2015).

Figura 4 – Subdivisões do campo da Inteligência Artificial

Fonte: Adaptada de Goodfellow et al. (2016)

Em virtude do surgimento de Unidades de Processamento Central (CPUs) mais rápidas e de GPUs, além do aumento de dados disponíveis em meios digitais, as redes de Aprendizagem Profunda têm obtido sucesso quando aplicadas na resolução de diferentes problemas antes solucionados por métodos clássicos de Visão Computacional e de Aprendizagem de Máquinas. Dentre tais problemas, é possível citar: estimação de profundidade (EIGEN; PUHRSCH; FERGUS, 2014), segmentação de instâncias (HE et al., 2017), determinação de pose (TOMPSON et al., 2014), reconhecimento de imagens (FARABET et al., 2012) e de textos (HINTON et al., 2012) e, até mesmo, problemas relacionados à: medicina (MA et al., 2015), biologia (XIONG et al., 2015), física (ADAM-BOURDARIOS et al., 2015) e linguística (SUTSKEVER; VINYALS; LE, 2014).

Na área de reconhecimento de imagens, os avanços nos métodos de Aprendizagem Profunda fizeram com que uma rede neural convolucional ganhasse o campeonato Imagenet, *Large Scale Visual Recognition Challenge* (ILSVRC) (RUSSAKOVSKY et al., 2015), pela primeira vez, diminuindo o erro top-5 no estado-da-arte de 26,1% para 15,3% (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). Desta forma, devido aos resultados cada vez melhores gerados por redes neurais convolucionais em diversas tarefas (YU; DENG; DAHL, 2010; SERMANET et al., 2013; CIREŞAN et al., 2012), empresas como: NVIDIA, Google, Apple, Microsoft e Netflix têm se beneficiado do potencial lucrativo de tais redes

(LECUN; BENGIO; HINTON, 2015).

2.2.1 Redes neurais convolucionais

CNNs (LECUN et al., 1989) são arquiteturas de rede adaptativas de Aprendizagem Profunda, capazes de aprender características espaciais de dados em um formato padrão de grade, como imagens, vídeos e séries temporais, de maneira automática. Tais redes são inspiradas na estrutura do córtex visual de animais (FUKUSHIMA, 1980) e elas diferenciam-se de redes neurais clássicas pela utilização da operação linear de convolução em ao menos uma de suas camadas (GOODFELLOW et al., 2016).

A operação de convolução, no plano contínuo, pode ser expressa pela Eq. 2.13, cuja forma reduzida é apresentada na Eq. 2.14. Em aplicações de redes neurais convolucionais, a função $x(a)$ diz respeito à entrada da operação de convolução, a função $w(t - a)$ refere-se ao *kernel* e $s(t)$ é denominado de mapa de características. No entanto, para que a operação de convolução seja utilizada em CNNs, a Eq. 2.13 e a Eq. 2.14 devem ser expressas no plano discreto. Assim, tem-se que a Eq. 2.15 representa a operação de convolução discreta, em que, no campo de Aprendizagem Profunda, a entrada e o *kernel* possuem formato de vetor multidimensional (tensor). Os valores do tensor que diz respeito ao *kernel* da convolução são atualizados durante o processo de execução do algoritmo de *backpropagation* (GOODFELLOW et al., 2016).

$$s(t) = \int x(a)w(t - a)da \quad (2.13)$$

$$s(t) = (x(a) * w(t - a))(t) \quad (2.14)$$

$$s(t) = (x(a) * w(t - a))(t) = \sum_{a=-\infty}^{\infty} x(a)w(t - a) \quad (2.15)$$

Uma vez que as CNNs recebem dados em formato de grade, considera-se que as operações de convolução são executadas de maneira multidimensional e em um número finito de dados, os quais correspondem às partes onde há informação no tensor de entrada da convolução (esparsidade). Assim, para uma imagem I , utiliza-se um *kernel* bidimensional K na convolução discreta da Eq. 2.16. Beneficiando-se da propriedade comutativa da convolução, pode-se determinar a relação da Eq. 2.17, a qual é mais utilizada em algoritmos de Aprendizagem Profunda (GOODFELLOW et al., 2016).

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n) \quad (2.16)$$

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i - m, j - n) K(m, n) \quad (2.17)$$

Como as convoluções discretas, além de serem esparsas, reutilizam os parâmetros do *kernel* em mais de uma região da entrada, é possível ilustrar essa operação por meio da Fig. 5. Nessa figura, observa-se que os valores dos elementos do *kernel* da convolução, grade 3×3 cinza, são aplicados a todas as regiões da imagem de entrada, grade 5×5 azul. O *kernel* segue um passo com valor 1 e as grades verdes representam os mapas de características obtidos na saída da convolução discreta, os quais são formados pela multiplicação dos elementos do *kernel* pelos elementos da imagem, quando estes são sobrepostos pela grade cinza. Desta forma, os resultados da multiplicação são somados para formar o mapa de características de saída. Para uma imagem com mais de uma dimensão, utiliza-se um *kernel* com o mesmo número de dimensões. Nesse caso, as operações de convolução discreta são executadas para cada dimensão separadamente (DUMOULIN; VISIN, 2016).

Figura 5 – Operação de convolução discreta

Fonte: Adaptada de Dumoulin e Visin (2016)

2.2.1.1 Topologias de CNN

Com respeito a estrutura, as CNNs podem ser divididas em uma etapa inicial de codificação, seguida de uma etapa de decodificação. Na etapa de codificação, são extraídos

mapas de características, de menor tamanho e com diferentes níveis de abstração, da imagem de entrada. Posteriormente, a etapa de decodificação é responsável por processar os mapas de características, produzidos na codificação, e gerar a saída desejada da rede para diferentes aplicações. Dependendo da aplicação, as topologias de CNNs, geralmente, apresentam modificações estruturais na etapa de decodificação, de modo a ajustar a saída da rede para retornar dados em formato de grade de maneiras distintas. A partir disso, de acordo com a topologia, as CNNs podem ser classificadas como totalmente convolucionais (FCN) ou contendo camadas totalmente conectadas (CNN-FC).

As FCN são compostas somente por camadas convolucionais na decodificação. No entanto, tais camadas são normalmente utilizadas em conjunto com operações de sobre-amostragem, como as operações bilinear e vizinhos mais próximos, para aumentar o tamanho da imagem na saída da rede. Os blocos constituídos de convolução e operação de sobre-amostragem são denominados deconvolução. A Fig. 6 ilustra um exemplo de topologia FCN (BADRINARAYANAN; KENDALL; CIPOLLA, 2017) aplicado na tarefa de segmentação semântica para navegação autônoma.

Figura 6 – Exemplo de topologia FCN (SegNet)

Fonte: Badrinarayanan, Kendall e Cipolla (2017)

Em contrapartida, as CNN-FC utilizam camadas neurais totalmente conectadas para decodificar mapas de características. Nesse caso, a etapa de decodificação apresenta as mesmas subdivisões estruturais de um MLP, em que o número de saídas da rede é limitado pela quantidade de neurônios presentes na camada totalmente conectada final. A Fig. 7 mostra um exemplo de topologia CNN-FC (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) utilizado para classificar imagens no campeonato ImageNet LSVRC-2010.

Nas próximas seções, são introduzidos os elementos que compõem as arquiteturas de CNNs. Especificamente, descreve-se os seguintes tópicos: unidade linear retificada, subamostragem, preenchimento, desligamento, normalização por lotes, regularização $\mathcal{L}_2$ e algoritmos de otimização. Tais elementos são indispensáveis para o entendimento de CNNs de um modo geral.

Figura 7 – Exemplo de topologia CNN-FC (AlexNet)

Fonte: Krizhevsky, Sutskever e Hinton (2012)

2.2.1.2 Unidade linear retificada (ReLU)

A Unidade Linear Retificada é uma função de ativação frequentemente utilizada em redes de Aprendizagem Profunda, uma vez que ela gera impactos positivos quando implementada no treinamento de tais redes. Um desses impactos está vinculado ao comportamento da ReLU em todo seu domínio, já que ela não satura como outras funções (sigmoidais) (PONTI; COSTA, 2018).

Para valores positivos de entrada, a ReLU se comporta como a função identidade e, para entradas negativas, ela retorna apenas valores nulos. Outro importante impacto está relacionado ao fato da ReLU evitar o processo de desaparecimento do gradiente (*vanishing gradient problem*) que ocorre quando este assume valores próximos a zero, dificultando assim o ajuste dos pesos da rede. A Fig. 8 fornece uma ilustração gráfica da função de ativação apresentada (PONTI; COSTA, 2018).

Figura 8 – Comportamento da função ReLU

Fonte: Adaptada de Ponti e Costa (2018)

A Fig. 9 expõe o comportamento de outras funções de ativação derivadas da função ReLU. O formato de tais funções varia, principalmente, nas regiões que tratam de valores

negativos da entrada. Essas variações podem mitigar os problemas de mudança covariável interna, de explosão e desaparecimento do gradiente e de *overfitting*.

Figura 9 – Funções de ativação para CNNs

Fonte: Adaptada de Zhang e Li (2018)

2.2.1.3 Subamostragem (*pooling*)

De acordo com Dumoulin e Visin (2016), a técnica de subamostragem utiliza operações estatísticas para condensar regiões de um mapa de características. Em termos práticos, tal técnica funciona como uma convolução, porém as operações estatísticas substituem as combinações lineares executadas pelo *kernel* da convolução. Como exemplo de tais operações, a subamostragem pelo máximo (*max pooling*) e pela média (*average pooling*), retornam o valor máximo e o valor médio de uma vizinhança retangular na imagem respectivamente (GOODFELLOW et al., 2016).

Segundo Goodfellow et al. (2016), se o mapa de características de entrada de uma subamostragem sofrer uma translação, a maior parte dos valores do mapa de saída permanecerá a mesma, comparando-se com o caso de não ocorrer translação. Logo, conclui-se que, por meio da técnica em questão, os mapas de características se tornam invariantes a translações locais. Isso é benéfico para as situações em que é mais vantajoso saber da presença de uma característica do que sua localização em uma CNN. A Fig. 10 apresenta o funcionamento de uma subamostragem pelo máximo com janela deslizante de dimensões 3×3 e passo 1 em uma imagem de tamanho 5×5 (DUMOULIN; VISIN, 2016).

2.2.1.4 Preenchimento (*padding*)

A operação de preenchimento é essencial para uma CNN, uma vez que, a partir dela, é possível modificar o tamanho do *kernel* sem alterar as dimensões do mapa de características na saída da convolução (GOODFELLOW et al., 2016). Na Fig. 11, tem-se o caso de uma convolução com *kernel* 3×3 e passo igual a 1 sendo aplicada em uma imagem 4×4 sem preenchimento. Para este caso, a resolução do mapa de saída pode ser

Figura 10 – Funcionamento da Subamostragem

Fonte: Adaptada de Dumoulin e Visin (2016)

obtida pela Eq. 2.18, em que i diz respeito ao tamanho da imagem de entrada e k refere-se ao tamanho do *kernel*. Por meio da Eq. 2.18 e da Fig. 11, verifica-se que o resultado da convolução aplicada a uma imagem sem preenchimento apresenta uma resolução menor que a da própria imagem (DUMOULIN; VISIN, 2016).

Figura 11 – Aplicação da operação de convolução em uma imagem sem preenchimento

Fonte: Adaptada de Dumoulin e Visin (2016)

$$o = (i - k) + 1 \quad (2.18)$$

Por outro lado, a Fig. 12 ilustra a operação de convolução aplicada a uma imagem com preenchimento semelhante. Nessa situação, a relação que determina o tamanho do produto da convolução é representada pela Eq. 2.19, para $k = 2n + 1$ e $p = \frac{k}{2} = n$ ($n \in \mathbb{N}$), em que p corresponde a quantidade de preenchimentos com zero. Desta forma, com base na Eq. 2.19 e na Fig. 12, nota-se que a imagem de entrada possui o mesmo

tamanho do mapa de características de saída. Além do preenchimento semelhante, existem os preenchimentos arbitrário e completo descritos pelas Eqs. 2.20 e 2.21 respectivamente (DUMOULIN; VISIN, 2016).

Figura 12 – Aplicação da operação de convolução em uma imagem com preenchimento semelhante

Fonte: Adaptada de Dumoulin e Visin (2016)

$$o = i + 2 \left\lfloor \frac{k}{2} \right\rfloor - (k - 1) \quad (2.19)$$

$$o = i + 2n - 2n$$

$$o = i$$

$$o = (i - k) + 2p + 1 \quad (2.20)$$

$$o = i + 2(k - 1) - (k - 1) \quad (2.21)$$

$$o = i + (k - 1)$$

2.2.1.5 Desligamento (*dropout*)

O desligamento é uma técnica que objetiva remover aleatoriamente neurônios da rede e todas as suas conexões, tanto de camadas visíveis como de camadas escondidas, a fim de prevenir o sobre-ajuste dos pesos da rede e de modificar sua estrutura para que mais tipos de arquiteturas sejam treinadas conjuntamente (GOODFELLOW et al., 2016). Os neurônios são desligados temporariamente com probabilidade p na fase de treinamento e, na fase de teste, nenhum neurônio é desligado, porém seus pesos w são multiplicados por p como se pode observar pela Fig. 13 (SRIVASTAVA et al., 2014).

Segundo Srivastava et al. (2014), por meio do desligamento, uma rede que contém n neurônios pode ser amostrada em 2^n redes reduzidas, as quais compartilham pesos e são dificilmente treinadas mais de uma vez. Na etapa de teste, utiliza-se apenas uma arquitetura de rede e não todas as 2^n possibilidades de redes reduzidas, as quais estão

Figura 13 – Representação do funcionamento da técnica de desligamento durante as fases de treinamento e de teste

Fonte: Adaptada de Srivastava et al. (2014)

implícitas na probabilidade p . A técnica em questão também pode ser utilizada como regularização, uma vez que ela impõe alterações nos mapas de características com a remoção temporária de neurônios, produzindo arquiteturas menos complexas (PONTI; COSTA, 2018). Devido a isso, o desligamento pode ser considerado como um método de *ensemble Bagging* (WARDE-FARLEY et al., 2013). A Fig. 14 (a) e (b) ilustram uma MLP comum com duas camadas escondidas e uma MLP com a aplicação de desligamento respectivamente.

Figura 14 – Resultado da implementação da técnica de desligamento

Fonte: Adaptada de Srivastava et al. (2014)

2.2.1.6 Normalização por lotes (*batch normalization*)

No treinamento de redes neurais profundas, a mudança nos parâmetros de uma camada neural altera a distribuição dos mapas de características de entrada de camadas subsequentes. Isso torna o processo de treinamento mais complexo e, ao mesmo tempo, mais

lento, exigindo cuidado ao iniciar os pesos da rede e taxas de aprendizagem mais baixas. Esse problema é denominado de mudança covariável interna, o qual pode ser solucionado pela normalização de cada *mini-batch* de entrada das camadas neurais anteriores (IOFFE; SZEGEDY, 2015). Redes neurais profundas de classificação no estado-da-arte, como as redes de He et al. (2016) e de Huang et al. (2017), frequentemente utilizam normalização por lotes, já que essa técnica também funciona como um regularizador e que, em determinados casos, dispensa a necessidade de desligamento. Tal método de normalização estabelece valores próximos a 0 e a 1 de ativação média e de desvio padrão respectivamente (PONTI; COSTA, 2018). A Eq. 2.22 representa a transformação linear dos dados normalizados do *mini-batch*, cujos parâmetros γ e β são ajustáveis e influenciam a normalização.

$$BN_{\gamma,\beta}(x_i) = \gamma \left(\frac{x_i \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \right) + \beta \quad (2.22)$$

Outras técnicas de normalização evitam aplicar as operações de média e variância nos pixels dos lotes, como a normalização por camada, a normalização por instância e a normalização por grupo. Esses tipos de normalização podem ser identificados na Fig. 15, em que: H e W representam os eixos dos mapas, C diz respeito ao canal e N caracteriza o batch.

Figura 15 – Métodos de normalização de mapas de características

Fonte: Wu e He (2018)

2.2.1.7 Regularização $\mathcal{L}_2$ (*weight decay*)

Segundo Krogh e Hertz (1992), uma rede tem um baixo poder de generalização quando ela é treinada com uma quantidade de amostras que não corresponde a complexidade de sua estrutura. Assim, a rede sobre-ajusta seus pesos durante o processo de retropropagação, causando a situação de memorização excessiva (*overfitting*). Como a complexidade da rede é dada pelo seu número total de parâmetros treináveis, a técnica de regularização $\mathcal{L}_2$, ou *weight decay*, controla o valor dos pesos da rede, de forma a limitar valores elevados, ajudando a evitar a condição de *overfitting*. A Eq. 2.23 mostra o termo de regularização *weight decay* $\frac{1}{2}\lambda \sum_i w_i^2$ aplicado a uma função de custo inicial $l_0(w)$. Nesta

equação, λ é uma constante de penalização que regula quanto os pesos de valor elevado devem ser penalizados (KROGH; HERTZ, 1992).

$$l(w) = l_0(w) + \frac{1}{2}\lambda \sum_i w_i^2 \quad (2.23)$$

2.2.2 Algoritmos de otimização

Os algoritmos de otimização, como o de Gradiente Descendente (GD), utilizam a definição de gradiente, bem como a regra de diferenciação em cadeia, para minimizar a função de custo e ajustar os parâmetros da rede durante o processo de retropropagação do erro (PONTI; COSTA, 2018). Como o algoritmo de GD computa e acumula o erro após a fase de propagação direta de cada amostra de treinamento (iterativamente), para depois retropropagar o erro quadrático médio, esse tipo de otimização é impraticável para CNNs, visto que estas redes são treinadas com uma quantidade de dados da ordem de 10^3 até 10^6 (PONTI; COSTA, 2018).

Assim, como seria inviável armazenar toda essa quantidade de dados em memória por época, os seguintes métodos de otimização surgiram para resolver tal problema no treinamento de CNNs: Gradiente Descendente Estocástico (SGD) (ROBBINS; MONRO, 1951; KIEFER; WOLFOWITZ et al., 1952), Momentum (QIAN, 1999), Gradiente Adaptativo (AdaGrad) (DUCHI; HAZAN; SINGER, 2011), Taxa de Aprendizagem Adaptativa (AdaDelta) (ZEILER, 2012), RMSProp (TIELEMAN; HINTON, 2012) e Estimação de Momento Adaptativa (ADAM) (KINGMA; BA, 2014).

Ao invés de processar todos os dados, o método de otimização de Gradiente Descendente Estocástico diminui o tempo de execução do treinamento da rede pela utilização, aleatória, de apenas uma amostra ou um *mini-batch* (de tamanho B) do conjunto total de dados para computar o gradiente e retropropagar o erro quadrático médio. A Eq. 2.24 expressa o cálculo de atualização dos pesos da rede com o método SGD. Nesta equação, $\nabla L(W; x_j^B)$ diz respeito ao cálculo do gradiente em função dos pesos atuais W_t e da j -ésima amostra de treinamento do *mini-batch* x_j^B .

$$w_{t+1} = w_t - \eta \sum_{j=1}^B \nabla L(w; x_j^B) \quad (2.24)$$

Ademais, W_{t+1} refere-se ao valor do peso ajustado e η é um parâmetro de proporcionalidade do SGD com valores no intervalo $[0, 1]$, denominado de taxa de aprendizagem, que indica a velocidade com que o algoritmo de treinamento da rede segue na direção de minimização da função de custo (erro). Geralmente, inicia-se η com um valor alto, o qual é reduzido exponencialmente com o tempo (*weight decay*). Ainda que o SGD acelere o processo de treinamento, ele não é capaz de convergir de maneira semelhante ao Gradiente

Descendente, pois a amostragem aleatória do conjunto de treinamento atua como uma fonte de ruído no cálculo do gradiente (GOODFELLOW et al., 2016).

Para acelerar o treinamento de redes profundas, o método de Momentum introduz um termo na atualização dos pesos da rede que favorece seus ajustes para gradientes com a mesma direção e diminui tal ajuste para gradientes divergentes. Isso evita que os parâmetros da rede sejam atualizados na direção oposta da minimização da função de custo, amortecendo assim o comportamento do SGD (PONTI; COSTA, 2018). A equação de atualização dos pesos da rede, considerando o método de Momentum, pode ser descrita pela Eq. 2.25, em que α é a constante de ajuste do fator de Momentum (PONTI; COSTA, 2018).

$$w_{t+1} = w_t + \alpha(w_t - w_{t-1}) + (1 - \alpha)[- \eta \nabla L(w_t)] \quad (2.25)$$

De acordo com Kingma e Ba (2014), o ADAM é um método de otimização estocástico computacionalmente eficiente que demanda apenas gradientes de primeira ordem e baixo uso de memória em relação a outros métodos de otimização, como o Gradiente Descendente. Além disso, o ADAM pode ser aplicado em situações em que há uma grande quantidade de instâncias de treinamento, em objetivos não-estacionários e em problemas que envolvem gradientes esparsos e ruidosos. Isso é alcançado uma vez que o otimizador em questão combina os benefícios dos métodos de otimização AdaGrad (DUCHI; HAZAN; SINGER, 2011) e RMSProp (TIELEMAN; HINTON, 2012).

Em termos práticos, tal método de otimização calcula taxas de aprendizagem adaptativas para cada parâmetro das estimações de momento de primeira ordem m e de segunda ordem g dos gradientes, os quais são descritos pelas Eqs. 2.26 e 2.27 respectivamente (PONTI; COSTA, 2018). Na Eq. 2.26, $\hat{m}$ equivale ao resultado da aplicação do decaimento no momento de primeira ordem. Assim, os parâmetros da rede são ajustados por meio da Eq. 2.28, em que o valor 10^{-8} é normalmente atribuído a constante ϵ (PONTI; COSTA, 2018).

$$m_{t+1} = \alpha_{t+1}m_t + (1 - \alpha_{t+1})\nabla L(w_t) \quad (2.26)$$

$$\hat{m}_{t+1} = \frac{m_{t+1}}{1 - \alpha_{t+1}}$$

$$g_{t+1} = \gamma_{t+1}g_t + (1 - \gamma_{t+1})\nabla L(w_t)^2 \quad (2.27)$$

$$\hat{g}_{t+1} = \frac{g_{t+1}}{1 - \gamma_{t+1}}$$

$$w_{t+1} = w_t - \frac{\eta \hat{m}_{t+1}}{\sqrt{\hat{g}_{t+1} + \epsilon}} \quad (2.28)$$

Existem variações de métodos de otimização estocásticos que partem do algoritmo de atualização de pesos do ADAM, como, por exemplo, o *Nesterov-accelerated Adaptive Moment Estimation* (NADAM) (SUTSKEVER et al., 2013; DOZAT, 2016). Tal método combina o ADAM com o momento de Nesterov para melhorar o processo de atualização dos momentos do ADAM, tornando sua convergência mais rápida.

2.2.3 *Frameworks* de desenvolvimento

Nesta seção, são apresentados os principais *Frameworks* de desenvolvimento que permitem a implementação de algoritmos de Aprendizagem Profunda de uma maneira rápida e eficiente:

- Caffe (JIA et al., 2014)
- CNTK (SEIDE; AGARWAL, 2016)
- Keras (CHOLLET et al., 2015)
- MXNet (CHEN et al., 2015)
- Tensorflow (ABADI et al., 2016)
- Pytorch (PASZKE et al., 2019)
- Theano (BERGSTRA et al., 2010)
- Pylearn2 (GOODFELLOW et al., 2013)

Essas ferramentas apresentam bibliotecas pré-programadas de algoritmos de regularização, otimização e inclusive de arquiteturas de rede de classificação que estão no estado-da-arte. Tais bibliotecas facilitam o desenvolvimento de novas estruturas de rede aplicadas a diferentes tarefas. Ademais, os *Frameworks* procuram fornecer suporte a aceleração em GPU e a processamento distribuído, tendo em vista o tamanho e a complexidade de CNNs.

Este trabalho utiliza o Tensorflow que é uma ferramenta baseada no fluxo de vetores multidimensionais. Ele apresenta suporte a GPU, linguagem de programação intuitiva e uma variedade de bibliotecas pré-desenvolvidas.

2.3 Estimação de profundidade

Os primeiros trabalhos realizados na área de inferência visual de profundidade, no contexto de Visão Computacional, basearam-se no desenvolvimento de algoritmos de correspondência estéreo e de reconstrução. Tais algoritmos conseguem recuperar a profundidade de cenas de maneira determinística por meio de técnicas precisas e robustas

de correspondência entre pontos de imagens retificadas, as quais são captadas por um par de câmeras estéreo calibradas (HARTLEY; ZISSERMAN, 2003; SZELISKI; SCHARSTEIN, 2002). Segundo Trucco e Verri (1998), técnicas de visão estéreo tem o objetivo de estimar distâncias e dados da estrutura 3D de cenas com o auxílio de duas ou mais imagens capturadas de locais diferentes.

As câmeras que geram as projeções das cenas 3D podem ser caracterizadas pelos seus parâmetros extrínsecos, os quais descrevem a orientação e a posição do sistema de coordenadas da câmera (c), em relação a um sistema de coordenadas global (w), e pelos seus parâmetros intrínsecos que são utilizados nas transformações de pontos no referencial da câmera, em milímetros, para o referencial da imagem da câmera (im) em pixels (TRUCCO; VERRI, 1998).

Como se pode observar pelas Eqs. 2.29 e 2.30, respectivamente, os parâmetros extrínsecos da câmera são definidos por uma matriz de rotação em formato homogêneo R_w^c , que representa a orientação da câmera em relação ao sistema de coordenadas global, e por um vetor de translação em formato homogêneo T_w^c , que diz respeito a posição da câmera em relação ao sistema de coordenadas global. Através de tais parâmetros extrínsecos, obtém-se a matriz de transformação homogênea H_w^c da Eq. 2.31 (TRUCCO; VERRI, 1998).

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) & 0 \\ 0 & \sin(\theta_x) & \cos(\theta_x) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.29)$$

$$R_y = \begin{bmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$R_z = \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$R_w^c = R_x \times R_y \times R_z$$

$$T_w^c = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.30)$$

$$H_w^c = T_w^c \times R_w^c \quad (2.31)$$

A matriz de transformação homogênea H_w^c pode ser expressa pela Eq. 2.32, em que $\hat{R}_w^c$ é a matriz quadrada 3×3 composta pelos elementos da matriz de rotação resultante em formato homogêneo R_w^c , da Eq. 2.29, e $\hat{T}_w^c$ consiste na matriz coluna 3×1 formada pelas 3 translações da última coluna da matriz de translação resultante em formato homogêneo T_w^c da Eq. 2.30.

$$H_w^c = \begin{bmatrix} \hat{R}_w^c & \hat{T}_w^c \\ 0^T & 1 \end{bmatrix}, \quad H_w^w = \begin{bmatrix} (\hat{R}_w^c)^T & -(\hat{R}_w^c)^T \times \hat{T}_w^c \\ 0^T & 1 \end{bmatrix}, \quad (2.32)$$

$$\hat{R}_w^c = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}, \quad \hat{T}_w^c = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$$

Por outro lado, os parâmetros intrínsecos dependem da distância focal da câmera f , das dimensões horizontal s_x e vertical s_y de cada pixel da imagem e das coordenadas de projeção do seu eixo óptico (o_x, o_y) . A partir disso, é possível determinar a matriz de parâmetros intrínsecos M_{int} da Eq. 2.33 (TRUCCO; VERRI, 1998).

$$M_{int} = \begin{bmatrix} \frac{-f}{s_x} & 0 & o_x \\ 0 & \frac{-f}{s_y} & o_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.33)$$

Desta forma, para transformar as coordenadas de um ponto no referencial da mundo $P_w = [X, Y, Z, 1]^T$ para o referencial da câmera $p_c = [x, y, z]^T$ (projeção perspectiva), utiliza-se a relação da Eq. 2.34, a qual é formada pelos parâmetros extrínsecos e intrínsecos da câmera. Assim, os pontos na imagem, em pixel, $p_{im} = [x_{im}, y_{im}]^T$ podem ser obtidos por meio da Eq. 2.35 que representa o resultado do mapeamento 3D para 2D, onde a informação de profundidade é perdida (TRUCCO; VERRI, 1998).

$$p_c = M_{int} \times H_w^w \times P_w \quad (2.34)$$

$$p_{im} = \frac{\hat{p}_c}{z}, \quad \hat{p}_c = [x, y]^T \quad (2.35)$$

Tendo isso em vista, aplicações de visão estéreo precisam solucionar problemas de correspondência entre pontos, de retificação e de reconstrução, a qual pode ser compreendida como a tarefa inversa ao mapeamento 3D para 2D. Em problemas de correspondência, o sistema estéreo precisa lidar com oclusões, i. e., com partes das imagens que não devem

ser relacionadas. Em problemas de reconstrução, o sistema deve ser capaz de gerar uma representação 3D da cena por meio do mapa de disparidades e da geometria do sistema estéreo (mapeamento 2D para 3D). O mapa de disparidades descreve a diferença entre todos os pontos das imagens envolvidas no problema (TRUCCO; VERRI, 1998).

Os métodos utilizados para resolver problemas de correspondência podem ser classificados nas categorias *correlation-based* e *feature-based*. Com relação a categoria *correlation-based*, os algoritmos consideram todos os pontos da imagem nos cálculos e o critério de similaridade é definido como a correlação entre janelas de tamanhos fixos nas imagens do par estéreo. Desta forma, a correspondência é obtida pela janela com maior critério de similaridade dentro da região analisada (TRUCCO; VERRI, 1998).

Na categoria *feature-based*, determina-se as correspondências a partir de conjuntos simbólicos e numéricos de características das imagens, os quais podem ser obtidos com os algoritmos: *scale invariant feature transform* (SIFT) (LOWE, 2004), *speed up robust feature* (SURF) (BAY; TUYTELAARS; GOOL, 2006), *robust independent elementary features* (BRIEF) (CALONDER et al., 2010) e *oriented fast and rotated BRIEF* (ORB) (RUBLEE et al., 2011). Logo, as janelas de tamanhos fixos são substituídas por descritores de características que fornecem as informações de similaridade necessárias para encontrar os elementos correspondentes (TRUCCO; VERRI, 1998).

A geometria epipolar do sistema estéreo pode ser representada pela Fig. 16, onde:

- $P = [X, Y, Z]^T$: ponto no referencial do mundo (P_w);
- $P_r = [X_r, Y_r, Z_r]^T$: ponto original no sistema de coordenadas da câmera direita;
- $P_l = [X_l, Y_l, Z_l]^T$: ponto original no sistema de coordenadas da câmera esquerda;
- $p_r = [x_r, y_r, z_r]^T = \frac{f_r}{z_r} P_r$: projeção do ponto na câmera direita;
- $p_l = [x_l, y_l, z_l]^T = \frac{f_l}{z_l} P_l$: projeção do ponto na câmera esquerda;
- T : vetor de translação (*baseline*: $O_l - O_r$ ou T_l^r);
- R : matriz de rotação (R_l^r).

No mapeamento 2D para 3D, os parâmetros extrínsecos do par estéreo, R e T , são responsáveis por associar o referencial das câmeras com centros de projeção em O_l e O_r e com planos de imagem em I_l e I_r respectivamente. A relação entre as coordenadas do ponto P nos referenciais das câmeras direita e esquerda pode ser expressa pela Eq. 2.36. Pela Fig. 16, têm-se que os vetores e_l e e_r correspondem aos epipolos do sistema estéreo,

Figura 16 – Geometria Epipolar

Fonte: Adaptada de Helmke et al. (2004)

os quais formam uma linha que liga os centros de projeção e que interceptam os planos de imagem das câmeras direita e esquerda.

$$P_r = R \times (P_l - T) \quad (2.36)$$

Ademais, as coordenadas do ponto P_l no referencial da câmera esquerda são transformadas nas coordenadas da projeção p_l através do termo f_l que corresponde a distância focal da câmera em questão. Essa relação também é válida para P_r e p_r . A partir da geometria epipolar da Fig. 16, é possível deduzir relações importantes, dentro da tarefa de reconstrução, como as da Eq. 2.37, relacionadas a matriz essencial E , e como as da Eq. 2.38, vinculadas a matriz fundamental F . Na Eq. 2.37, o termo S depende de parâmetros extrínsecos do par estéreo e, na Eq. 2.38, os termos M_r e M_l referem-se as matrizes de parâmetros intrínsecos M_{int} das câmeras direita e esquerda respectivamente (TRUCCO; VERRI, 1998).

$$S = \begin{bmatrix} 0 & -T_z & T_y \\ T_z & 0 & -T_x \\ -T_y & T_x & 0 \end{bmatrix}, \quad (2.37)$$

$$E = R \times S,$$

$$p_r^T \times E \times p_l = 0$$

$$F = M_r^{-T} \times E \times M_l^{-1}, \quad (2.38)$$

$$\bar{p}_r^T \times F \times \bar{p}_l = 0$$

Acrescentando-se a isso, os termos $\bar{p}_r$ e $\bar{p}_l$ correspondem aos pontos p_r e p_l com coordenadas em pixel e a Eq. 2.38 pode ser solucionada para a matriz F pelo método dos oito-pontos correspondentes. O mesmo se aplica para calcular a matriz E , a qual pode ser determinada tanto pela relação 2.37 como pela 2.38. De posse da matriz fundamental, é possível localizar os epípolos da geometria epipolar com o auxílio da Eq. 2.39 (TRUCCO; VERRI, 1998).

$$\bar{p}_r^T \times F \times \bar{e}_l = 0, \quad (2.39)$$

$$F \times \bar{e}_l = 0$$

Para facilitar o processo de correspondência, técnicas de retificação buscam encontrar uma transformação nas imagens esquerda e direita de um sistema estéreo, de forma que seus pares de linhas epipolares fiquem alinhados paralelamente a um dos planos da imagem. Sendo assim, os epípolos ficarão localizados no infinito das linhas epipolares retificadas e os métodos de correspondência serão restritos a encontrar similaridades em apenas uma dimensão (TRUCCO; VERRI, 1998). A Fig. 17 apresenta um processo de retificação de um par de imagens estéreo.

Figura 17 – Retificação de um par estéreo

Fonte: Adaptada de Fusiello, Trucco e Verri (2000)

Após estabelecer os pontos correspondentes e encontrar a geometria epipolar, o problema de reconstrução pode ser solucionado conforme a disponibilidade de dados sobre o par de câmeras estéreo. Quando os parâmetros intrínsecos e extrínsecos do sistema estéreo são conhecidos, aplica-se a técnica de triangulação para reconstruir a cena. No entanto, conhecendo-se apenas os parâmetros intrínsecos, a cena é reconstruída e os parâmetros extrínsecos são inferidos a menos de um fator de escala. Finalmente, se não há informações

sobre os parâmetros intrínsecos e extrínsecos, é possível resolver o problema de reconstrução 3D a menos de uma transformação projetiva global (TRUCCO; VERRI, 1998).

No entanto, existem limitações quanto a aplicações de visão estéreo para estimar profundidade, uma vez que esse método é dependente de uma *baseline* e ele exige algoritmos complexos de correspondência de imagens, calibração eficiente e esforço computacional em cálculos de triangulação. Isso restringe a implementação desse tipo de sistema em robôs móveis de pequeno porte (PELUSO et al., 2019) e em áreas médicas (FORSTER; TOZZI, 2000). Além disso, em locais da cena onde não ocorre a presença de características relevantes, como em regiões de baixa textura e refletoras, o sistema estéreo fica mais suscetível a falhas (MA; KARAMAN, 2018).

Por outro lado, no que diz respeito a sensores que medem profundidade, os LiDARs são proibitivos em termos de custo e realizam medições esparsas e ruidosas de profundidade a longas distâncias, enquanto os sensores Kinect são sensíveis à luz e têm um limite de medição de distância mais baixo quando comparado ao LiDAR. Tendo isso em vista, o uso de câmeras monoculares torna-se vantajoso no contexto da percepção robótica, sabendo-se que elas exigem um espaço menor de instalação, são de baixo custo e apresentam um consumo energético menor em relação a outros sensores (LiDAR e Kinect) (MA; KARAMAN, 2018). Sendo assim, a próxima seção trata das primeiras técnicas desenvolvidas para estimação monocular de profundidade, em contraste aos métodos que abordam visão estéreo.

2.3.1 Estimação monocular de profundidade

O problema de estimação monocular de profundidade é ambíguo e também não bem-posto, uma vez que uma única imagem no plano 2D pode ser gerada a partir de uma quantidade infinita de possibilidades de cenas 3D do mundo real. Os primeiros trabalhos nesse campo basearam-se no uso de características artesanais selecionadas manualmente para inferir relações geométricas na estrutura de uma cena (HOIEM; EFROS; HEBERT, 2005a; HOIEM; EFROS; HEBERT, 2005b; SAXENA et al., 2007). Esses métodos recorrem a pistas monoculares locais, como oclusões, escalas, perspectivas, texturas e locais de artefatos, e a características monoculares globais, como a forma e a configuração da cena (SAXENA; CHUNG; NG, 2006).

Com esse escopo, Hoiem, Efros e Hebert (2005a) introduziram um método automático que infere a estrutura 3D de uma cena utilizando apenas uma imagem. Para isso, por meio de cálculos probabilísticos, tal método gera classes geométricas do tipo “chão”, “vertical” e “céu” com o intuito de rotular os pixels da imagem. Pistas monoculares da cena, como intensidade dos pixels, estruturas geométricas e textura são selecionadas por uma árvore de decisão que auxilia na determinação das classes de cada pixel. A Fig. 18 mostra como ocorre o processo de construção do modelo 3D a partir dos rótulos.

O método proposto por Hoiem, Efros e Hebert (2005a) para modelar estruturas

Figura 18 – Modelo 3D gerado a partir dos rótulos geométricos da imagem

Fonte: Hoiem, Efros e Hebert (2005a)

3D apresenta limitações acerca da sua capacidade de generalização e do detalhamento de seus resultados. Em uma continuação deste trabalho, Hoiem, Efros e Hebert (2007) desenvolveram uma técnica mais elaborada que estima a superfície 3D de cenas *outdoor* de maneira monocular a partir da construção de modelos baseados em aparência. Tais modelos são compostos por classes geométricas que estão vinculadas às orientações dos objetos presentes na imagem. A Fig. 19 exemplifica o funcionamento das classes geométricas de acordo com as cores dos rótulos principais e as marcações dos rótulos secundários.

Figura 19 – *Layout* da superfície da imagem desenvolvido por meio de classes geométricas

Fonte: Hoiem, Efros e Hebert (2007)

Beneficiando-se de modelos gráficos, Saxena, Chung e Ng (2006) introduziram uma forma de estimar profundidade a partir de uma única imagem através do treinamento de um *Markov Random Field* (MRF) que utiliza características monoculares locais e globais da imagem para determinar a profundidade dos pixels de um *patch* e sua relação com os pixels de *patches* vizinhos. A Fig. 20 ilustra como a imagem é dividida em *patches* que contém vetores com características locais, como variações e gradientes de textura e neblina.

No entanto, o método introduzido por Saxena, Chung e Ng (2006) é fundamentalmente dependente de pistas monoculares e sofre com o problema de estimar formas de profundidade bem definidas a partir de objetos finos. Dando seguimento a este trabalho, Saxena, Sun e Ng (2008) propuseram um algoritmo de segmentação de superpixel que divide a imagem de entrada em superfícies planares. Assim, para cada superpixel, um MRF estima a posição e orientação da superfície 3D. Abordando segmentação semântica, o trabalho de Liu, Gould e Koller (2010) visa reconstruir a profundidade de imagens,

Figura 20 – Vetores de características de *patches* na imagem

Fonte: Saxena, Chung e Ng (2006)

também de maneira monocular, utilizando rótulos semânticos que fornecem informações geométricas sobre a profundidade de cada classe. Deste modo, as profundidades dos pixels da imagem são determinadas por meio de características artesanais, as quais são utilizadas no treinamento de um modelo linear que prediz profundidade-log. As predições de tal modelo são, posteriormente, melhoradas por MRFs. Seguindo uma linha semelhante, Ladicky, Shi e Pollefeys (2014) formularam um classificador que estima a probabilidade dos pixels de uma imagem estarem em uma profundidade canônica específica. Tal classificador, ilustrado na Fig. 21, também estima classes semânticas, de forma que ambas as tarefas são beneficiadas.

Em um trabalho paralelo, Liu, Salzmann e He (2014) elaboraram um modelo gráfico supervisionado que recorre ao conceito de *Conditional Random Field* (CRF) para estimar profundidade a partir de uma única imagem. Tal modelo utiliza variáveis contínuas e discretas que descrevem a profundidade de superpixels planares na imagem e as condições de oclusão entre superpixels próximos respectivamente. Com uma abordagem distinta, Karsch, Liu e Kang (2014) desenvolveram um método não paramétrico (Fig. 22) que utiliza funções de transformação de imagem, inferidas pelo algoritmo SIFT Flow (LOWE, 2004), e informações temporais para solucionar problemas de SIDE, até mesmo para objetos em movimento de vídeos.

Assim, nesta seção, foram expostos os principais trabalhos de SIDE que empregam características monoculares artesanais de imagens, como: sombreamento, iluminação, sobreposição, tamanho relativo e absoluto de objetos e elevação, para estimar profundidade. Apesar do auxílio de relações geométricas (HEDAU; HOIEM; FORSYTH, 2010; GUPTA

Figura 21 – Classificador de profundidade e semântica

Fonte: Ladicky, Shi e Pollefeys (2014)

Figura 22 – Técnica de estimação de profundidade que utiliza o algoritmo SIFT (LOWE, 2004) e informações temporais extraídas de vídeos

Fonte: Karsch, Liu e Kang (2014)

et al., 2010), rótulos semânticos (LADICKY; SHI; POLLEFEYS, 2014; LIU; GOULD; KOLLER, 2010) e de modelos gráficos (SAXENA et al., 2007; LIU; SHEN; LIN, 2015), esses trabalhos sofrem com falta de generalização e dependência de pistas monoculares manualmente selecionadas. Suprimindo esses problemas, os métodos de SIDE que abordam aprendizagem são introduzidos no Capítulo 3.

3 TRABALHOS CORRELATOS

Neste capítulo, apresenta-se os trabalhos da literatura de SIDE que abordam técnicas de Aprendizagem Profunda. Particularmente, para o campo em questão, objetiva-se citar e descrever tipos diferentes de topologias e de arquiteturas de CNN que estão no estado-da-arte. Enfatiza-se os métodos com aprendizagem supervisionada, já que este é o foco da presente proposta, porém também são expostos métodos com aprendizagem semi-supervisionada e auto-supervisionada devido as suas importâncias para a área. Acrescentando-se a isso, trabalhos de densificação de profundidade são referenciados, uma vez que esta é uma tarefa correlata a tarefa de SIDE.

3.1 Aprendizagem profunda e estimação monocular de profundidade

Como se pode observar, existe uma quantidade significativa de metodologias na literatura que resolvem problemas de estimação monocular de profundidade (HOIEM; EFROS; HEBERT, 2008; HOIEM; EFROS; HEBERT, 2007; HEDAU; HOIEM; FORSYTH, 2009). No entanto, com o avanço de redes neurais mais complexas e eficientes no contexto de Aprendizagem Profunda, como as redes de Szegedy et al. (2015), He et al. (2016), Huang et al. (2017), Howard et al. (2017), Xie et al. (2017), Chollet (2017), Vaswani et al. (2017), Hu, Shen e Sun (2018), Zoph et al. (2018), Tan e Le (2019), começaram a surgir novos trabalhos que aplicam tais redes, além de modelos gráficos, para extrair mapas de características densos em aplicações de SIDE. As Figs. 23 e 24 ilustram os blocos fundamentais que compõem as redes de He et al. (2016), Huang et al. (2017), respectivamente, as quais são frequentemente utilizadas em arquiteturas de rede de estimação monocular de profundidade.

Figura 23 – Estrutura fundamental da ResNet

Fonte: adaptada de He et al. (2016)

Assumindo mapas de características de entrada x_0 , e, para cada camada neural ℓ , uma função de transformação linear H_ℓ , CNNs comuns, como a de Krizhevsky, Sutskever e

Figura 24 – Estrutura fundamental da DenseNet

Fonte: Huang et al. (2017)

Hinton (2012), conectam a saída da camada ℓ com a entrada da camada $\ell + 1$, gerando a expressão $x_{\ell+1} = H_{\ell+1}(x_\ell)$ que corresponde a saída atual, sendo que a função $H_{\ell+1}$ pode ser composta por operações de convolução, normalização por lotes, *pooling*, entre outras. No entanto, para o caso da ResNet, há uma concatenação entre a saída da camada ℓ com a saída da transformação linear da camada $\ell + 1$, produzindo a Eq. 3.1.

Com o intuito de melhorar o fluxo de informações na rede, o qual é limitado pela operação de concatenação entre os resultados de $H_{\ell+1}(x_\ell)$ e da aplicação de uma função identidade em x_ℓ na ResNet, a DenseNet apresenta conexões entre os mapas de características da camada atual $x_{\ell+1}$ com todos os mapas de características de camadas anteriores $x_0, x_1, \dots, x_\ell$. A partir disso, tem-se a Eq. 3.2 que representa a formação dos mapas de características da camada atual da DenseNet.

$$x_{\ell+1} = H_{\ell+1}(x_\ell) + x_\ell \quad (3.1)$$

$$x_{\ell+1} = H_{\ell+1}([x_0, x_1, \dots, x_\ell]) \quad (3.2)$$

Para a tarefa apresentada, os modelos de CNN típicos têm uma estrutura totalmente convolucional na forma codificação-decodificação. O codificador é responsável por extrair mapas de características da entrada e o decodificador sobre-amostra tais mapas para recuperar imagens de profundidade de alta resolução. No entanto, as redes base de extração de características usam séries de camadas de *pooling* espacial e operações de passo que reduzem as dimensões dos mapas densos, o que prejudica a qualidade das previsões (FU et al., 2018).

Esse efeito pode ser atenuado quando ASPP Denso (YANG et al., 2018a) e mapas de características multi-escala (ROY; TODOROVIC, 2016) são utilizados na arquitetura. No âmbito de Aprendizagem Profunda, o processo de estimação monocular de profundidade é, geralmente, realizado pela minimização de uma função de custo de regressão como o erro quadrático médio (MSE) (FU et al., 2018) e o erro absoluto médio (MAE) (GODARD; AODHA; BROSTOW, 2017). Em relação ao tipo de aprendizado, os modelos de SIDE podem ser divididos em três categorias: supervisionado, semi-supervisionado e auto-supervisionado.

3.1.1 Aprendizado supervisionado

Esse tipo de abordagem de SIDE utiliza mapas de profundidade esparsos e ruidosos de referência, construídos por meio das nuvens de pontos de varreduras de *laser*, para treinar modelos de rede profunda de maneira supervisionada. Nesse escopo, Eigen, Puhrsch e Fergus (2014) realizaram um trabalho fundamental no contexto de estimação monocular de profundidade, em que foi proposta a arquitetura de rede da Fig. 25, a qual é composta por duas etapas bem definidas.

A primeira etapa utiliza camadas convolucionais de início e Camadas Totalmente Conectadas (FC) na parte final. Esta etapa é responsável por estimar a profundidade da cena de um ponto de vista global a partir de uma imagem RGB (*Red, Green, Blue*). A segunda etapa da rede é constituída apenas por camadas convolucionais de escala fina. Nesta etapa, são realizados refinamentos a nível local por meio do mapa produzido na primeira etapa e da imagem RGB. Ademais, o trabalho descrito também introduz uma função de custo de Escala Invariante (SILog) que foca nas relações de profundidade dos pixels da imagem, ao invés de focar na escala geral.

Explorando os benefícios de modelos gráficos, Liu, Shen e Lin (2015) combinaram uma CNN e um CRF contínuo, formando uma arquitetura de campo neural convolucional profundo, para estimar profundidade a partir de uma única imagem. Além desta arquitetura de rede, o trabalho em questão também propôs um novo método de *pooling* de superpixel que aborda o problema de perda de informações na imagem devido a sucessivas operações de subamostragem.

Em um outro trabalho fundamental, Laina et al. (2016) introduziram uma arquitetura de rede FCN composta por blocos de *up-projection* (expostos na Fig. 26) que utilizam operações de deconvolução, juntamente com artifícios de aprendizagem residual (HE et al., 2016), para prever mapas de profundidade de melhor qualidade e de uma maneira mais rápida. Acrescentando-se a isso, no trabalho citado, foi enfatizado os benefícios da utilização a função de custo inversa de Huber (BerHu) (ZWALD; LAMBERT-LACROIX, 2012; GIRSHICK, 2015) para treinar redes de SIDE.

Diferentemente de uma regressão comum, o trabalho Cao, Wu e Shen (2017) utiliza

Figura 25 – Arquitetura constituída de uma CNN de escala grossa global e uma CNN de escala fina local

Fonte: Eigen, Puhrsch e Fergus (2014)

Figura 26 – Blocos de *up-projection*

Fonte: adaptada de Laina et al. (2016)

métodos de classificação para estimar profundidade de maneira monocular por meio da discretização dos dados de referência em intervalos fixos no espaço logarítmico. Neste trabalho, foi treinada uma rede residual totalmente convolucional com o auxílio de uma função de custo logística multinomial que considera as predições da rede que são próximas ao valor do *ground truth* para a atualização dos pesos, diferentemente de outras tarefas que utilizam métodos de classificação comuns como a segmentação semântica (YUAN; CHEN; WANG, 2019).

A função de custo apresentada faz uso da distribuição de probabilidade gerada nas previsões da rede, as quais também servem de entrada para um CRF integrado como pós-processamento. Os resultados levantados nos experimentos de Cao, Wu e Shen (2017) mostram que a estimação de profundidade por classificação supera a regressão convencional, uma vez que esta pode gerar maiores erros nas regiões mais próximas e mais distantes da câmera devido a sua convergência para os valores médios de profundidade.

Contudo, a estimação por classificação é dependente do número de faixas de discretização e a função de custo da rede não é capaz de diferenciar valores de profundidade dentro de uma faixa. A Fig. 27 apresenta a arquitetura de classificação proposta pelos autores.

Figura 27 – Modelo de SIDE baseado em classificação que utiliza CRF

Fonte: Cao, Wu e Shen (2017)

Inspirados por Cao, Wu e Shen (2017), Liao et al. (2017) introduziram uma rede neural denominada de *residual of residual* para estimação monocular de profundidade, a qual recebe uma imagem RGB concatenada com um mapa denso de referência gerado a partir de dados de uma faixa de *laser* 2D. Acrescentando-se a isso, no treinamento de tal rede, é combinada uma função de custo de regressão com uma função de custo de classificação, cuja parte de regressão funciona como um ajuste fino para a parte de classificação caso a predição de profundidade caia na faixa discretizada correta. Além disso, o método de classificação é capaz de fornecer os valores de incertezas da predição da rede, enquanto que o método de regressão é capaz de penalizar os erros preditivos maiores que não são distinguidos pela classificação.

Também abordando problemas de SIDE de maneira inovadora, Fu et al. (2018) formularam um método de regressão ordinal quantizado com base em uma estratégia de discretização de aumento de espaço e em uma função de custo ordinal de treinamento. A estratégia em questão visa discretizar uniformemente o *ground truth* contínuo em intervalos de profundidade no espaço logarítmico, cujos valores são utilizados para computar a função de custo que se beneficia das correlações ordinais das profundidades discretizadas. A arquitetura de rede profunda desenvolvida no trabalho Fu et al. (2018) está ilustrada na Fig. 28, em que é possível observar a presença de uma etapa de extração de características, de um bloco de ASPP (CHEN et al., 2017), de convoluções com *kernel* 1×1 , de um

módulo que processa informações contextuais globais e de uma fase de regressão ordinal de decodificação.

Figura 28 – Método de regressão ordinal para estimar profundidade de maneira monocular

Fonte: Fu et al. (2018)

Em um trabalho relacionado, Xu et al. (2017) propuseram a estrutura de rede da Fig. 29, para prever mapas de profundidade, a qual integra uma etapa principal de CNN com um módulo de fusão composto por CRFs contínuos. O módulo de fusão, com formato de CNN multi-escala, conecta mapas de características sobre-amostrados do decodificador, o que melhora o desempenho do modelo como um todo. Explorando técnicas de Visão Computacional, Yang et al. (2019) utilizaram métodos de estimação de pose da câmera, estimação e propagação de incerteza e inferência bayesiana para prever mapas de profundidade.

Com o auxílio de estratégias de aprendizagem por transferência, Alhashim e Wonka (2018) empregaram o modelo DenseNet-169 (HUANG et al., 2017) em uma nova arquitetura totalmente convolucional, pré-treinada no conjunto de dados ILSVRC (RUSSAKOVSKY et al., 2015), para realizar estimação monocular de profundidade de alta qualidade. Na etapa de treinamento dessa rede, os autores implementaram uma função de custo que computa as diferenças tanto a nível de pixel quanto estruturais entre o *ground truth* e as previsões.

Recentemente, Lee et al. (2019) introduziram uma arquitetura de rede supervisionada em formato codificação-decodificação, cuja estrutura de codificação utiliza, principalmente, a DenseNet como rede base. Acoplado a DenseNet, tem-se um módulo denso de aprendizagem multi-escala de características (ASPP Denso), seguido pela estrutura de decodificação, a qual é composta por blocos de convolução e *upconvolution*, operações de subamostragem e concatenação e por camadas de orientação planar local (LPG) e redução 1×1 . Essas camadas de LPG fornecem orientações geométricas para formar o mapa de profundidade de saída da rede por meio da estimação de quatro parâmetros (*4D plane coefficients*) que são convertidos em pistas monoculares locais de fragmentos de tamanhos específicos do mapa de resolução total. A arquitetura de rede desenvolvida por Lee et al. (2019) está exposta na Fig. 30.

Figura 29 – Arquitetura de rede FCN com um módulo de fusão multi-escala

Fonte: Xu et al. (2017)

Figura 30 – CNN composta por camadas de orientação planar local

Fonte: Lee et al. (2019)

3.1.1.1 Temporalidade

Beneficiando-se de características temporais de sequências de imagem para a tarefa de SIDE, Mancini et al. (2017) mostraram como a adição de camadas *long short-term memory* (LSTM) (HOCHREITER; SCHMIDHUBER, 1997) totalmente conectadas (FC-LSTM) em uma arquitetura de CNN pode melhorar a precisão dos resultados, principalmente devido às informações que são armazenadas nas células de memória entre as etapas de codificação e decodificação. Tal trabalho também mostra como a utilização de conjuntos de dados sintéticos de treinamento, além de informações temporais, torna o modelo de CNN mais robusto a aplicações reais como de veículos micro-aéreos.

No entanto, a fim de utilizar sequências de imagem como entrada em uma FC-LSTM, estas sequências precisam ser transformadas em vetores 1D, o que prejudica a compreensão da rede sobre as relações espaciais contidas na cena. A Conv-LSTM é um tipo de camada neural que possibilita a extração de características espaciais e temporais de imagens através da combinação de habilidades de camadas convolucionais e LSTM. Tal camada neural foi introduzida por Xingjian et al. (2015) para o problema espaço-temporal de previsão da quantidade de precipitação que ocorre em um local específico ao longo de um período limitado.

Ademais, a camada Conv-LSTM foi empregada na arquitetura de rede supervisionada multi-escala, em formato U-net (RONNEBERGER; FISCHER; BROX, 2015), proposta no trabalho de Kumar, Bhandarkar e Prasad (2018a) com o intuito de realizar estimativa monocular de profundidade. Em um trabalho correlato, Wang, Pizer e Frahm (2019) utilizaram camadas Conv-LSTM em uma CNN que estima mapas de profundidade densos e odometria visual. Tal rede pode ser treinada tanto de maneira supervisionada como auto-supervisionada.

3.1.1.2 Atenção

Mecanismos baseados em atenção também têm sido enfatizados por trabalhos no estado-da-arte em contextos de Aprendizagem Profunda, já que essas técnicas tornam as arquiteturas de CNN mais eficientes tanto para tarefas de SIDE como para tarefas de segmentação semântica e de densificação de profundidade (JIAO et al., 2018). Modelos baseados em atenção são adaptativos e permitem que apenas características que apresentam informações significativas sejam focalizadas durante o treinamento, processando assim as características menos relevantes a resoluções mais baixas (HU; SHEN; SUN, 2018).

Seguindo este escopo, Xu et al. (2018) propuseram uma CNN multi-escala de estimativa monocular de profundidade composta por um módulo de CRF guiado por atenção, o qual é responsável por limitar as informações dos mapas gerados na etapa de extração de características da rede. Paralelamente, Hao et al. (2018) alcançaram resultados expressivos com uma arquitetura de rede constituída de um bloco denso de extração de características, o qual utiliza uma ResNet e convoluções dilatadas, e de um bloco gerador de mapas de profundidade que utiliza técnicas baseadas em atenção para estimar profundidade a partir dos mapas de características, enfatizando as informações mais importantes para a tarefa em questão. Chen, Zhao e Hu (2019) introduziram uma CNN de agregação de contexto baseada em atenção, a qual, na etapa de decodificação, é composta por camadas de inferência ordinal de saída e por um bloco de extração e agregação de informações contextuais a nível de pixel e a nível global.

Recentemente, trabalhos que exploram conceitos de atenção em suas arquiteturas de rede têm empregado os mecanismos de atenção de Transformer (VASWANI et al.,

2017; DOSOVITSKIY et al., 2020), os quais não dependem de camadas convolucionais, somente de estruturas de atenção multi-cabeça que realizam operações de soma ponderada entre os vetores de *values* e o resultado da função de compatibilidade aplicada ao produto dos vetores de *query* pelas *keys*. A Fig. 31 ilustra a composição de uma estrutura de atenção multi-cabeça, em que V, K e Q representam os vetores *values*, *keys* e *queries* respectivamente.

Figura 31 – Arquitetura de rede Transformer com detalhamento da atenção ao produto ponto em escala

Fonte: Vaswani et al. (2017)

Como exemplo do sucesso da aplicação de redes supervisionadas baseadas em Transformer, os autores Bhat, Alhashim e Wonka (2020) utilizaram uma rede convolucional com camadas neurais Transformer Visual (ViT) (DOSOVITSKIY et al., 2020) no *decoder* a fim de estimar, de maneira adaptativa, intervalos de profundidade, cujos centros são submetidos a uma combinação linear para determinar o mapa de profundidade de saída. Ranftl, Bochkovskiy e Koltun (2021) também se beneficiaram de ViT, porém eles o utilizam como *backbone* (*encoder*) para um *decoder* convolucional. Uma vez que o ViT recebe *patches* da imagem como entrada, ele computa representações com um campo receptivo global e dispensa operações de subamostragem.

No entanto, as redes ViT apresentam limitações acerca da sua capacidade de generalização, já que elas não conseguem explorar correlações espaciais como as CNNs. Sendo assim, as arquiteturas Transformer precisam ser treinadas em uma quantidade maior de dados para suprirem a falta de equivariância de translação e localidade. Além disso, como os Vision Transformers processam sequências de vetores em seu *backbone*, eles dependem da aprendizagem de MLPs, cuja quantidade de neurônios pode elevar significativamente o número de parâmetros treináveis de uma rede de SIDE.

3.1.1.3 Profundidade e semântica

Segmentação semântica e estimação monocular de profundidade são tarefas importantes e correlatas dentro do campo da Visão Computacional, uma vez que, a partir delas, é possível extrair informações relevantes de imagens como a geometria, a escala e o local de artefatos, a estrutura e disposição global da cena e as distâncias em que se encontram os objetos (JIAO et al., 2018). Acrescentando-se a isso, as tarefas citadas se beneficiam de uma maneira mútua, pois a compreensão das profundidades da cena auxilia a segmentação semântica durante o processo de definição das categorias, principalmente para diferentes artefatos com estruturas semelhantes e profundidades distintas (WANG et al., 2015). Além disso, rótulos semânticos fornecem informações geométricas e de perspectiva que ajudam na estimação das profundidades de cada objeto da cena. Um exemplo de tais informações geométricas pode ser observado na diferenciação entre as superfícies horizontal e vertical para classes do tipo “chão” e “prédio” respectivamente (JIAO et al., 2018).

Antes do advento de redes neurais profundas, alguns trabalhos já utilizavam métodos abordando segmentação semântica para estimar profundidade com o auxílio de técnicas clássicas de análise e compreensão de projeções 2D (HOIEM; EFROS; HEBERT, 2011; LIU; GOULD; KOLLER, 2010; GOULD; FULTON; KOLLER, 2009). Ademais, outras metodologias visavam determinar classes semânticas mais precisas por meio de imagens RGBD (SILBERMAN et al., 2012; REN; BO; FOX, 2012; GUPTA et al., 2014). Explorando as capacidades de redes neurais convolucionais, trabalhos recentes utilizam redes supervisionadas por rótulos semânticos e por mapas de profundidade a fim de se beneficiar da correspondência de ambas as tarefas e gerar resultados de melhor qualidade.

Nesse contexto, Eigen e Fergus (2015) introduziram uma arquitetura de rede multi-escala totalmente convolucional capaz de gerar predições de profundidade, normais de superfície e semântica a partir de uma única imagem RGB de entrada. Com uma abordagem próxima, Mousavian, Pirsaviash e Košecká (2016) elaboraram um método supervisionado que consiste no treinamento conjunto de uma CNN e de um CRF para estimar profundidade e semântica. Tal treinamento é realizado com o auxílio de uma função de custo que aborda as duas tarefas.

Seguindo um escopo semelhante, Wang et al. (2015) propuseram uma rede que realiza predições de mapas de profundidade e de classes semânticas, simultaneamente, por meio de duas CNNs e de um modelo gráfico denominado *Hierarchical Conditional Random Field*. Partindo dessa mesma diretriz, Jiao et al. (2018) desenvolveram uma rede de estimação monocular de profundidade e de segmentação semântica, cujo treinamento supervisionado é baseado na minimização de uma função de custo orientada a atenção, a qual ajuda a melhorar o desempenho da rede em regiões mais profundas da imagem de entrada.

3.1.1.4 Profundidade e normais de superfície

No campo da Visão Computacional, estudos sobre a compreensão 3D de cenas buscaram, primeiramente, informações de luz, sombra e cor nas imagens (ZHANG et al., 1999; MALLICK et al., 2005), características geométricas, como contornos (ROBERTS, 1963; SUGIHARA, 1986; WALTZ, 1975) e formas poliédricas (ROBERTS, 1963), e formas volumétricas como agrupamentos de cilindros (BINFORD, 1971) e geons (BIEDERMAN, 1987). Devido a dificuldade de se obter algumas dessas características primitivas e devido as suas limitações em aplicações mais complexas, trabalhos posteriores focaram em explorar outras formas de pistas geométricas como segmentos e *vanishing points* (HOIEM; EFROS; HEBERT, 2007; HOIEM; EFROS; HEBERT, 2005b; LEE; HEBERT; KANADE, 2009), bordas (MALLYA; LAZEBNIK, 2015), representações de cubo, de blocos e de caixa 3D (XIAO; RUSSELL; TORRALBA, 2012; SCHWING; URTASUN, 2012; HOIEM; EFROS; HEBERT, 2007; GUPTA; EFROS; HEBERT, 2010), topologias 3D de objetos (XIANG; SAVARESE, 2012; CHOI et al., 2013; ZHAO; ZHU, 2013) e super-pixels (SAXENA; SUN; NG, 2007). No entanto, alguns desses métodos dependem de relações volumétricas (SCHWING et al., 2013), de segmentação (HOIEM; EFROS; HEBERT, 2005b) e de restrições estruturais como a de *Manhattan World* (COUGHLAN; YUILLE, 2001). Além disso, essas técnicas clássicas utilizam características primitivas artesanais e elas geram layouts 3D esparsos e grosseiros.

Com o surgimento de *datasets* construídos por meio de sensores RGBD, como o Kinect e o FARO Focus, e com o intuito de gerar representações geométricas mais detalhadas, métodos subsequentes buscaram estimar o layout 2,5D de cenas projetadas em imagens 2D por meio de estimações de profundidade e de normais de superfície, as quais são altamente correlacionadas. As normais de superfície da imagem são caracterizadas por vetores perpendiculares a planos tangentes locais e, devido a isso, elas fornecem importantes pistas geométricas da estrutura 3D da cena.

Com o auxílio dos mapas de profundidade gerados pelos sensores RGBD, técnicas baseadas apenas em geometria focaram na obtenção dos vetores normais as superfícies que compõem a cena por meio do ajuste de superfícies planares ou curvas a uma vizinhança local da nuvem de pontos (KLASING et al., 2009). Com a determinação da vizinhança de pontos, através de técnicas como *k nearest neighbor* (KNN) ou *Delaunay tessellation* (DT), esses tipos de técnica conseguem estimar as normais por meio das seguintes abordagens, por exemplo: mínimos quadrados (KLASING; WOLLHERR; BUSS, 2009; JORDAN; MORDOHAI, 2014), relações entre vetores (produtos vetoriais) (KLASING et al., 2009), derivadas parciais das superfícies esféricas tangenciais (BADINO et al., 2011) e o gradiente do inverso da profundidade ou do mapa de disparidades (FAN et al., 2021).

Ainda nesse contexto, Silberman et al. (2012) introduziram um modelo de geração de relações de suporte (como normais de superfície, planos 3D e segmentação) que recebe

imagens RGB alinhadas com seus respectivos mapas de profundidade, e Ladický, Zeisl e Pollefeys (2014) desenvolveram um método de regressão de coeficientes, com aprendizagem discriminativa, o qual emprega técnicas de segmentação para gerar mapas de características. Ademais, outros trabalhos visavam utilizar características primitivas com informações geométricas relevantes e visualmente discriminativas (FOUHEY; GUPTA; HEBERT, 2013) e bordas côncavas e convexas da imagem RGB de entrada (FOUHEY; GUPTA; HEBERT, 2014) para estimar normais de superfície de cenas.

Beneficiando-se do advento de métodos de Aprendizagem Profunda, novos trabalhos passaram a utilizar CNNs para estimar o layout 2,5D de cenas de maneira monocular. Neste cenário, Gupta et al. (2014) desenvolveram uma técnica baseada em aprendizagem que gera máscaras de segmentação e que também detecta contornos e objetos na imagem por meio dos mapas de profundidade dos pares RGBD de entrada, os quais são representados no formato {disparidade, altura, angulação}. No âmbito de estimação de normais de superfície, Wang, Fouhey e Gupta (2015) propuseram um CNN multi-escala que combina informações do layout global da imagem RGB com características locais, como as bordas dos artefatos que compõem a cena, para prever normais de superfície a nível de pixel. Por outro lado, em um âmbito multitarefa, Eigen e Fergus (2015) construíram um modelo de rede profunda multi-escala que se adequa a 3 tipos de tarefas correlacionadas (estimação de profundidade, segmentação semântica e estimação de normais de superfície) através de modificações estruturais em partes das escalas que compõem a rede. Com uma abordagem paralela, Wang et al. (2016) introduziram um *framework* que estima profundidade e normais de superfície com o auxílio de uma *fourstream* CNN, acoplada a um CRF denso, a qual retorna superfícies e bordas planares, assim como profundidade e normais de superfície parciais. Posteriormente, no trabalho de Bansal, Russell e Gupta (2016), foi apresentada a CNN *skip-network* capaz de prever normais de superfície, assim como uma *two-stream* CNN que recebe imagens RGB, juntamente com as saídas da *skip-network*, para gerar dados de modelos de CAD.

Inspirados pelo trabalho de Fouhey, Gupta e Hebert (2013), Qi et al. (2018) elaboraram uma rede com duas ramificações denominadas *normal-to-depth* e *depth-to-normal*. Essa primeira ramificação utiliza um *least square module* que retorna normais de superfície ruidosas, as quais são suavizadas por uma CNN. A ramificação *depth-to-normal* refina mapas de profundidade com o auxílio de operações de *kernel regression*. Com base na taxa de afinidade entre pares de pixels de imagens produzidas em tarefas correlacionadas, Zhang et al. (2019) propuseram uma CNN multitarefa denominada Pattern-Affinitive Propagation (PAP) que explora padrões de similaridade entre pixels correspondentes, por meio de uma matriz de afinidade, a fim de gerar mapas de profundidade, de semântica e de normais de superfície simultaneamente.

Ao invés de empregar estruturas de decodificação multi-escala, Yin et al. (2019)

desenvolveram uma CNN que prediz profundidade de maneira monocular e se beneficia de características geométricas 3D da cena por meio de uma função de custo de treinamento que computa informações do espaço tridimensional no formato de *virtual normals* (VN). De acordo com Yin et al. (2019), *Virtual normals* são vetores perpendiculares aos planos formados pela nuvem de pontos reconstruída diretamente do mapa de profundidade estimado. Sob outra perspectiva, Ramamonjisoa e Lepetit (2019) introduziram a rede SharpNet, que prediz profundidade, normais de superfície e *occluding contours*, além de uma função de custo que força a consistência entre profundidade e *occluding contours* e entre profundidade e normais de superfície. Ramamonjisoa e Lepetit (2019) também montaram um *dataset* para avaliação de *occluding contours*, denominado NYUv2-OC dataset, o qual apresenta 100 imagens de teste retiradas do *dataset* NYU Depth V2.

Recentemente, no contexto amplo de SIDE, é possível observar que trabalhos baseados em aprendizado supervisionado têm incorporado questões vinculadas a: redes menos complexas para predições em tempo real em sistemas embarcados (WOFK et al., 2019), reconstrução 3D (TATENO et al., 2017), aplicações médicas (SHANKARANARAYANA et al., 2019; MAHMOOD; DURR, 2018), redes multi-tarefa (PIASCO et al., 2019; YIN et al., 2017), módulos em formato de pirâmide (CHEN; CHEN; ZHA, 2019), incertezas (ZHANG et al., 2021), redes Transformer (YANG et al., 2021), densificação de profundidade (WANG; AN, 2020) e normais de superfície (HANE; LADICKY; POLLEFEYS, 2015; WANG et al., 2020; ZENG et al., 2019; LENSSEN; OSENDORFER; MASCI, 2020; PAPADAKIS; WILLIS, 2017; WANG et al., 2019). Treinamentos adversariais também têm sido empregados para a tarefa em questão em aplicações de veículos autônomos para ambientes internos e externos (CHEN et al., 2018).

3.1.2 Aprendizado semi-supervisionado

Após o surgimento dos métodos de estimação de profundidade a partir de uma única imagem baseados em aprendizado supervisionado, novos trabalhos tratando sobre estratégias de aprendizagem semi-supervisionada e auto-supervisionada começaram a ser desenvolvidos com a premissa de que os métodos supervisionados necessitam de uma grande quantidade de dados de *ground truth* de profundidade, os quais demandam tempo e esforços significativos para serem produzidos, tornando a aplicação de redes supervisionadas um grande desafio.

Com escopo semi-supervisionado, Kuznietsov, Stuckler e Leibe (2017) propuseram um método de aprendizado que explora pistas supervisionadas e auto-supervisionadas através da utilização de varreduras de *laser* esparsas e ruidosas como *ground truth*, bem como do alinhamento entre as imagens do par estéreo. Inspirados por isso, Ramirez et al. (2018) elaboraram uma rede semi-supervisionada de Aprendizagem Profunda que realiza segmentação semântica e estimação monocular de profundidade. Nesta rede, a parte de

estimação de profundidade é auto-supervisionada e usa a função de custo que contém uma etapa de verificação de consistência esquerda-direita introduzida por Godard, Aodha e Brostow (2017). A parte de segmentação semântica é supervisionada por rótulos presentes em um *ground truth* anotado com semântica a nível de pixel.

Também se beneficiando de treinamentos semi-supervisionados, Amiri, Loo e Zhang (2019) criaram um *framework* que utiliza tanto as imagens esquerda e direita do par estéreo como os mapas de profundidade do *ground truth* ruidoso anotado para estimar profundidade. Da mesma forma que Ramirez et al. (2018), a função de custo semi-supervisionada proposta por Amiri, Loo e Zhang (2019) inclui o termo de consistência esquerda-direita de Godard, Aodha e Brostow (2017).

Explorando recursos de odometria visual estéreo, o *pipeline* proposto por Yang et al. (2018b) usa uma estratégia de treinamento auto-supervisionada, baseada na fotoconsistência de imagens estéreo, para produzir mapas de profundidade, os quais são refinados em um procedimento de treinamento supervisionado, cujos rótulos são adquiridos por *Stereo Direct Sparse Odometry* (WANG; SCHWORER; CREMERS, 2017). Inspirados por isso, Andraghetti et al. (2019) empregam pistas geométricas, obtidas a partir de um algoritmo de odometria visual, para aumentar a precisão de sua abordagem estéreo auto-supervisionada.

3.1.3 Aprendizado auto-supervisionado

O surgimento de métodos auto-supervisionados mostrou que é possível treinar modelos de SIDE utilizando apenas pares de imagens estéreo sincronizados (GARG et al., 2016; GODARD; AODHA; BROSTOW, 2017; PILLAI; AMBRUŞ; GAIDON, 2019) ou vídeo monocular (ZHOU et al., 2017).

3.1.3.1 Treinamento com imagens estéreo

Seguindo a primeira linha, Garg et al. (2016) propuseram um modelo auto-supervisionado para prever profundidade por meio do treinamento de uma CNN que recebe pares de imagens estéreo. A partir do par estéreo, essa CNN é capaz de aprender as transformações não lineares necessárias para recuperar o mapa de profundidade com o auxílio de uma função de custo equivalente à diferença fotométrica entre as imagens das duas câmeras.

Inspirados por isso, Godard, Aodha e Brostow (2017) formularam uma rede convolucional profunda, denominada de Monodepth, que recebe apenas a imagem esquerda do par estéreo como entrada (durante as fases de teste) e prediz mapas de profundidade. Durante o treinamento, a rede aprende a inferir os mapas de disparidade de ambas as imagens das câmeras ao mesmo tempo, as quais são forçadas a serem consistentes pela função de custo de treinamento. Tal função de custo é computada considerando-se três componentes

principais: correspondência de aparência, suavização de disparidade e consistência de disparidade esquerda-direita.

Posteriormente, Godard et al. (2019) aperfeiçoaram a rede Monodepth, criando a rede Monodepth2, através de uma função de custo capaz de reduzir o número de artefatos, de lidar com pixels oclusos e de desconsiderar pixels que não são alterados em uma sequência de imagens.

Inspirados nos trabalhos de Garg et al. (2016), Godard, Aodha e Brostow (2017), Li et al. (2018a) introduziram uma rede que estima profundidade e pose com 6 graus de liberdade denominada de UnDeepVO. A função de custo utilizada no treinamento de tal rede aborda aspectos espaciais e temporais do par de imagens estéreo de entrada.

Também em uma configuração de correspondência estéreo, Luo et al. (2018b) apresentaram o primeiro *framework* de estimação monocular de profundidade que sintetiza a vista direita do par estéreo a partir da imagem esquerda original, as quais são processadas por uma CNN de correspondência estéreo capaz de produzir mapas de disparidade em sua saída. Motivados por isso, Tosi et al. (2019) desenvolveram a arquitetura *end-to-end* monoResMatch que também gera uma vista direita sintética das imagens estéreo. Além do treinamento estéreo auto-supervisionado, os autores empregaram o algoritmo de Correspondência Semi-Global (HIRSCHMULLER, 2005) que retorna o *proxy ground truth* aplicado no treinamento supervisionado da monoResMatch.

Em uma extensão do trabalho de Ummenhofer et al. (2017), cuja CNN DeMoN é supervisionada, Babu et al. (2018) desenvolveram a rede UnDEMoN que também prediz mapas de profundidade e pose da câmera de forma monocular, porém com um melhor comportamento em relação a rede DeMoN devido a utilização de uma função de custo fundamentada na penalidade de Charbonnier (CHARBONNIER et al., 1994), combinada a uma função de custo de reconstrução temporal. Somando-se a isso, o desempenho promissor da rede UnDEMoN dá-se pela utilização de uma rede específica de estimação de profundidade que emprega informações de disparidade, como a rede de Godard, Aodha e Brostow (2017).

Com uma abordagem distinta, Poggi, Tosi e Mattoccia (2018) utilizaram um método trinocular de treinamento para a rede 3Net de SIDE baseada em disparidade. Por meio desse treinamento, os problemas em regiões próximas a oclusões e nas bordas da imagem, normalmente gerados em treinamentos binoculares, são amenizados. Posteriormente, Tosi et al. (2019) propuseram a monoResMatch que é uma arquitetura de rede profunda que estima profundidade por meio da correspondência estéreo entre os mapas de características produzidos a partir da imagem esquerda e da imagem direita virtual do par estéreo. Ademais, outros métodos binoculares da literatura contam com dados de treinamento sintéticos (GUO et al., 2018) e destilação de conhecimento adquirido de um *framework* cíclico (PILZER et al., 2018).

Outros métodos binoculares recorrem a treinamentos adversariais generativos (KUMAR; BHANDARKAR; PRASAD, 2018b; ALEOTTI et al., 2018; GUPTA; MITRA, 2019), manejo de incertezas (CHEN et al., 2020; POGGI et al., 2020) e manuseio de oclusão (GONZALEZBELLO; KIM, 2020). Arquiteturas de rede que são mais leves e rápidas para aplicações em tempo real em sistemas embarcados (POGGI et al., 2018; PELUSO et al., 2019; ELKERDAWY; ZHANG; RAY, 2019) também são aplicadas na abordagem de correspondência estéreo auto-supervisionada de SIDE.

3.1.3.2 Treinamento com vídeo monocular

Seguindo a linha de pesquisa que utiliza vídeo monocular, Zhou et al. (2017) desenvolveram um modelo auto-supervisionado que realiza estimação monocular de profundidade e que também estima a pose da câmera através de duas CNNs específicas que são treinadas conjuntamente com o auxílio de uma função de custo de reconstrução fotométrica. A rede de estimação de pose recebe três imagens como entrada, sendo que uma delas representa a cena principal e as outras são cenas próximas. Em contrapartida, a rede que prediz profundidade recebe apenas uma imagem como entrada.

Em seguida, Yang et al. (2018) elaboraram um método baseado em CNN que estima profundidade a partir de sequências de imagem capturadas por uma câmera em movimento e que também infere representações de normais de superfície por meio de camadas neurais da CNN que processam a consistência entre profundidade e superfície-normal com reconhecimento de borda. Em um trabalho posterior, Yang et al. (2018) introduziram a estrutura de rede profunda LEGO que estima mapas de profundidade, superfícies normais e representações de borda de maneira auto-supervisionada. Nesta rede, as três predições são constantemente estimuladas a serem consistentes por meio de um método de regularização denominado de *3D as-smooth-as-possible*.

Paralelamente, Yin e Shi (2018) propuseram a Geonet que é uma arquitetura de rede auto-supervisionada que prediz profundidade, fluxo óptico e o movimento da câmera simultaneamente. A Geonet é composta por uma etapa de estimação da geometria das estruturas rígidas da cena e por uma etapa de localização de artefatos dinâmicos. Durante o treinamento, os pesos de tal rede são atualizados com o auxílio de uma função de custo de consistência geométrica adaptativa, a qual é robusta a oclusões e a *outliers*. Seguindo o mesmo princípio, Mahjourian, Wicke e Angelova (2018) introduziram um método auto-supervisionado que estima profundidade e odometria visual da câmera a partir de imagens adjacentes. Por meio dos vídeos monoculares, o método consegue estimar as nuvens de pontos 3D das cenas, as quais são correlacionadas entre si e ao movimento da câmera através de uma função de custo geométrica 3D com máscaras de validação.

Com uma proposta multitarefa, Zou, Luo e Huang (2018) desenvolveram uma função de custo de consistência entre tarefas capaz de explorar as relações geométricas

entre as predições de fluxo óptico e de profundidade de duas redes distintas (porém treinadas conjuntamente) e auto-supervisionadas por vídeos não rotulados. Mostrando resultados promissores de estimação de profundidade e de odometria visual, a técnica auto-supervisionada por múltiplas imagens de Casser et al. (2019a) pode modelar estruturas dinâmicas presentes na cena e se adaptar a domínios de dados diferentes, como na transição de ambientes externos para internos, por meio de um refinamento *online*.

Com base na estrutura de CNN proposta por Zhou et al. (2017), a abordagem auto-supervisionada de Wang et al. (2018) considera operações de normalização de profundidade e uma versão diferenciável de um algoritmo de odometria visual direta, enquanto a abordagem de Klodt e Vedaldi (2018) depende do uso de uma estratégia de aprendizagem probabilística e dados de supervisão extra de algoritmos de SfM.

Outras estratégias de SIDE, auto-supervisionadas por vídeos monoculares, utilizam semântica, fluxo e informações de movimento da câmera (TEED; DENG, 2018; CASSER et al., 2019b; RANJAN et al., 2019; CHEN; SCHMID; SMINCHISESCU, 2019; GUIZILINI et al., 2020), bem como outros tipos de restrições geométricas (CHEN; SCHMID; SMINCHISESCU, 2019; GUIZILINI et al., 2019; SHARMA; VENTURA, 2019) e CNNs mais leves e rápidas (LIU et al., 2020).

A partir dos métodos auto-supervisionados descritos, é possível notar que a vertente que utiliza vídeo monocular é desafiadora, posto que o modelo precisa estimar o movimento da câmera (transformações da câmera) através das imagens sequenciais de entrada, além da profundidade da cena. Os modelos da outra vertente não precisam estimar a pose da câmera, uma vez que eles exploram a correspondência de ambas as imagens do par estéreo. No entanto, como desvantagem, podem ocorrer problemas nas estimações de profundidade de áreas próximas a oclusões (GODARD et al., 2019).

3.2 Aprendizagem profunda e densificação de profundidade

Essa tarefa é distinta da tarefa de SIDE, pois ela visa estimar mapas densos de profundidade a partir de medições esparsas de nuvens de pontos, obtidas por meio de sensores de luz-estruturada, sensores LIDAR ou algoritmos de SLAM/SfM (CHENG et al., 2019).

Além de preencher dados em espaços de canais de profundidade sem informação (*depth inpainting*) (HERRERA et al., 2013; KULKARNI; RAJAGOPALAN, 2013; CHEN et al., 2014; ZHANG; YU; WANG, 2018), a tarefa de densificação de profundidade também se concentra em *depth denoising* (ZHANG; WU, 2016; SHEN; CHEUNG, 2013; FU et al., 2012; HU et al., 2013), para restaurar imagens de alta qualidade de sua condição ruidosa, e *depth super-resolution* (FREEMAN; JONES; PASZTOR, 2002; LU et al., 2011; YU et al., 2013; LU; FORSYTH, 2015), para refinar a resolução de mapas de profundidade.

Inicialmente, os trabalhos de densificação de profundidade dependiam de mapas de características artesanais, como *wavelets* e *contourlets*, (LIU; CHAN; NGUYEN, 2015) para reconstruir imagens de profundidade/disparidade (QIU et al., 2019). Nesse contexto, Ku, Harakeh e Waslander (2018) introduziram um algoritmo de densificação de profundidade simples e rápido, baseado em técnicas clássicas de processamento de imagens, como inversão, dilatação, fechamento e preenchimento de orifícios, que podem ser executadas em CPU.

No entanto, essas abordagens artesanais sofrem com a falta de generalização, exigindo ajustes de parâmetros específicos para diferentes tipos de imagens. Com o surgimento das DCNNs, as metodologias baseadas em aprendizado superaram o desempenho dos algoritmos clássicos quando se trata de densificação de profundidade. As abordagens recentes podem ser divididas em duas categorias principais vinculadas à presença ou não de uma imagem RGB para guiar o mapa esparsos de profundidade até a geração da predição.

3.2.1 Densificação de profundidade a partir de amostras esparsas

Nesta categoria, os métodos de densificação de profundidade utilizam apenas mapas esparsos como entrada para predizer mapas densos. Com um trabalho fundamental para a área, Uhrig et al. (2017) propuseram uma rede profunda invariante a esparsidade capaz de sobre-amostrar medidas esparsas de profundidade de varreduras de *laser*. Para alcançar essa invariância à esparsidade da entrada, tal arquitetura é composta por camadas de convolução esparsa que empregam *kernels* convolucionais ponderados na entrada normalizada de acordo com uma máscara de observação binária. As informações dos pixels válidos são transportadas pela rede com a ajuda de camadas de *max pooling*.

Em uma abordagem semelhante, Eldesokey, Felsberg e Khan (2019) propuseram uma CNN multi-escala de densificação de profundidade com convoluções normalizadas que propagam máscaras de confiança para a camada seguinte. No mesmo contexto, Chodosh, Wang e Lucey (2018) desenvolveram um algoritmo de *dictionary learning* para sobre-amostrar os pontos esparsos de LiDAR. Esse método é baseado na Rede Neural de Direção Alternada, introduzida por Murdock, Chang e Lucey (2018), que é capaz de aprender os *dictionaries* e os hiperparâmetros das entradas esparsas.

Os autores Huang et al. (2019) implementaram a rede HMS-Net que lida com mapas esparsos de profundidade por meio de novas operações invariantes a esparsidade, como a adição, a concatenação e a sobre-amostragem bilinear. Recentemente, as convoluções invariantes de esparsidade (UHRIG et al., 2017) estão sendo revisitadas e complementadas por novas operações *mask aware* para lidar com a tarefa de densificação de profundidade guiada por imagens RGB (YAN; LIU; BELYAEV, 2020). Outros poucos trabalhos estão focados em fechar a lacuna entre os métodos guiados e não guiados (LU et al., 2020).

3.2.2 Densificação de profundidade guiada

Os métodos guiados por imagem utilizam informações de guia incrementais para melhorar os resultados da densificação de profundidade. Essas informações de guia, geralmente, são representadas por imagens RGB que fornecem pistas adicionais da cena, como cantos, arestas e semânticas a nível de pixel (TANG et al., 2019).

Seguindo esse escopo, Schneider et al. (2016) utilizaram limites de objetos e pistas de semântica para estimar mapas de profundidade de alta resolução de maneira guiada por imagem. Em uma abordagem baseada em aprendizado, Ma e Karaman (2018) desenvolveram uma DCNN, na qual a codificação é formada pela ResNet-50 (HE et al., 2016) e a decodificação é formada pelos blocos de *up-projection* introduzidos por Laina et al. (2016). Este modelo de regressão utiliza imagens RGB e dados esparsos de profundidade como entrada e produz mapas de profundidade como saída.

Em um trabalho posterior, Ma, Cavalheiro e Karaman (2019) alcançaram resultados relevantes para a tarefa de densificação de profundidade guiada por imagem RGB através de um modelo profundo com estrutura de treinamento auto-supervisionada. Esse tipo de treinamento utiliza informações temporais por meio de sequências de imagem RGBD. Com o mesmo objetivo, Cheng, Wang e Yang (2018) desenvolveram um método baseado em uma Rede Convolucional de Propagação Espacial (CSPN) que estima profundidade por meio do aprendizado da matriz de afinidade, com relação aos pixels vizinhos, e que densifica os mapas de profundidade através do processo de propagação. Em seguida, Cheng et al. (2019) melhoraram o desempenho da CSPN por meio de processos adaptativos de convolução e de propagação.

Inspirada na CSPN, a arquitetura de rede proposta por Park et al. (2020) aprende as afinidades dos pixels estimados do mapa de profundidade, e suas confidências, em uma vizinhança não local. Em outra variação, Xu, Wang e Yao (2020) projetaram uma rede de propagação espacial deformável que se propaga com campos receptivos adaptativos e matrizes de afinidade.

Por outro lado, Chen et al. (2019) desenvolveu uma CNN de densificação de profundidade guiada que aprende recursos 2D-3D de forma multinível, enquanto Li et al. (2020) utilizaram blocos de redes *hourglass* alimentados com amostras esparsas e guiado pelos mapas de características do *backbone*. Recentemente, Tang et al. (2019) projetaram um método guiado que emprega kernels de convolução espacialmente variáveis. Outros trabalhos também se beneficiam de operadores morfológicos (DIMITRIEVSKI; VEELAERT; PHILIPS, 2018), informações adicionais de normais de superfície (QIU et al., 2019; XU et al., 2019), máscaras de fase (WU et al., 2019) e técnicas de guia cruzada (LEE et al., 2020).

3.3 Considerações finais

Pelos trabalhos apresentados, observa-se que diferentes tipos de arquiteturas de CNN supervisionadas foram exploradas para aplicações de estimação monocular de profundidade. Dentre essas variações de estruturas de rede, é possível citar estratégias relevantes como decodificação multi-escala (EIGEN; PUHRSCH; FERGUS, 2014), blocos de sobre-amostragem em formato residual (LAINA et al., 2016), métodos de classificação (CAO; WU; SHEN, 2017), recursos de regressão ordinal (FU et al., 2018), modelos gráficos (LIU et al., 2015), e camadas de orientação planar local (LEE et al., 2019).

Técnicas que se beneficiam de características temporais, como camadas Conv-LSTM (XINGJIAN et al., 2015), mecanismos baseados em atenção, como as redes Transformer (RANFTL; BOCHKOVSKIY; KOLTUN, 2021), e abordagens conjuntas de profundidade e semântica, como a rede de Jiao et al. (2018), também são artifícios explorados por métodos recentemente desenvolvidos que visam estimar profundidade de maneira monocular.

No entanto, as metodologias de SIDE que estão no estado-da-arte, geralmente, utilizam dados esparsos e ruidosos de profundidade como referência durante o treinamento, o que dificulta o processo de aprendizagem da rede (XU et al., 2017; KUMAR; BHANDARKAR; PRASAD, 2018a; HAO et al., 2018; RAMAMONJISOA; LEPETIT, 2019; YANG et al., 2021). Além disso, as CNNs que produzem os melhores resultados, como as de Alhashim e Wonka (2018), Fu et al. (2018), Yin et al. (2019), Lee et al. (2019), Bhat, Alhashim e Wonka (2020), Ranftl, Bochkovskiy e Koltun (2021), empregam redes complexas, com uma quantidade elevada de parâmetros treináveis, e exigem muito esforço computacional para serem treinadas e implementadas em uma aplicação real. Na Tabela 1, verifica-se que esses métodos supervisionados atingem uma velocidade de predição que não ultrapassa 26 *fps*, utilizando *hardwares* de processamento gráfico atuais e redes de extração de características com uma quantidade de parâmetros treináveis acima de 25 *M*.

No âmbito que trata sobre mecanismos de atenção, avanços podem ser atingidos, principalmente, acerca da forma com que a função de custo de treinamento leva em conta a distribuição de pixels de profundidade da cena (JIAO et al., 2018) e acerca da capacidade de generalização e tamanho total da rede (DOSOVITSKIY et al., 2020). Observando o contexto que emprega *ground truth* de semântica manualmente anotado (EIGEN; FERGUS, 2015; WANG et al., 2015; MOUSAVIAN; PIRSIYAVASH; KOŠECKÁ, 2016), têm-se que há lacunas relacionadas a forma como essas informações de referência são aproveitadas pela rede. O pré-processamento da CNN com *ground truth* de semântica é um exemplo de uma dessas lacunas.

A maior parte das metodologias de Aprendizagem Profunda que abordam profundidade e normais de superfície, com exceção da técnica proposta por Yin et al. (2019), estimam essas informações em ramos, *streams* e estruturas diferentes da rede (EIGEN;

FERGUS, 2015; WANG et al., 2016; BANSAL; RUSSELL; GUPTA, 2016; QI et al., 2018; ZHANG et al., 2019; RAMAMONJISOA; LEPETIT, 2019). Deste modo, novas formas de interação entre profundidade e normais de superfície podem ser aproveitadas, partindo-se da premissa de que elas são altamente correlacionadas.

Considerando o emprego de dados complementares de profundidade (MA; KARAMAN, 2018; MA; CAVALHEIRO; KARAMAN, 2019), nota-se que a precisão dos resultados pode ser melhorada, sobretudo pela forma como esses dados são amostrados, e que pouco ainda foi estudado sobre o comportamento de CNNs quando estas são expostas a informações adicionais de profundidade e de outras tarefas ao mesmo tempo.

Além de arquiteturas supervisionadas de SIDE, outros trabalhos têm focado em desenvolver soluções baseadas em aprendizagem semi-supervisionada e auto-supervisionada. Esses trabalhos argumentam que a elaboração de *ground truth* de profundidade é impraticável devido a necessidade de uma quantidade considerável de esforço humano durante sua construção.

No entanto, comparando os resultados obtidos pelos trabalhos de Yang et al. (2018), Luo et al. (2018b), Guo et al. (2018), Casser et al. (2019a), Chen et al. (2019), Ochs, Kretz e Mester (2019), Godard et al. (2019), Guizilini et al. (2020), Gur e Wolf (2019), Zhu et al. (2019), Watson et al. (2019) com os de Fu et al. (2018), Lee et al. (2019), Yin et al. (2019), Bhat, Alhashim e Wonka (2020), Ranftl, Bochkovskiy e Koltun (2021), nota-se que os métodos supervisionados ainda obtêm predições de profundidade mais precisas. Acrescentando-se a isso, CNNs auto-supervisionadas, em alguns casos, necessitam de técnicas de odometria visual (MAHJOURIAN; WICKE; ANGELOVA, 2018; CASSER et al., 2019b), de estimações de fluxo (YIN; SHI, 2018; ZOU; LUO; HUANG, 2018), de ambas as imagens do par estéreo durante o treinamento (GODARD et al., 2019; POGGI et al., 2018; TOSI et al., 2019), bem como de informações complementares de semântica e de normais de superfície (CASSER et al., 2019a; GUIZILINI et al., 2020; YANG et al., 2018; YANG et al., 2018) para poderem gerar resultados de melhor qualidade. Em outros casos, essas redes apresentam estruturas complexas, como é possível observar na Tabela 1, dificultando assim suas aplicações em situações reais.

Finalmente, os métodos recentes de densificação de profundidade têm aplicado novas estruturas de CNN que recebem apenas mapas esparsos (UHRIG et al., 2017; ELDESOKEY; FELSBURG; KHAN, 2019; HUANG et al., 2019) ou mapas esparsos e imagens RGB de guia como entrada (CHENG; WANG; YANG, 2018; CHENG et al., 2019; PARK et al., 2020; XU; WANG; YAO, 2020). Contudo, esses métodos não abordam a utilização dos mapas densificados como referência para o treinamento de redes de estimação monocular de profundidade. Logo, o processo de aprendizagem dessas redes pode ser auxiliado pelos mapas densos, com mais informações de profundidade, gerados por diferentes técnicas de densificação, como a que foi proposta no trabalho de Alhashim

e Wonka (2018).

Tendo isso em vista, este trabalho visa explorar os benefícios de redes supervisionadas, juntamente com outras técnicas de Aprendizagem Profunda, para solucionar problemas que envolvem a tarefa de SIDE. O próximo capítulo mostra como a proposta vigente está vinculada as arquiteturas de FCN, no estado-da-arte, que empregam estratégias supervisionadas para realizar estimação monocular de profundidade. Outros métodos de Aprendizagem Profunda, que compreendem as tarefas de segmentação semântica, estimação de normais de superfície e de densificação de profundidade, também são abordados neste trabalho e estão descritos detalhadamente nas próximas seções.

Tabela 1 – Especificações de trabalhos no estado-da-arte da tarefa de SIDE. O símbolo † indica o número total de parâmetros treináveis do *encoder*, enquanto ‡ refere-se ao tamanho total da estrutura de rede

Método	Dataset	GPU	Encoder	Parâmetros	Velocidade de predição
BTS (LEE et al., 2019)	KITTI Depth / NYU Depth V2	NVIDIA GTX 1080 Ti	DenseNet-161	47 M†	16 fps
DORN (FU et al., 2018)	KITTI Depth / Make3D / NYU Depth V2	NVIDIA Titan X Pascal	ResNet-101	110 M†	2 fps
VNL (YIN et al., 2019)	KITTI Depth / NYU Depth V2	Huawei GPU-accelerated Cloud Server	ResNext-101 (32 × 4d)	44 M†	2 fps
DenseDepth (ALHASHIM; WONKA, 2018)	KITTI Depth / NYU Depth V2	NVIDIA TITAN Xp	DenseNet-169	42 M†	-
ACAN (CHEN; ZHAO; HU, 2019)	KITTI Depth / NYU Depth V2	NVIDIA GTX 1080 Ti	ResNet-101	44 M†	-
PAP-Depth (ZHANG et al., 2019)	KITTI Depth / SUNRGBD / NYU Depth V2	NVIDIA P40	ResNet-50	25 M†	5 fps
DAEC (LI et al., 2018b)	ScanNet / KITTI Depth	NVIDIA GTX 1080	ResNext-101 (64 × 4d)	-	1 fps
APMoE (KONG; FOWLKES, 2019)	NYU Depth V2 / BSDS500 / Stanford-2D-3D / Cityscapes	NVIDIA Titan X	ResNet-50	25 M†	5 fps
CSWS (LI; DAI; HE, 2018)	NYU Depth V2 / KITTI Depth	NVIDIA Tesla Titan X	ResNet-152	60 M†	5 fps
Cao et al. (CAO; WU; SHEN, 2017)	NYU Depth V2 / KITTI Depth / SUNRGBD	-	ResNet-152	82 M†	-
Kuznietsov et al. (KUZNIETSOV; STUCKLER; LEIBE, 2017)	KITTI Depth	NVIDIA GTX 980 Ti	ResNet-50	80 M†	-
LSIM (GOLDMAN; HASSNER; AVIDAN, 2019)	KITTI Depth	NVIDIA Titan X	ResNet-50	25 M†	12 fps
DHGRl (ZHANG et al., 2018a)	KITTI Depth / Make3D / NYU Depth V2	NVIDIA K80	ResNet-50	25 M†	5 fps
SDNet (OCHS; KRETZ; MESTER, 2019)	KITTI Depth / Cityscapes	NVIDIA Titan Xp	ResNet-50	25 M†	5 fps
monoResMatch (TOSI et al., 2019)	KITTI Depth / Cityscapes	NVIDIA 2080 Ti	monoResMatch	42 M†	29 fps
3Net (POGGI; TOSI; MATTOCCIA, 2018)	KITTI Depth / Cityscapes	Jetson TX2	monoResMatch	42 M†	1 fps
3Net (POGGI; TOSI; MATTOCCIA, 2018)	KITTI Depth / Cityscapes	NVIDIA 2080 Ti	ResNet-50	25 M†	49 fps
VOMonodepth (ANDRAGHETTI et al., 2019)	KITTI Depth / Cityscapes	NVIDIA Jetson TX2	ResNet-50	25 M†	2 fps
VOMonodepth (ANDRAGHETTI et al., 2019)	KITTI Depth	NVIDIA 2080 Ti	ResNet-50	25 M†	41 fps
Monodepth (GODARD; AODHA; BROSTOW, 2017)	KITTI Depth	NVIDIA Jetson TX2	ResNet-50	25 M†	1 fps
Monodepth2 (GODARD et al., 2019)	KITTI Depth / Make3D	NVIDIA Titan X	ResNet-50	31 M†	28 fps
SA-Attention (ZHOu et al., 2019)	KITTI Depth / Make3D	NVIDIA Titan X	ResNet-18	14 M†	-
Struct2depth (CASSER et al., 2019a)	KITTI Depth / Cityscapes / Fetch Indoor Navigation	NVIDIA GTX 1080Ti	ResNet-50	34 M†	-
AdaBins (BHAT; ALHASHIM; WONKA, 2020)	KITTI Depth / NYU Depth V2 / SUNRGBD	NVIDIA V100	Struct2depth	14 M†	30 fps
DPT-Hybrid (RANFTL; BOCHKOVSKIY; KOLTUN, 2021)	DIW / ETH3D / Sintel / KITTI / NYU / TUM	NVIDIA RTX 2080	EfficientNet B5	78 M†	-
			ResNet-50 / ViT	123 M†	26 fps

4 METODOLOGIA

Neste capítulo, são explorados os conceitos fundamentais do método supervisionado proposto. Um dos objetivos principais da Seção 4.1 é apresentar os detalhes da arquitetura de rede convolucional desenvolvida, a qual é dividida nas fases de codificação e decodificação bem definidas. Juntamente a essa arquitetura, 4 modelos diferentes de *backbones*, ou *encoders*, são introduzidos na Seção 4.1.1. Esses modelos são utilizados em estudos de ablação descritos no Capítulo 5 a fim de verificar aquele que melhor se ajusta a estrutura de CNN base (*baseline*). Além disso, na Seção 4.2, são abordadas uma variedade significativa de funções de custo aplicadas nos estudos de estimação monocular de profundidade.

Os *pipelines* projetados para abordar densificação de profundidade, segmentação semântica e estimação de normais de superfície são descritos na Seção 4.1.3. No primeiro *pipeline*, emprega-se a estratégia de amostragem baseada na distribuição categórica para densificar mapas de profundidade esparsos e ruidosos para ambientes internos e externos.

Por meio de pequenas modificações na *baseline* proposta, a segunda abordagem utiliza pesos pré-treinados em segmentação semântica para aumentar a precisão da inferência de profundidade. Na terceira estratégia, um novo módulo de predição de normais de superfície é desenvolvido, o qual se beneficia dos mapas de profundidade retornados pela rede de SIDE, dos mapas de características produzidos pela CNN e de restrições de borda. Ademais, uma função de custo geométrica simples é introduzida para lidar com as predições de profundidade e de normais de superfície com eficiência. Técnicas de desfoque gaussiano são adicionadas ao módulo proposto para verificar sua robustez a entradas ruidosas.

Todas essas variações da *baseline*, introduzida na Seção 4.1, podem ser executadas a uma taxa de *frames* acima de 20. Finalmente, nas Seções 4.3, 4.4 e 4.5, respectivamente, apresenta-se as fases experimentais que englobam os detalhes de implementação, os *datasets* utilizados e os critérios de avaliação aplicados aos resultados. Assim, no Capítulo 5, são expostos os estudos de ablação, com análises quantitativas e qualitativas, e a conclusão encontra-se no Capítulo 6.

4.1 Arquitetura de rede

A arquitetura de FCN utilizada neste trabalho para estimar a profundidade de imagens RGB de maneira monocular segue o modelo de aprendizado supervisionado, i. e., as predições da rede são relacionadas ao *ground truth*, normalmente gerado a partir de sensores do tipo *structured-light/time-of-flight/phase-shift* e sensores LiDAR, por meio de uma função de custo de regressão. O *ground truth* de profundidade é composto por

Figura 32 – *Baseline* da CNN proposta

Fonte: do autor

imagens de apenas um canal, cujos pixels fornecem informações precisas das distâncias em metros dos objetos que compõem a cena. Por outro lado, também fazendo parte do escopo deste trabalho, o *ground truth* de semântica é formado por imagens rotuladas a nível de pixel, cujos valores estão vinculados as classes semânticas das estruturas que compõem as cenas, e os mapas de referência de normais de superfície são constituídos pelas coordenadas dos vetores ortonormais aos planos da cena.

A estrutura base para a rede proposta (*baseline*), ilustrada na Fig. 32, possui um formato de codificação-decodificação, em que a etapa de codificação é constituída de camadas convolucionais, além de camadas de *pooling*, as quais são os elementos fundamentais da DCNN base de extração de características, que também pode ser identificada como *encoder* ou *backbone*. Tal *baseline* é inspirada na arquitetura supervisionada de FCN desenvolvida por Lee et al. (2019) (Fig. 30), uma vez que este é um dos modelos que estão no estado-da-arte para a tarefa de SIDE. Contudo, modificou-se o modelo completo da Fig. 30 para ter até cinquenta e seis vezes menos parâmetros treináveis, já que não são empregadas camadas de LPG, blocos de subamostragem (*ds*) e Redução 1×1 (*reduc1x1*). Como outras modificações na etapa de codificação, incorpora-se redes de extração de características mais leves e rápidas e desconsidera-se informações adicionais como a distância focal da lente, bem como outros parâmetros intrínsecos e extrínsecos da câmera.

As próximas seções tratam sobre as partes principais da *baseline* da Fig. 32. A Seção 4.1.1 foca nos diferentes tipos de redes de extração de características que são utilizados na etapa de codificação da CNN da Fig. 32, destacando-se a arquitetura dos modelos

Figura 33 – Estrutura de CNN introduzida para a tarefa de preensão robótica

Fonte: adaptada de Ribeiro e Grassi (2019)

desenvolvidos para esta etapa. Por outro lado, na Seção 4.1.2, resalta-se os componetes da decodificação da *baseline*. Outros detalhes da CNN proposta, além de modificações menores e aplicações, podem ser encontrados no Capítulo 5.

4.1.1 Etapa de codificação da arquitetura de rede

Como DCNN base de extração de características, são utilizadas a MobileNet (SANDLER et al., 2018), a DenseNet-121 (HUANG et al., 2017), e novas variações da rede de preensão robótica desenvolvida por Ribeiro e Grassi (2019). As redes citadas têm 0,6M (até 4M), 6M e 0,3M (até 9M) de parâmetros treináveis, respectivamente, o que torna possível avaliar o *framework* completo a uma alta taxa de quadros sem perder a precisão. Implementou-se a MobileNet e a DenseNet-121 como *encoder* a fim de comparar o desempenho das variações propostas da rede de Ribeiro e Grassi (2019), na etapa de codificação da *baseline* da Fig. 32, com outras redes no estado-da-arte da tarefa de SIDE.

A CNN da Fig. 33 refere-se a uma das variações do modelo proposto por Ribeiro e Grassi (2019) que, devido a sua eficiência relativa ao seu número total de parâmetros, é aplicada como *backbone* na etapa de extração de características. Seguindo a arquitetura de Ribeiro e Grassi (2019), a quantidade de filtros ($\{32, 164, 96, 128\}$) e o tamanho do *kernel* (3×3) de cada convolução são mantidos, porém são removidas as operações de *flatten*, as camadas FC e a camada de saída. Outras alterações feitas na arquitetura em questão correspondem ao local das normalizações por lotes, as quais são inseridas entre as operações de convolução e as funções de ativação.

A configuração da rede da Fig. 33 em formato denso também é uma variação proposta. Como se pode observar pela Fig. 34, a estrutura padrão da DenseNet segue uma sequência de blocos densos de convolução intercalados com camadas de transição

Figura 34 – Modelo fundamental de DenseNet

Fonte: adaptada de Huang et al. (2017)

compostas por operações de convolução e de *pooling* pela média. Assim, as camadas 1, 2, 3 e 4 da arquitetura da Fig. 33 são transformadas em blocos densos e alternadas com camadas de transição.

No entanto, mantêm-se a relação de filtros ($\{32, 164, 96, 128\}$), o tamanho do *kernel* (3×3) e as operações de *pooling* pelo máximo, de acordo com o que é proposto por Ribeiro e Grassi (2019). Os detalhes dessa arquitetura estão expostos na Tabela 2, em que são comparadas as 3 configurações otimizadas da CNN proposta com a DenseNet-121 e a Densenet-161.

Nesta Tabela, k refere-se a taxa de crescimento dos filtros, que é somada ao final de cada bloco denso, θ corresponde a compressão dos filtros e t diz respeito ao número total de parâmetros treináveis. Além das diferenças apresentadas na Tabela 2, inicializa-se as camadas neurais das estruturas de rede propostas com a seguinte quantidade de filtros:

- Convolução Inicial: 32 filtros;
- Blocos Densos 1, 2 e 3: 164 filtros;
- Camadas de Transição 1, 2 e 3: 96 filtros;
- Bloco Denso 4: 128 filtros;

as primeiras convoluções dos blocos densos são configuradas com $2 \times$ o número atual de filtros.

4.1.2 Etapa de decodificação da arquitetura de rede

Após a etapa de extração de características pela DCNN base, os mapas densos são direcionados a um par de blocos de *upconvolution*, concatenação e convolução e, em

Tabela 2 – Composição dos modelos propostos para serem aplicados como *backbone* na etapa de extração de características da *baseline*. Para todos os modelos, k indica a taxa de crescimento dos filtros, θ é a compressão dos filtros e t é o número total de parâmetros

Camadas	Modelo 1 ($k = 6$, $\theta = 1$, $t = 1M$)	Modelo 2 ($k = 16$, $\theta = 1$, $t = 3M$)	Modelo 3 ($k = 24$, $\theta = 1$, $t = 9M$)	DenseNet-121 ($k = 32$, $\theta = 0.5$, $t = 7M$)	DenseNet-161 ($k = 48$, $\theta = 0.5$, $t = 27M$)
Convolução Inicial	3×3 conv, passo 2	3×3 conv, passo 2	3×3 conv, passo 2	7×7 conv, passo 2	7×7 conv, passo 2
Pooling Inicial	3×3 pooling pelo máximo, passo 2	3×3 pooling pelo máximo, passo 2	3×3 pooling pelo máximo, passo 2	3×3 pooling pelo máximo, passo 2	3×3 pooling pelo máximo, passo 2
Bloco Denso 1	3×3 conv, passo 1 3×3 conv, passo 1 3×3 conv, passo 1 $\times 6$	3×3 conv, passo 1 3×3 conv, passo 1 3×3 conv, passo 1 $\times 6$	3×3 conv, passo 1 3×3 conv, passo 1 3×3 conv, passo 1 $\times 6$	1×1 conv, passo 1 3×3 conv, passo 1 $\times 6$	1×1 conv, passo 1 3×3 conv, passo 1 $\times 6$
Camada de Transição 1	2×2 pooling pelo máximo, passo 2 3×3 conv, passo 1	2×2 pooling pelo máximo, passo 2 3×3 conv, passo 1	2×2 pooling pelo máximo, passo 2 3×3 conv, passo 1	1×1 conv, passo 1	1×1 conv, passo 1
Bloco Denso 2	3×3 conv, passo 1 3×3 conv, passo 1 3×3 conv, passo 1 $\times 12$	3×3 conv, passo 1 3×3 conv, passo 1 3×3 conv, passo 1 $\times 12$	3×3 conv, passo 1 3×3 conv, passo 1 3×3 conv, passo 1 $\times 12$	1×1 conv, passo 1 3×3 conv, passo 1 $\times 12$	1×1 conv, passo 1 3×3 conv, passo 1 $\times 12$
Camada de Transição 2	2×2 pooling pelo máximo, passo 2 3×3 conv, passo 1	2×2 pooling pelo máximo, passo 2 3×3 conv, passo 1	2×2 pooling pelo máximo, passo 2 3×3 conv, passo 1	1×1 conv, passo 1	1×1 conv, passo 1
Bloco Denso 3	3×3 conv, passo 1 3×3 conv, passo 1 3×3 conv, passo 1 $\times 18$	3×3 conv, passo 1 3×3 conv, passo 1 3×3 conv, passo 1 $\times 18$	3×3 conv, passo 1 3×3 conv, passo 1 3×3 conv, passo 1 $\times 24$	1×1 conv, passo 1 3×3 conv, passo 1 $\times 24$	1×1 conv, passo 1 3×3 conv, passo 1 $\times 36$
Camada de Transição 3	2×2 pooling pelo máximo, passo 2 3×3 conv, passo 1	2×2 pooling pelo máximo, passo 2 3×3 conv, passo 1	2×2 pooling pelo máximo, passo 2 3×3 conv, passo 1	1×1 conv, passo 1	1×1 conv, passo 1
Bloco Denso 4	3×3 conv, passo 1 3×3 conv, passo 1 3×3 conv, passo 1 $\times 16$	3×3 conv, passo 1 3×3 conv, passo 1 3×3 conv, passo 1 $\times 14$	3×3 conv, passo 1 3×3 conv, passo 1 3×3 conv, passo 1 $\times 18$	1×1 conv, passo 1 3×3 conv, passo 1 $\times 16$	1×1 conv, passo 1 3×3 conv, passo 1 $\times 24$

seguida, a um bloco de ASPP totalmente conectado (ASPP Denso), os quais substituem as últimas camadas de *pooling* da rede com o intuito de evitar a redução excessiva do campo receptivo durante o processo de geração dos mapas de características. O ASPP Denso é mantido da rede de Lee et al. (2019).

De acordo com Chen et al. (2017), Os módulos ASPP apresentam uma estrutura paralela de camadas de convolução *atrous*, com diferentes taxas de dilatação, as quais geram mapas de características com campos receptivos maiores e de múltiplos tamanhos sem alterar as dimensões do mapa e sem aumentar o número de parâmetros do *kernel*. No entanto, para imagens com alta resolução, é preciso aumentar a taxa de dilatação do módulo ASPP, de modo que o campo receptivo seja grande o suficiente para produzir as características dessas imagens apropriadamente (YANG et al., 2018a). Com o aumento da taxa de dilatação, o módulo ASPP perde eficiência e não produz os efeitos desejados no processo de extração de características.

Devido a isso, assim como em Yang et al. (2018a), este trabalho utiliza o módulo denso de ASPP que é composto por camadas de convolução dilatada em cascata e concatenadas. Como o ASPP denso não sofre os mesmos problemas de degradação do ASPP comum, a saída do *backbone* da rede da Fig. 32 é reconfigurada para ser acoplada ao módulo em questão, o qual possui as seguintes taxas de dilatação: 3, 6, 12, 18, 24.

Tendo isso em vista, a etapa de decodificação da rede recebe mapas de características maiores, com campos receptivos aumentados e de múltiplos tamanhos, e realiza menos processos de sobre-amostragem para recuperar o tamanho original da imagem (H) na saída da rede. Pela Fig. 32, observa-se que a etapa de decodificação é uma etapa que não acrescenta camadas neurais altamente complexas que poderiam aumentar o número de parâmetros treináveis e diminuir a velocidade de predição da rede.

Nesta etapa, os blocos de *upconvolution* são formados por convoluções com *kernel* 3×3 , as quais são precedidas de camadas que utilizam operações de sobre-amostragem por vizinhos mais próximos que aumentam as dimensões de altura e largura dos mapas de características em $2\times$. Tais blocos são seguidos por operações de normalização por lotes e, posteriormente, por *skip-connections* e concatenações entre os mapas de características sobre-amostrados e os mapas produzidos em camadas específicas do *encoder*. A configuração dos blocos de *upconvolution* também é mantida do trabalho de Lee et al. (2019).

Ao longo da estrutura da rede, são utilizadas funções de ativação do tipo ReLU (NAIR; HINTON, 2010; DAHL; SAINATH; HINTON, 2013; XU et al., 2015) e eLU (CLEVERT; UNTERTHINER; HOCHREITER, 2015), com exceção da camada de saída que apresenta a função sigmóide. O mapa de profundidade, obtido após a aplicação da função de ativação sigmóide, é multiplicado pelo valor máximo de profundidade que compreende o *dataset* utilizado (10 m para os *datasets indoor* (SILBERMAN et al., 2012) e 80 m para os *datasets outdoor* (UHRIG et al., 2017)).

Tabela 3 – Dimensões de entrada e saída das camadas da CNN proposta para imagens dos *datasets outdoor* e *indoor*

Camada	Dimensões - <i>outdoor</i> - (entrada/saída)	Dimensões - <i>indoor</i> - (entrada/saída)
Convolução Inicial	$(352 \times 704)/(176 \times 352)$	$(416 \times 544)/(208 \times 272)$
<i>Pooling</i> Inicial	$(176 \times 352)/(88 \times 176)$	$(208 \times 272)/(104 \times 136)$
Bloco Denso 1	$(88 \times 176)/(88 \times 176)$	$(104 \times 136)/(104 \times 136)$
Camada de Transição 1	$(88 \times 176)/(44 \times 88)$	$(104 \times 136)/(52 \times 68)$
Bloco Denso 2	$(44 \times 88)/(44 \times 88)$	$(52 \times 68)/(52 \times 68)$
Camada de Transição 2	$(44 \times 88)/(22 \times 44)$	$(52 \times 68)/(26 \times 34)$
Bloco Denso 3	$(22 \times 44)/(22 \times 44)$	$(26 \times 34)/(26 \times 34)$
Camada de Transição 3	$(22 \times 44)/(11 \times 22)$	$(26 \times 34)/(13 \times 17)$
Bloco Denso 4	$(11 \times 22)/(11 \times 22)$	$(13 \times 17)/(13 \times 17)$
<i>Upconvolution</i> 1	$(11 \times 22)/(22 \times 44)$	$(13 \times 17)/(26 \times 34)$
Convolução 1	$(22 \times 44)/(22 \times 44)$	$(26 \times 34)/(26 \times 34)$
<i>Upconvolution</i> 2	$(22 \times 44)/(44 \times 88)$	$(26 \times 34)/(52 \times 68)$
Convolução 2	$(44 \times 88)/(44 \times 88)$	$(52 \times 68)/(52 \times 68)$
ASPP Denso	$(44 \times 88)/(44 \times 88)$	$(52 \times 68)/(52 \times 68)$
<i>Upconvolution</i> 3	$(44 \times 88)/(88 \times 176)$	$(52 \times 68)/(104 \times 136)$
Convolução 3	$(88 \times 176)/(88 \times 176)$	$(104 \times 136)/(104 \times 136)$
<i>Upconvolution</i> 4	$(88 \times 176)/(176 \times 352)$	$(104 \times 136)/(208 \times 272)$
Convolução 4	$(176 \times 352)/(176 \times 352)$	$(208 \times 272)/(208 \times 272)$
<i>Upconvolution</i> 5	$(176 \times 352)/(352 \times 704)$	$(208 \times 272)/(416 \times 544)$
Convolução Final	$(352 \times 704)/(352 \times 704)$	$(416 \times 544)/(416 \times 544)$

A rede recebe imagens RGB com 352×704 e 416×544 pixels como entrada e retorna mapas de profundidade com 352×704 e 416×544 pixels em sua saída para os *datasets indoor* e *outdoor* respectivamente. Pela Tabela 3, é possível verificar as dimensões dos tensores na entrada e na saída de todas as camadas neurais do modelo da Fig. 32.

A seção seguinte introduz os *frameworks* que compreendem as tarefas de densificação de profundidade, segmentação semântica e estimação de normais de superfície. Tais *frameworks* são construídos a partir da *baseline*.

4.1.3 *Frameworks* derivados da arquitetura de rede

Além do desenvolvimento da arquitetura de FCN da Fig. 32, este trabalho visa analisar o comportamento dos resultados produzidos pela rede em questão quando esta é treinada e testada com dados adicionais de profundidade na entrada, os quais podem ser obtidos por meio de LiDARs 2D de baixo custo e das saídas de algoritmos de odometria visual e de SLAM (ANDRAGHETTI et al., 2019). Esse modelo de treinamento que aborda dados complementares de profundidade na entrada da rede pode ser observado na Fig. 35.

A simulação de algoritmos de odometria visual ou de sensores de profundidade utilizados como suporte é realizada por meio da técnica de amostragem baseada na distribuição multinomial com um número de experimentos independentes $n = 1$. Logo, apenas uma tentativa de amostragem aleatória de um dado esparsa de profundidade χ é executada para os pixels do *ground truth* vinculado as imagens de entrada da rede. Esse

Figura 35 – *Framework* para análise de informações adicionais de profundidade

Fonte: do autor

caso especial de distribuição multinomial é denominado de distribuição categórica, em que o número de categorias é fixado em $k = 2$. A fórmula 4.1 diz respeito a função massa de probabilidade associada a variável aleatória χ .

$$F(\chi | \varphi) = \prod_{i=0}^{k-1} \varphi_i^{\chi_i}, \quad \varphi_i > 0, \quad \sum_{i=0}^{k-1} \varphi_i = 1 \quad (4.1)$$

As probabilidades φ_1 e φ_0 referem-se, respectivamente, ao sucesso e ao fracasso de se amostrar um dado esparsos de profundidade. Portanto, assim como proposto por Ma e Karaman (2018), $\varphi_1 = \frac{d}{d^*}$ e $\varphi_0 = 1 - \frac{d}{d^*}$, em que d e d^* representam a quantidade desejada de amostras e a quantidade de dados válidos disponíveis por mapa de profundidade respectivamente.

Somando-se a isso, a influência de informações semânticas adicionais sobre a FCN proposta é analisada através do seu pré-treinamento com imagens constituídas de rótulos semânticos como *ground truth* (Fig. 36). Em seguida a esse pré-treinamento, o *framework* é treinado como uma rede de SIDE comum que recebe uma imagem RGB de entrada e retorna seu respectivo mapa de profundidade.

Tal estratégia é denominada de *fine-tuning*, em que a CNN é primeiramente treinada para segmentação semântica e parte dos pesos computados são transferidos para a rede de estimacão monocular de profundidade. A função de custo empregada para a tarefa de segmentação semântica pode ser representada pela Eq. 4.2, em que K é igual ao número total de categorias, y_i^* é o *ground truth* no formato de codificação *one-hot*, y_i é a predição da CNN, após a aplicação da função de ativação softmax, e t representa os elementos do mapa de características de entrada.

Apenas parte dos valores dos pesos obtidos com o *pipeline* de segmentação semântica

Figura 36 – *Framework* para análise de informações adicionais de semântica

Fonte: do autor

é ajustada na fase de estimação de profundidade devido às diferenças estruturais entre as duas CNNs, especialmente nas últimas camadas.

$$\mathcal{L}_{cce}(y, y^*) = - \sum_{i=1}^K y_i^* \log(y_i), \quad y_i = \frac{e^{t_i}}{\sum_j^K e^{t_j}} \quad (4.2)$$

Diferentemente dos trabalhos anteriores (WANG et al., 2016; QI et al., 2018; ZHANG et al., 2019) e mais próximo da abordagem de Yin et al. (2019), explora-se a correlação entre a estimação de profundidade e de normais de superfície em apenas um *framework*, o qual está exposto na Fig. 37. Nessa figura, nota-se que a estrutura de rede é alimentada por imagens RGB e retorna mapas de profundidade e de normais de superfície. Portanto, não são empregadas quaisquer ramificações extras ou outros modelos suplementares para processar as restrições geométricas 3D das normais de superfície. Ao invés disso, tais normais são inferidas por meio dos mapas de profundidade de saída e das mesmas camadas neurais usadas para SIDE.

A fim de treinar a CNN proposta para prever normais de superfície, o *ground truth* é obtido como nos trabalhos de Fouhey, Gupta e Hebert (2013), Qi et al. (2018), Zhang et al. (2019). Sendo assim, primeiramente, a nuvem de pontos no espaço 3D é recuperada dos mapas de profundidade, assumindo o modelo de câmera *pinhole* de acordo com a Eq. 4.3. Em tal equação, (v_i, ν_i) referem-se às posições dos pixels na imagem, (x_i, y_i, z_i) representa as coordenadas 3D da nuvem de pontos, onde os valores de z_i são iguais às profundidades da cena ρ_i , (o_x, o_y) é a localização do eixo óptico e (f_x, f_y) são as distâncias focais dos

Figura 37 – *Framework* para análise de informações adicionais de normais de superfície

Fonte: do autor

eixos (x, y) respectivamente.

$$x_i = \frac{z_i (\nu_i - o_x)}{f_x}, \quad y_i = \frac{z_i (\nu_i - o_y)}{f_y}, \quad z_i = \rho_i \quad (4.3)$$

Conforme proposto por Qi et al. (2018), após reconstruir a nuvem de pontos, determina-se os planos tangentes aos dados de profundidade da nuvem pela definição de uma vizinhança de pontos de tamanho m que pertence ao mesmo plano. Uma vez calculados os planos locais, os vetores normais a superfície $s = (s_x, s_y, s_z)$ podem ser obtidos com o auxílio da relação $Ds = c$, em que $D_{m \times 3}$ é a matriz de coordenadas da nuvem de pontos e $c_{m \times 1}$ é um vetor com valores constantes. Para minimizar $\|Ds - c\|_2^2$, a solução de mínimos quadrados $\hat{s}_{3 \times m}$ é aplicada ao sistema de equações normais da Eq. 4.4.

$$D^T D \hat{s} = D^T c \quad (4.4)$$

Pela Eq. 4.4, obtém-se à Eq. 4.5 que é utilizada para calcular diretamente os vetores ortonormais de superfície. Seguindo o que foi estabelecido por Qi et al. (2018), o vetor $c_{m \times 1}$ é definido com valores unitários.

$$\hat{s} = \frac{(D^T D)^{-1} D^T c}{\|(D^T D)^{-1} D^T c\|_2} \quad (4.5)$$

O módulo desenvolvido compartilha os mesmos mapas de características produzidos na etapa de SIDE, porém eles são concatenados com o mapa de profundidade retornado e com a saída do algoritmo de detecção de bordas de Sobel (NIXON; AGUADO, 2019). Este algoritmo é computacionalmente eficiente, é robusto ao ruído presente na imagem em escalas de cinza de entrada e recupera as características de alto valor do gradiente da

imagem, o que ajuda a CNN proposta a raciocinar melhor sobre as profundidades nos limites dos objetos da cena. As Eqs. 4.7 e 4.8 descrevem as máscaras H_x e H_y usadas pelo filtro Sobel para calcular a aproximação dos gradientes da imagem I (Eq. 4.6). Com a determinação do vetor gradiente, sua magnitude H e direção Ψ são calculadas para delimitar os pixels nas regiões de fronteira, os quais seguem a relação $H(i, j) > \tau$ em que τ é igual ao valor limite.

$$I_x = \frac{\partial I}{\partial x}, \quad I_y = \frac{\partial I}{\partial y} \quad (4.6)$$

$$I_x(i, j) = (I * H_x)(i, j), \quad I_y(i, j) = (I * H_y)(i, j) \quad (4.7)$$

$$H_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad H_y = -H_x^T, \quad (4.8)$$

$$H = \sqrt{I_x^2 + I_y^2}, \quad \Psi = \arctan\left(\frac{I_y}{I_x}\right)$$

Para validar a robustez a ruído do módulo de normais de superfície, uma camada convolucional com kernel gaussiano (AFSHARI; GADSDEN; HABIBI, 2017) é implementada antes do detector de bordas de Sobel. Essa camada convolucional é responsável por filtrar o ruído presente na imagem em escalas de cinza de entrada I , tornando a saída da etapa de detecção de bordas menos afetada por eles. O filtro Gaussiano linear proposto está livre de picos duplos no domínio da frequência e pode ser expresso pela Eq. 4.9, em que I_g é a imagem filtrada e G refere-se ao kernel $m \times n$ que obedece a uma distribuição Gaussiana com $\sigma = 1,76$.

$$I_g(i, j) = (I * G)(i, j) = \sum_m \sum_n G(m, n) I(i - m, j - n), \quad (4.9)$$

$$G(m, n) = \frac{1}{2\pi\sigma^2} e^{-\frac{m^2 + n^2}{2\sigma^2}}$$

Para a abordagem conjunta de estimação de profundidade e de normais de superfície, foi projetada a função de custo geométrica $\mathcal{L}_{2,5D}$, cuja expressão compreende as funções $L(y, y^*)$ e $\Phi(n, n^*)$. A primeira função é substituída por aquela que gera as predições de profundidade mais precisas. A segunda função calcula a similaridade do cosseno entre os pixels válidos das predições de normais de superfície n e do *ground truth* n^* . Além disso, o termo $\Phi(n, n^*)$ é ponderado por uma constante de proporcionalidade ψ .

$$\mathcal{L}_{2,5D} = L(y, y^*) + \psi(\Phi(n, n^*)), \quad (4.10)$$

$$\Phi(n, n^*) = 1 - \left(\frac{\langle n, n^* \rangle}{\|n\|_2 \|n^*\|_2} \right)$$

Os resultados produzidos por meio destas três diferentes análises e os resultados obtidos com o esquema exposto na Fig. 32 são comparados de maneira quantitativa e qualitativa no próximo capítulo. Acrescentando-se a isso, o desempenho da rede proposta, quando esta é treinada com mapas de referência densificados, também é apresentado no Capítulo 5.

4.2 Funções de custo

As funções de custo: Erro de Escala de Invariante ($\mathcal{L}_{SILog}$) (EIGEN; PUHRSCHE; FERGUS, 2014), Erro de Escala de Invariante Modificado ($\mathcal{L}_{SILog}^+$) (LEE et al., 2019), Huber ($\mathcal{L}_{Huber}$) (CARVALHO et al., 2018), BerHu ($\mathcal{L}_{BerHu}$) (ZWALD; LAMBERT-LACROIX, 2012), Erro Absoluto Médio ($\mathcal{L}_1$) (GODARD; AODHA; BROSTOW, 2017), Erro Quadrático Médio ($\mathcal{L}_2$) (FU et al., 2018), Charbonnier ($\mathcal{L}_{Charbo}$) (SUN; ROTH; BLACK, 2014) e Log-cosh ($\mathcal{L}_{cosh}$) (CHEN; CHEN; ZHANG, 2018) são utilizadas no treinamento da rede da Fig. 32. Os pixels sem informações de profundidade são desconsiderados pelas funções de custo, pois não influenciam o ajuste dos pesos da rede.

A função de custo $\mathcal{L}_{SILog}$ foi introduzida por Eigen, Puhersch e Fergus (2014) e tem sido frequentemente utilizada em CNNs de estimação monocular de profundidade, como nas redes de Lee et al. (2019), Kumar, Bhandarkar e Prasad (2018a), Carvalho et al. (2018), uma vez que ela é uma função eficiente para determinar as relações entre as profundidades de uma cena sem que sua escala influencie no cálculo (EIGEN; PUHRSCHE; FERGUS, 2014). Tal função de custo, modificada por Lee et al. (2019), pode ser expressa pela Eq. 4.12, na qual y_i^* representa os valores de profundidade do *ground truth*, y_i refere-se a predição da rede para o i – ésimo pixel e N é igual ao número total de amostras.

$$\mathcal{L}_{SILog}(h) = \frac{1}{N} \sum_i h_i^2 - \frac{\lambda}{N^2} \left(\sum_i h_i \right)^2, \quad (4.11)$$

$$h_i = \log(y_i) - \log(y_i^*), \quad \lambda = 0.85$$

$$\mathcal{L}_{SILog}^+ = \theta \sqrt{\mathcal{S}(h)}, \quad (4.12)$$

$$\mathcal{S}(h) = \frac{1}{N} \sum_{i=1}^N h_i^2 - \left(\frac{1}{N} \sum_{i=1}^N h_i \right)^2 + (1 - \lambda) \left(\frac{1}{N} \sum_{i=1}^N h_i \right)^2,$$

$$h_i = \log(y_i) - \log(y_i^*), \quad \lambda = 0.85, \quad \theta = 10.0$$

A BerHu, função inversa de $\mathcal{L}_{Huber}$ (Eq. 4.13), também é aplicada neste trabalho devido as suas vantagens vinculadas a combinação das funções de custo $\mathcal{L}_1$ e $\mathcal{L}_2$. Dentre

essas vantagens, têm-se que os residuais ($|y_i - y_i^*|$) com valores elevados são altamente penalizados pela função $\mathcal{L}_2$ que é sensível a *outliers*. Por outro lado, os residuais com valores baixos são penalizados pela função $\mathcal{L}_1$, que retorna valores de profundidade maiores do que aqueles retornados pela função $\mathcal{L}_2$. A função de custo BerHu pode ser expressa pela Eq. 4.14 com o termo $\kappa = 5$ padrão.

$$\mathcal{L}_{Huber}(h) = \begin{cases} |h| & |h| \geq g, \\ \frac{h^2 + g^2}{2g} & |h| < g, \end{cases} \quad (4.13)$$

$$h = \log(y) - \log(y^*), \quad g = \frac{1}{\kappa} \max_i (|\log(y_i) - \log(y_i^*)|)$$

$$\mathcal{L}_{BerHu}(h) = \begin{cases} |h| & |h| \leq g, \\ \frac{h^2 + g^2}{2g} & |h| > g \end{cases} \quad (4.14)$$

$$h = \log(y) - \log(y^*), \quad g = \frac{1}{\kappa} \max_i (|\log(y_i) - \log(y_i^*)|)$$

Além das funções de custo $\mathcal{L}_{SILog}$, $\mathcal{L}_{SILog}^+$, $\mathcal{L}_{BerHu}$ e $\mathcal{L}_{Huber}$, as funções $\mathcal{L}_1$ e $\mathcal{L}_2$, representadas pelas Eqs. 4.15 e 4.16, respectivamente, são utilizadas no treinamento da rede proposta, visto que elas são amplamente empregadas em CNNs no âmbito de SIDE (GODARD; AODHA; BROSTOW, 2017; MA; KARAMAN, 2018; FU et al., 2018). A função de custo Charbonnier, expressa pela Eq. 4.17, tem sido aplicada em problemas que envolvem regressão, principalmente em abordagens auto-supervisionadas de SIDE (BABU et al., 2018), e de estimação de fluxo (SUN; ROTH; BLACK, 2014). Os parâmetros a e α da Eq. 4.17 são ajustados de acordo com o que foi estabelecido no trabalho de Babu et al. (2018).

$$\mathcal{L}_1(h) = \frac{1}{N} \sum_{i=1}^N |h_i|, \quad h_i = \log(y_i) - \log(y_i^*) \quad (4.15)$$

$$\mathcal{L}_2(h) = \frac{1}{N} \sum_{i=1}^N (h_i)^2, \quad h_i = \log(y_i) - \log(y_i^*) \quad (4.16)$$

$$\mathcal{L}_{Charbo}(h) = \frac{1}{N} \sum_{i=1}^N (h_i^2 + \alpha^2)^a, \quad (4.17)$$

$$h_i = \log(y_i) - \log(y_i^*), \quad a = 0.45, \quad \alpha = 10^{-3}$$

A função Log-cosh (Eq. 4.18) apresenta um comportamento inverso ao da BerHu, funcionando como $\mathcal{L}_1$ para residuais altos e como $\mathcal{L}_2$ para residuais baixos. Isso torna seu

comportamento próximo ao da Huber, porém a Log-cosh tem a vantagem de ser totalmente diferenciável.

$$\mathcal{L}_{cosh}(h) = \sum_{i=1}^N \log(\cosh(h_i)), \quad h_i = y_i - y_i^* \quad (4.18)$$

A partir das funções de custo apresentadas, visa-se utilizar aquela que gera os melhores resultados nos testes da rede proposta em substituição ao termo $L(y, y^*)$ da função de custo baseada em atenção a profundidades distantes, Eq. 4.19, fundamentada no trabalho de Jiao et al. (2018). Essa função permite que a rede consiga focar em regiões onde a disponibilidade de pixels com informações de profundidade é menor, diminuindo os erros gerados por tais regiões, devido a distribuição desigual de pixels de profundidade, e mantendo a precisão para as regiões mais próximas.

A constante β , o termo de atenção γ em escala logarítmica e a função $\mathcal{L}_a$ em configuração de somatória são as modificações propostas com relação a expressão original (JIAO et al., 2018). Nessa abordagem, β é definido como 10, 0 e 1, 0 para os *datasets indoor* e *outdoor*, respectivamente, devido às variações de escala.

$$\mathcal{L}_a = \sum_{i=1}^N (\gamma + \epsilon) L(y, y^*), \quad (4.19)$$

$$\gamma = \frac{\log(y_i^*)}{\max_i(\log(y_i^*))}, \quad \epsilon = 1.0 - \frac{\min(\log(\beta y_i), \log(\beta y_i^*))}{\max(\log(\beta y_i), \log(\beta y_i^*))}$$

Por fim, as funções de custo $\mathcal{L}_1^+$ e $\mathcal{L}_2^+$ são semelhantes as funções $\mathcal{L}_1$ e $\mathcal{L}_2$, respectivamente, porém as primeiras são calculadas somente pela soma total do resultado da equação, ao invés do cálculo pela média. Com base em análises experimentais, essas variações da $\mathcal{L}_1$ e da $\mathcal{L}_2$ são aplicadas apenas durante o treinamento da rede com imagens dos *datasets indoor*.

$$\mathcal{L}_1^+(h) = \sum_{i=1}^N |h_i|, \quad h_i = \log(y_i) - \log(y_i^*) \quad (4.20)$$

$$\mathcal{L}_2^+(h) = \sum_{i=1}^N (h_i)^2, \quad h_i = \log(y_i) - \log(y_i^*) \quad (4.21)$$

4.3 Detalhes de implementação

Para implementar e treinar a rede proposta, utilizou-se um servidor com: CPU 4,20GHz, *hard drive* (HD) 2TB, *solid state drive* (SSD) 200GB, sistema operacional Ubuntu 16.04, *Framework* de desenvolvimento Tensorflow 1.13.1 e duas GPUs NVIDIA Titan X (as duas são usadas nas fases de treinamento e apenas uma nas fases de teste). Como

algoritmo de otimização, defini-se o ADAM (KINGMA; BA, 2014) como padrão para todos os treinamentos, o qual apresenta $\alpha = 10^{-3}$, $\beta_1 = 0,9$, $\beta_2 = 0,999$ e $\epsilon = 10^{-8}$, conforme proposto por Kingma e Ba (2014). Ademais, a taxa de aprendizagem é estabelecida com valor 10^{-4} e decaimento polinomial a uma taxa de 0,9. Em cada experimento, a rede é treinada por 50 épocas com um tamanho de *batch* igual a 16, exceto nos experimentos que consideram a MobileNetV2-50, a rede da Fig. 33 e o Modelo Proposto 1, em que o *batch* tem tamanho 32.

Apesar dos mapas de características gerados pelas primeiras camadas de convolução de CNNs apresentarem informações visuais primitivas, os parâmetros das normalizações por lotes e os pesos das duas primeiras convoluções da rede base de extração de características não são fixados (como ocorre nos métodos propostos por Fu et al. (2018), Lee et al. (2019)).

A Fig. 38 mostra as etapas de treinamento da rede da Fig. 32, desde os treinamentos preliminares com a *baseline* até os treinamentos finais com o *framework* de densificação de profundidade. A função de custo é alterada de acordo com as expressões expostas na seção 4.2 e o *backbone* é substituído pelos modelos apresentados na seção 4.1. A etapa de *Fine-tuning* compreende tanto o *pipeline* de segmentação semântica como os diferentes *datasets* que são discutidos na próxima seção (4.4).

O *framework* e as funções de custo empregados nos treinamentos que envolvem normais de superfície e densificação de profundidade também são apresentados na seção 4.1. As etapas de treinamento da Fig. 38 compreendem tanto cenas de ambientes externos como internos. Além dessas etapas principais, este trabalho aborda estudos de ablação sobre modificações estruturais menores no *decoder* da CNN desenvolvida, Odometria Visual e outros experimentos combinatórios disponíveis no capítulo de resultados (Capítulo 5).

A fim de evitar o sobre-ajuste da rede e com o intuito de aumentar artificialmente a quantidade de amostras de treinamento, são empregados três tipos de *data augmentation online*. Dentre os tipos de *data augmentation*, têm-se as operações de *flip* horizontal aleatório, com probabilidade de 50% de ocorrência, e de rotação no intervalo de $[-2, 5^\circ, 2, 5^\circ]$ e $[-1.0^\circ, 1.0^\circ]$ para os *datasets indoor* e *outdoor* respectivamente. Também são consideradas distorções de brilho $[0, 75, 1, 25]$, cor $[0, 9, 1, 1]$ e contraste $[0, 9, 1, 1]$ com probabilidade de 50% de ocorrência.

4.4 Bases de dados

4.4.1 KITTI Depth

Em trabalhos anteriores, o KITTI Raw (GEIGER et al., 2013) foi utilizado como *ground truth*, cujas imagens são obtidas diretamente das nuvens de pontos esparsas geradas pelo sensor LiDAR. No entanto, recentemente, o KITTI Vision Benchmark Suite disponibilizou o *dataset* oficial, KITTI Depth, que é composto por mapas de profundidade

Figura 38 – Etapas de treinamento do modelo proposto

Fonte: do autor

mais densos devido a combinação de 11 varreduras de *laser* (UHRIG et al., 2017). O KITTI Depth é constituído de imagens estéreo em escalas de cinza e coloridas, as quais foram captadas pelos dois pares de câmeras estéreo que compõem a plataforma de navegação autônoma capaz de percorrer ambientes externos para coletar dados.

Além dos pares de imagens estéreo, o *dataset* descrito disponibiliza as medições de profundidade geradas pelas varreduras do LiDAR Velodyne HDL-64E em formato de nuvens de pontos. Outro componente importante do KITTI Depth é o *ground truth* mais denso, em relação as nuvens de pontos, que apresenta um formato de imagem com apenas um canal, cujos pixels fornecem as informações de profundidade das cenas esquerda e direita do par de câmeras estéreo.

Tanto as imagens do par estéreo quanto o *ground truth* densificado apresentam uma resolução típica de 375×1242 pixels e são divididos em 61 cenas para treinamento e teste, as quais compreendem as categorias “*city*”, “*residential*”, “*road*” e “*campus*”. Em um primeiro momento, o KITTI Raw (GEIGER et al., 2013) foi subdividido no trabalho de Eigen, Puhersch e Fergus (2014) em 32 cenas diferentes para treinamento, contendo 23488 imagens, e 29 cenas distintas para teste, as quais contém 697 imagens ao todo. Trazendo essa subdivisão para o KITTI Depth (UHRIG et al., 2017), este trabalho visa utilizar 23158 imagens aleatoriamente selecionadas do total de 23488 imagens de treinamento. A Fig. 39 apresenta exemplos de amostras que compõem o *dataset* KITTI Depth.

Para a tarefa de segmentação semântica, o KITTI Vision Benchmark dispõe de 200 mapas anotados manualmente para treinamento e 200 imagens de teste com resolução típica de 375×1242 pixels. Devido a isso, dados adicionais de treinamento do *dataset*

Figura 39 – Exemplos de par imagem RGB e mapa de profundidade do *dataset* KITTI Depth

Fonte: do autor

Cityscapes (CORDTS et al., 2016) são incluídos. O Cityscapes é composto por 5K mapas semânticos (com resolução de 1024×2048 pixels), os quais são constituídos por anotações finas que formam um total de 30 classes diferentes, das quais apenas 19 são utilizadas. Das 5K imagens, 2975, 1525 e 500 são reservadas para treinamento, teste e validação respectivamente. Logo, neste trabalho, os 200 mapas semânticos de treinamento do KITTI são complementados pelos 3475 mapas do Cityscapes.

De forma correspondente aos 5K mapas semânticos, o Cityscapes também possui a mesma quantidade de mapas de disparidade, os quais podem ser utilizados para reconstruir seus respectivos mapas de profundidade. Devido a disponibilidade dos mapas de disparidade para treinamento, teste e validação, são empregados 5K mapas de profundidade reconstruídos do Cityscapes para pré-treinar a rede de SIDE proposta.

O *dataset* KITTI Odometry (GEIGER; LENZ; URTASUN, 2012) também é aplicado nos estudos de VO com a FCN proposta. Esse conjunto de dados é composto de 22 sequências com imagens estéreo de cenários externos. No entanto, apenas as primeiras 11 sequências têm suas respectivas trajetórias de referência, enquanto as restantes são usadas para avaliação de *benchmark*. Conforme proposto por Loo et al. (2019), este trabalho tem foco nas sequências estéreo com *ground truth* disponível.

4.4.2 NYU Depth V2

O NYU Depth V2 (SILBERMAN et al., 2012) é um *dataset* composto por pares de imagens RGB e mapas de profundidade, com resolução padrão de 480×640 pixels, captados pela câmera RGB e pelo sensor de profundidade de um Microsoft Kinect. Ao todo, o *dataset* oficial é formado por 240K pares de amostras RGBD de treinamento que integram 464 cenas de ambientes internos.

Assim como em outros trabalhos na literatura de SIDE (CHEN; ZHAO; HU, 2019; YIN et al., 2019; ALHASHIM; WONKA, 2018), objetiva-se utilizar a subdivisão oficial dos dados do NYU Depth V2 em 249 cenas de treinamento e 215 cenas para teste. Das 249 cenas de treinamento, que totalizam 120K imagens, 24931 imagens são selecionadas para treinar a rede proposta e, das 215 cenas de teste, 654 imagens são pré-definidas para testar tal rede. A Fig. 40 apresenta exemplos de amostras que constituem o *dataset* NYU Depth V2.

Ademais, 8587 imagens adicionais, provenientes dos *datasets* SUNRGBD (SONG; LICHTENBERG; XIAO, 2015), Berkeley B3DO (JANOCH et al., 2013) e SUN3D (XIAO; OWENS; TORRALBA, 2013), são empregadas em etapas de pré-treinamento. Para a tarefa de segmentação semântica, o NYU Depth V2 fornece 1449 imagens, com 35064 objetos diferentes, que apresentam anotações semânticas relacionadas a 894 categorias diferentes. Dessas categorias, 13 rótulos finos anotados são abordados durante as etapas de treinamento. Um total de 10336 amostras do *dataset* SUNRGBD (SONG; LICHTENBERG; XIAO, 2015) também são adicionadas, com seus respectivos rótulos semânticos, às 1449 imagens de treinamento disponíveis.

O *ground truth* de normais de superfície, tanto para o KITTI Depth como para o NYU Depth V2, é obtido de acordo com o que é proposto nos trabalhos de Fouhey, Gupta e Hebert (2013), Qi et al. (2018).

Figura 40 – Exemplos de imagens RGB, mapas de profundidade, mapas de semântica e de normais de superfície encontrados no *dataset* NYU Depth V2

Fonte: do autor

4.5 Métricas de avaliação

Como mencionado na subseção 4.4.1, a FCN da Fig. 32 é testada e avaliada de acordo com a subdivisão introduzida por Eigen, Puhrsch e Fergus (2014), denominada de *Eigen Split*. No entanto, como o *Eigen Split* foi originalmente proposto para o *dataset* KITTI Raw (GEIGER et al., 2013), apenas 652 das 697 imagens de teste apresentam *ground truth* no KITTI Depth. Acrescentando-se a isso, as predições da rede são avaliadas considerando-se o corte central estabelecido por Garg et al. (2016) e os intervalos de limitação de distância de 50m e de 80m. Para a avaliação da rede em ambientes internos, utiliza-se o corte central proposto por Eigen, Puhrsch e Fergus (2014) nas predições correspondentes as 654 imagens de teste do *dataset* NYU Depth V2, como discorrido na subseção 4.4.2.

Assim, para apresentar os resultados quantitativos da rede, as seguintes métricas são aplicadas na etapa de avaliação, em que T equivale à quantidade total de pixels com informações de profundidade (de acordo com o *ground truth*):

$$\text{Erro de Escala Invariante: Eq. 4.12 com } \lambda = 0.5 \quad (4.22)$$

$$\text{Limiar: \% de } y_i, \text{ tal que } \max\left(\frac{y_i}{y_i^*}, \frac{y_i^*}{y_i}\right) = \delta < \text{limiar} \quad (4.23)$$

$$\text{Diferença Relativa Absoluta: } \frac{1}{|T|} \sum_{y \in T} \frac{|y - y^*|}{y^*} \quad (4.24)$$

$$\text{Diferença Relativa Quadrática: } \frac{1}{|T|} \sum_{y \in T} \frac{\|y - y^*\|^2}{y^*} \quad (4.25)$$

$$\text{Log10: } \frac{1}{|T|} \sum_{y \in T} |\log_{10} y - \log_{10} y^*| \quad (4.26)$$

$$\text{Erro Absoluto Médio: } \frac{1}{|T|} \sum_{y \in T} |y - y^*| \quad (4.27)$$

$$\text{RMSE (linear): } \sqrt{\frac{1}{|T|} \sum_{y \in T} \|y - y^*\|^2} \quad (4.28)$$

$$\text{RMSE (log): } \sqrt{\frac{1}{|T|} \sum_{y \in T} \|\log y - \log y^*\|^2} \quad (4.29)$$

Os mapas de normais de superfície retornados pelo *pipeline* desenvolvido são avaliados de acordo com as métricas introduzidas por Fouhey, Gupta e Hebert (2013), as quais são comumente implementadas por trabalhos recentes (BANSAL; RUSSELL; GUPTA, 2016; WANG et al., 2016; QI et al., 2018; ZHANG et al., 2019). Tais métricas consideram a diferença angular para cada pixel válido entre as normais de superfície estimadas e do

ground truth através dos cálculos da média, da mediana, da Raiz do Erro Quadrático Médio (RMSE) e da precisão dos pixels estimados. Esta precisão é medida pela porcentagem de vetores cuja diferença angular não excede um limite $\tilde{n} = \{11, 25^\circ, 22, 5^\circ, 30^\circ\}$.

5 RESULTADOS

Nesse capítulo, são apresentados e discutidos os resultados obtidos com a FCN da Fig. 32, denominada de DenseSIDENet (DSN), a qual pode ser melhor detalhada pela Fig. 41. Por meio dessa ilustração, observa-se que os principais *backbones* propostos são essencialmente formados por convoluções, blocos densos, e operações de *pooling* pelo máximo.

No *decoder*, têm-se, primeiramente, blocos de convolução e de *upconvolution*, além do módulo de ASPP Denso. Tais blocos são seguidos por mais etapas de convolução e de *upconvolution* até a formação do mapa de profundidade de alta resolução de saída. Operações de normalização por lotes são realizadas entre convoluções, tanto no *encoder* como no *decoder*, a fim de reduzir o problema de *internal covariate shift*.

Na Fig. 41, nota-se também o local exato das *skip-connections* que reduzem o problema de *vanishing gradient* e que realizam o compartilhamento de mapas de características com diferentes níveis de abstração. A partir disso, divide-se a análise dos resultados da DSN em duas etapas principais que compreendem estudos com cenas de ambientes externos e internos.

5.1 Estudos de ablação em ambientes externos

5.1.1 Análises preliminares

Primeiramente, com relação às abordagens de densificação de profundidade para aprimorar os resultados de estimação de profundidade, utilizou-se a operação morfológica de fechamento para densificar parcialmente os mapas de referência do KITTI Depth (UHRIG et al., 2017), gerando o *dataset* KITTI Morphological (MENDES et al., 2020). Juntamente com a técnica de *closing*, aplicou-se o método RGB_guide&certainty, introduzido por Gansbeke et al. (2019), e a estratégia de densificação de profundidade de Hilbert Maps (ROSA; GUIZILINI; GRASSI, 2019) para produzir o KITTI Completed e o KITTI Continuous respectivamente.

Por meio das métricas do KITTI Benchmark, os resultados de densificação de profundidade pela aplicação de 5 iterações da técnica morfológica de fechamento são comparados a outros métodos da literatura na Tabela 4. Nessa Tabela, é possível notar que o operador de *closing*, com kernel 3×3 , é o método que melhor preserva as informações de profundidade do *ground truth*. No entanto, as técnicas baseadas em aprendizagem retornam mapas de profundidade totalmente densos, enquanto as de fechamento podem ser ajustadas para gerar imagens com diferentes níveis de densificação.

Na Tabela 5, pode-se inferir que quanto maior o número de iterações (*it*) mais denso

Figura 41 – Detalhes da *Baseline* proposta

Fonte: adaptada de (MENDES et al., 2021)

Tabela 4 – Comparação de métodos de densificação de profundidade no estado-da-arte de acordo com as métricas do KITTI Benchmark (UHRIG et al., 2017) ($iMAE = \frac{1}{MAE}$ e $iRMSE = \frac{1}{RMSE}$)

Método	MAE↓	RMSE↓	iMAE↓	iRMSE↓
SGDU (SCHNEIDER et al., 2016)	605,470	2312,570	2,050	7,380
NN+CNN (UHRIG et al., 2017)	416,140	1419,750	1,290	3,250
ADNN (CHODOSH; WANG; LUCEY, 2018)	439,480	1325,370	3,190	59,390
IP-Basic (KU; HARAKEY; WASLANDER, 2018)	302,600	1288,460	1,290	3,780
DFuseNet (SHIVAKUMAR et al., 2019)	429,930	1206,660	1,790	3,620
VOICED (WONG et al., 2020)	299,410	1169,970	1,200	3,560
Morph-Net (DIMITRIEVSKI; VEELAERT; PHILIPS, 2018)	310,490	1045,450	1,570	3,840
CSPN (CHENG; WANG; YANG, 2018)	279,460	1019,640	1,150	2,930
DCrgb_80b_3coef (IMRAN et al., 2019)	215,750	965,870	0,980	2,430
Conf-Net (HEKMATIAN; AL-STOUHI; JIN, 2019)	257,540	962,280	1,090	3,100
DFineNet (ZHANG et al., 2019)	304,170	943,890	1,390	3,210
Spade-RGBsD (JARITZ et al., 2018)	234,810	917,640	0,950	2,170
IR_L2 (LU et al., 2020)	292,360	901,430	1,350	4,920
DDP (YANG; WONG; SOATTO, 2019)	203,960	832,940	0,850	2,100
NConv (ELDESOKEY; FELSBURG; KHAN, 2019)	233,260	829,980	1,030	2,600
Sparse-to-Dense (MA; CAVALHEIRO; KARAMAN, 2019)	249,950	814,730	1,210	2,800
CrossGuidance (LEE et al., 2020)	253,980	807,420	1,330	2,730
Revisiting NN+CNN (YAN; LIU; BELYAEV, 2020)	225,810	792,800	0,990	2,420
PwP (XU et al., 2019)	235,170	777,050	1,130	2,420
RGB_guide&certainty (GANSBEKE et al., 2019)	215,020	772,870	0,930	2,190
DSPN (XU; WANG; YAO, 2020)	220,360	766,740	1,030	2,470
MSG-CHN (LI et al., 2020)	220,410	762,190	0,980	2,300
DeepLiDAR (QIU et al., 2019)	226,500	758,380	1,150	2,560
UberATG-FuseNet (CHEN et al., 2019)	221,190	752,880	1,140	2,340
CSPN++ (CHENG et al., 2019)	209,280	743,690	0,900	2,070
NLSPN (PARK et al., 2020)	199,590	741,680	0,840	1,990
GuideNet (TANG et al., 2019)	218,830	736,240	0,990	2,250
<i>closing</i> ($k = 3$, $it = 5$)	148,963	519,606	0,597	1,082
<i>closing</i> ($k = 3$, $it = 4$)	129,353	457,346	0,533	0,972
<i>closing</i> ($k = 3$, $it = 3$)	106,925	391,138	0,458	0,867
<i>closing</i> ($k = 3$, $it = 2$)	82,137	326,323	0,363	0,747
<i>closing</i> ($k = 3$, $it = 1$)	46,724	226,214	0,211	0,541

se torna o *dataset* KITTI Morphological. Além disso, tem-se que o KITTI Completed é o conjunto de dados mais denso, uma vez que ele é gerado através da abordagem baseada em aprendizagem RGB_guide&certainty. Os resultados expostos na Tabela 6 mostram que o *pipeline* desenvolvido atinge um desempenho melhor quando treinado em conjunto com o *dataset* parcialmente denso e a função de custo BerHu.

No entanto, a precisão do método diminui quando ele é treinado com os conjuntos de dados KITTI Completed e KITTI Continuous devido à presença de maiores distorções nos mapas de profundidade de referência. Ademais, embora o KITTI Continuous seja menos denso que o KITTI Completed, ele inclui uma quantidade maior de artefatos que dificultam o aprendizado da rede. Portanto, o KITTI Morphological apresenta o melhor equilíbrio entre densificação e alteração nos valores originais do *ground truth*, de modo que a DSN consegue entender melhor a estrutura da cena.

Tabela 5 – Análise dos graus de densificação dos *datasets* utilizados de acordo com o número médio e a porcentagem média de pixels válidos. A resolução 375×1242 é típica para o KITTI Depth (UHRIG et al., 2017) e 352×704 é o tamanho da imagem que é alimentada no *framework*

Dataset	Número médio de pixels válidos		Porcentagem média de pixels válidos	
	375×1242	352×704	375×1242	352×704
KITTI Depth	75407	43649	16,190%	17,614%
KITTI Morphological ($k = 3$, $it = 1$)	142309	82377	30,555%	33,242%
KITTI Morphological ($k = 3$, $it = 2$)	166765	96525	35,806%	38,951%
KITTI Morphological ($k = 3$, $it = 3$)	182949	105884	39,280%	42,728%
KITTI Morphological ($k = 3$, $it = 4$)	195587	113172	41,994%	45,669%
KITTI Morphological ($k = 3$, $it = 5$)	205478	118875	44,118%	47,971%
KITTI Continuous	291489	158255	62,585%	63,862%
KITTI Completed	428031	247807	91,901%	100%

5.1.2 Estudos estruturais

Os seguintes estudos são realizados com o auxílio do KITTI Morphological, na configuração $k = 3$ e $it = 2$, visto que ele é benéfico para o treinamento da DSN. A fim de explorar a resposta do *pipeline* proposto a outras redes como *encoder*, foram realizados novos experimentos, cujos resultados são apresentados na Tabela 7. A partir das métricas da Tabela 7, pode-se observar que a DSN, juntamente com o Modelo 3 como *backbone*, atinge o melhor *trade-off*, considerando o número de parâmetros treináveis, a velocidade de predição e a precisão da CNN. Analisando os métodos com tamanhos comparáveis, é possível afirmar que todas as redes leves de extração de características (Fig. 33, Modelo 1, Modelo 2 e Modelo 3) ultrapassam a MobileNetV2 e a DenseNet-121 no estado-da-arte e não geram problemas de *underfitting*.

Para aprimorar o desempenho da DSN, com o Modelo 3 como rede de extração de características (DSN 3), foram estabelecidas modificações na estrutura do *decoder* da CNN e em seu procedimento de treinamento. Por meio da Tabela 8, é possível verificar que o uso da função de custo baseada em atenção, combinada com a expressão $\mathcal{L}_{BerHu}$, diminui a precisão do *framework*, enquanto a aplicação da função $\mathcal{L}_{BerHu}^+$ melhora os resultados. Isso indica que a BerHu já é capaz de equilibrar as penalidades para os residuais gerados em regiões mais próximas e distantes da cena, sem a necessidade de um termo de atenção. Além disso, o aprendizado da rede é potencializado quando o termo $\mathcal{L}_1$ da BerHu atua em uma faixa mais ampla de residuais, considerando o *dataset* KITTI Morphological.

Os módulos de pirâmide são blocos simples que concatenam mapas de características em várias escalas da DSN 3. No *decoder*, o módulo Pyramid¹ concatena os mapas de características ampliados dos últimos 3 blocos de convolução com a saída da *upconvolution* final, enquanto o Pyramid² adiciona os mapas de características da primeira convolução a

Tabela 6 – Avaliação do método proposto considerando variações na função de custo e no *dataset* utilizado durante o treinamento. A DenseNet-121 é fixada como a rede de extração de características no *encoder* da DSN

Método	Custo	Dataset	Cap	Abs Rel↓	Sqr Rel↓	RMSE↓	RMSE (log)↓	SILog↓	log10↓	$\delta < 1, 25\uparrow$	$\delta < 1, 25^2\uparrow$	$\delta < 1, 25^3\uparrow$
DenseNet-121	$\mathcal{L}_2$	KITTI Depth	0 – 80 m	0,114	0,663	4,367	0,173	15,890	0,050	0,853	0,963	0,989
DenseNet-121	$\mathcal{L}_1$	KITTI Depth	0 – 80 m	0,108	0,624	4,200	0,166	15,294	0,047	0,868	0,965	0,990
DenseNet-121	$\mathcal{L}_{Charbo}$	KITTI Depth	0 – 80 m	0,108	0,639	4,255	0,167	15,323	0,047	0,866	0,966	0,989
DenseNet-121	$\mathcal{L}_{Huber}$	KITTI Depth	0 – 80 m	0,099	0,538	4,076	0,154	14,308	0,044	0,883	0,972	0,992
DenseNet-121	$\mathcal{L}_{SILog}$	KITTI Depth	0 – 80 m	0,097	0,513	3,939	0,150	13,569	0,043	0,890	0,975	0,993
DenseNet-121	$\mathcal{L}_{cosh}$	KITTI Depth	0 – 80 m	0,096	0,518	3,855	0,150	13,800	0,042	0,892	0,974	0,993
DenseNet-121	$\mathcal{L}_{SILog}^+$	KITTI Depth	0 – 80 m	0,095	0,494	3,856	0,146	13,214	0,042	0,894	0,976	0,994
DenseNet-121	$\mathcal{L}_{BerHu}$	KITTI Depth	0 – 80 m	0,089	0,478	3,788	0,142	13,004	0,040	0,903	0,977	0,994
DenseNet-121	$\mathcal{L}_{BerHu}$	KITTI Morphological ($k = 3, it = 1$)	0 – 80 m	0,088	0,471	3,792	0,141	12,750	0,040	0,903	0,979	0,994
DenseNet-121	$\mathcal{L}_{BerHu}$	KITTI Morphological ($k = 3, it = 2$)	0 – 80 m	0,088	0,461	3,780	0,140	12,747	0,040	0,905	0,978	0,994
DenseNet-121	$\mathcal{L}_{BerHu}$	KITTI Morphological ($k = 3, it = 3$)	0 – 80 m	0,088	0,458	3,800	0,142	12,860	0,040	0,903	0,977	0,994
DenseNet-121	$\mathcal{L}_{BerHu}$	KITTI Morphological ($k = 3, it = 4$)	0 – 80 m	0,088	0,479	3,855	0,143	13,024	0,040	0,904	0,976	0,993
DenseNet-121	$\mathcal{L}_{BerHu}$	KITTI Morphological ($k = 3, it = 5$)	0 – 80 m	0,088	0,474	3,881	0,143	13,053	0,040	0,903	0,977	0,994
DenseNet-121	$\mathcal{L}_{BerHu}$	KITTI Continuous	0 – 80 m	1,640	36,498	24,577	0,978	21,627	0,415	0,013	0,036	0,072
DenseNet-121	$\mathcal{L}_{BerHu}$	KITTI Completed	0 – 80 m	0,220	1,741	6,269	0,249	20,577	0,084	0,698	0,924	0,980
DenseNet-121	$\mathcal{L}_2$	KITTI Depth	0 – 50 m	0,110	0,509	3,203	0,161	14,807	0,047	0,867	0,970	0,992
DenseNet-121	$\mathcal{L}_1$	KITTI Depth	0 – 50 m	0,104	0,480	3,102	0,155	14,293	0,045	0,881	0,970	0,992
DenseNet-121	$\mathcal{L}_{Charbo}$	KITTI Depth	0 – 50 m	0,103	0,487	3,122	0,156	14,277	0,045	0,879	0,972	0,991
DenseNet-121	$\mathcal{L}_{Huber}$	KITTI Depth	0 – 50 m	0,094	0,393	2,892	0,142	13,165	0,041	0,897	0,978	0,994
DenseNet-121	$\mathcal{L}_{SILog}$	KITTI Depth	0 – 50 m	0,093	0,377	2,833	0,139	12,542	0,041	0,904	0,980	0,995
DenseNet-121	$\mathcal{L}_{cosh}$	KITTI Depth	0 – 50 m	0,092	0,390	2,817	0,140	12,839	0,040	0,904	0,979	0,994
DenseNet-121	$\mathcal{L}_{SILog}^+$	KITTI Depth	0 – 50 m	0,090	0,365	2,771	0,136	12,229	0,040	0,907	0,981	0,996
DenseNet-121	$\mathcal{L}_{BerHu}$	KITTI Depth	0 – 50 m	0,085	0,352	2,721	0,131	11,981	0,037	0,916	0,982	0,995
DenseNet-121	$\mathcal{L}_{BerHu}$	KITTI Morphological ($k = 3, it = 1$)	0 – 50 m	0,084	0,342	2,732	0,130	11,723	0,037	0,916	0,983	0,996
DenseNet-121	$\mathcal{L}_{BerHu}$	KITTI Morphological ($k = 3, it = 2$)	0 – 50 m	0,083	0,335	2,701	0,129	11,722	0,037	0,919	0,983	0,996
DenseNet-121	$\mathcal{L}_{BerHu}$	KITTI Morphological ($k = 3, it = 3$)	0 – 50 m	0,083	0,331	2,688	0,130	11,784	0,037	0,916	0,982	0,995
DenseNet-121	$\mathcal{L}_{BerHu}$	KITTI Morphological ($k = 3, it = 4$)	0 – 50 m	0,083	0,345	2,729	0,131	11,906	0,037	0,917	0,982	0,995
DenseNet-121	$\mathcal{L}_{BerHu}$	KITTI Morphological ($k = 3, it = 5$)	0 – 50 m	0,083	0,338	2,744	0,131	11,912	0,037	0,916	0,983	0,995
DenseNet-121	$\mathcal{L}_{BerHu}$	KITTI Continuous	0 – 50 m	1,545	28,857	19,418	0,939	23,091	0,395	0,029	0,069	0,128
DenseNet-121	$\mathcal{L}_{BerHu}$	KITTI Completed	0 – 50 m	0,215	1,454	5,163	0,240	19,124	0,082	0,709	0,933	0,983

Tabela 7 – Comparação dos resultados obtidos pelo framework proposto com diferentes redes de extração de características no *dataset* KITTI Depth (UHRIG et al., 2017). O Modelo 1, o Modelo 2 e o Modelo 3 referem-se a DenseSIdENet com os *encoders* da Tabela 2. A MobileNetV2-50 e a MobileNetV2-140 são versões da MobileNetV2 com multiplicadores de profundidade 1, 4 e 0, 5 respectivamente. A função de custo BerHu é empregada em todos os experimentos e a coluna Mapa indica o número total de mapas de características de entrada para o estágio de decodificação da DSN

Método	Mapa	Tamanho	Velocidade	Cap	Abs Rel $\downarrow$	Sqr Rel $\downarrow$	RMSE $\downarrow$	RMSE (log) $\downarrow$	SILog $\downarrow$	log10 $\downarrow$	$\delta < 1, 25^\uparrow$	$\delta < 1, 25^2\uparrow$	$\delta < 1, 25^3\uparrow$
MobileNetV2-50	128	2 M	87 <i>fps</i>	0 – 80 <i>m</i>	0,122	0,760	4,612	0,184	17,074	0,052	0,842	0,955	0,986
MobileNetV2-50	256	5 M	81 <i>fps</i>	0 – 80 <i>m</i>	0,116	0,729	4,508	0,178	16,326	0,051	0,853	0,959	0,987
MobileNetV2-140	256	10 M	62 <i>fps</i>	0 – 80 <i>m</i>	0,104	0,596	4,084	0,161	14,799	0,045	0,877	0,969	0,991
MobileNetV2-140	512	20 M	39 <i>fps</i>	0 – 80 <i>m</i>	0,100	0,575	4,065	0,158	14,206	0,044	0,883	0,970	0,991
DenseNet-121	256	12 M	41 <i>fps</i>	0 – 80 <i>m</i>	0,088	0,461	3,780	0,140	12,747	0,040	0,905	0,978	0,994
Fig. 33 CNN	256	3 M	88 <i>fps</i>	0 – 80 <i>m</i>	0,102	0,594	4,094	0,160	14,797	0,045	0,876	0,968	0,991
Modelo 1	256	3 M	59 <i>fps</i>	0 – 80 <i>m</i>	0,093	0,517	3,941	0,149	13,776	0,041	0,895	0,974	0,992
Modelo 2	256	6 M	57 <i>fps</i>	0 – 80 <i>m</i>	0,091	0,501	3,851	0,146	13,243	0,041	0,901	0,977	0,993
Modelo 3	256	12 M	50 <i>fps</i>	0 – 80 <i>m</i>	0,083	0,434	3,646	0,135	12,326	0,037	0,914	0,980	0,994
MobileNetV2-50	128	2 M	87 <i>fps</i>	0 – 50 <i>m</i>	0,117	0,592	3,431	0,172	15,892	0,049	0,856	0,961	0,989
MobileNetV2-50	256	5 M	81 <i>fps</i>	0 – 50 <i>m</i>	0,111	0,560	3,339	0,166	15,139	0,048	0,866	0,965	0,989
MobileNetV2-140	256	10 M	62 <i>fps</i>	0 – 50 <i>m</i>	0,100	0,463	3,044	0,151	13,810	0,043	0,889	0,974	0,993
MobileNetV2-140	512	20 M	39 <i>fps</i>	0 – 50 <i>m</i>	0,095	0,434	2,973	0,146	13,172	0,042	0,896	0,975	0,993
DenseNet-121	256	12 M	41 <i>fps</i>	0 – 50 <i>m</i>	0,083	0,335	2,701	0,129	11,722	0,037	0,919	0,983	0,996
Fig. 33 CNN	256	3 M	88 <i>fps</i>	0 – 50 <i>m</i>	0,098	0,456	3,030	0,150	13,796	0,043	0,888	0,974	0,993
Modelo 1	256	3 M	59 <i>fps</i>	0 – 50 <i>m</i>	0,089	0,380	2,843	0,138	12,712	0,039	0,907	0,980	0,994
Modelo 2	256	6 M	57 <i>fps</i>	0 – 50 <i>m</i>	0,086	0,370	2,789	0,134	12,186	0,038	0,914	0,981	0,994
Modelo 3	256	12 M	50 <i>fps</i>	0 – 50 <i>m</i>	0,079	0,314	2,592	0,124	11,315	0,035	0,926	0,984	0,996

essa concatenação. Em relação às métricas apresentadas na Tabela 8, ambos os módulos de pirâmide são benéficos para o *framework* proposto, porém eles diminuem a velocidade de processamento da rede.

Além disso, os resultados obtidos quando a DSN 3 é pré-treinada com mapas semânticos e de disparidade, os quais são convertidos em mapas de profundidade durante o treinamento, mostram que ambos os pré-treinamentos melhoram os valores das métricas. Embora a DSN 3 adquira pistas importantes dos mapas semânticos, como as bordas de objetos, as características monoculares aprendidas na mesma tarefa de estimação de profundidade causam melhorias mais significativas. Todas as modificações estruturais no decodificador do *pipeline* proposto reduzem sua velocidade de inferência, porém o número de parâmetros treináveis não é consideravelmente acrescido.

5.1.3 Análises de normais de superfície

Testou-se o módulo de normais de superfície na rede DSN 3, juntamente com: o módulo Pyramid¹, a etapa de detecção de borda, a função de custo geométrica 2,5D e os pesos pré-treinados nos mapas de profundidade de referência do Cityscapes (CORDTS et al., 2016), os quais são produzidos a partir de seus respectivos mapas de disparidade. Com base na métrica Abs Rel da Tabela 9, pode-se verificar que a configuração de rede com $\psi_1 = 10^6$, $\psi_2 = 10^6$ e sem a presença do filtro Gaussiano é aquela que gera os menores erros, implicando que o detector de bordas do módulo de normais de superfície é robusto ao ruído na imagem de entrada.

Considerando os erros e a precisão dos métodos expostos na Tabela 9, têm-se que as pistas geométricas, fornecidas pelo *ground truth* de normais de superfície e computadas pela função de custo geométrica, suportam a DSN na predição de mapas 2,5D mais consistentes com a cena. O módulo de normais de superfície tem pouca influência no tamanho total da FCN e pode ser utilizado em outros métodos que lidam com problemas de SIDE. Apenas a velocidade de predição é afetada negativamente pelo uso da estratégia discorrida.

Uma visão geral dos trabalhos que abordam a tarefa SIDE e que avaliam seus métodos no KITTI Depth (UHRIG et al., 2017) pode ser encontrada na Tabela 10. Os resultados obtidos pela DSN são comparáveis aqueles gerados por métodos no estado-da-arte com diferentes tipos de procedimentos de treinamento. Com base na Tabela 1, a CNN proposta tem menos parâmetros treináveis e, devido à sua velocidade de predição, pode ser aplicada em cenários reais de navegação autônoma.

Os mapas de profundidade retornados pela configuração da DSN da Tabela 10 são ilustrados na Fig. 42f. Em comparação com os métodos das Figs. 42c, 42d e 42e, a CNN proposta é capaz de identificar melhor a forma dos artefatos que compõem a cena, produzindo contornos mais suaves e bem demarcados, inclusive em regiões próximas a

Tabela 8 – Comportamento da DSN quando sujeita a modificações em sua estrutura e processo de treinamento. A DSN 3 é a *CNN baseline* com o Modelo 3 como codificador. O Pyramid¹ e o Pyramid² são módulos em forma de pirâmide implementados no decodificador. CS e Sem referem-se ao pré-treinamento com as disparidades do conjunto de dados do Cityscapes (CORDTS et al., 2016) e com os mapas semânticos do KITTI (ALHAIJA et al., 2018) e do Cityscapes (CORDTS et al., 2016) respectivamente. A expressão $\mathcal{L}_{BerHu}^+$ é igual à $\mathcal{L}_{BerHu}$, porém com $\kappa = 4$

Método	Custo	Tamanho	Velocidade	Cap	Abs Rel↓	Sqr Rel↓	RMSE↓	RMSE (log)↓	SILog↓	log10↓	$\delta < 1,25^\uparrow$	$\delta < 1,25^2^\uparrow$	$\delta < 1,25^3^\uparrow$
DSN 3	$\mathcal{L}_{BerHu}$	12 M	50 fps	0 – 80 m	0,083	0,434	3,646	0,135	12,326	0,037	0,914	0,980	0,994
DSN 3	$\mathcal{L}_a + \mathcal{L}_{BerHu}$	12 M	50 fps	0 – 80 m	0,084	0,437	3,621	0,133	12,259	0,037	0,914	0,981	0,994
DSN 3	$\mathcal{L}_{BerHu}^+$	12 M	50 fps	0 – 80 m	0,083	0,431	3,628	0,133	12,146	0,037	0,915	0,982	0,995
DSN 3 + Pyramid ¹	$\mathcal{L}_{BerHu}$	12 M	39 fps	0 – 80 m	0,082	0,418	3,593	0,132	12,122	0,037	0,919	0,981	0,995
DSN 3 + Pyramid ¹ + Sem	$\mathcal{L}_{BerHu}$	12 M	39 fps	0 – 80 m	0,081	0,414	3,557	0,133	12,287	0,036	0,916	0,980	0,995
DSN 3 + Pyramid ¹ + CS	$\mathcal{L}_{BerHu}^+$	12 M	39 fps	0 – 80 m	0,078	0,405	3,399	0,125	11,490	0,034	0,926	0,985	0,995
DSN 3 + Pyramid ² + CS	$\mathcal{L}_{BerHu}^+$	12 M	32 fps	0 – 80 m	0,078	0,376	3,344	0,124	11,323	0,034	0,926	0,985	0,996
DSN 3	$\mathcal{L}_{BerHu}$	12 M	50 fps	0 – 50 m	0,079	0,314	2,592	0,124	11,315	0,035	0,926	0,984	0,996
DSN 3	$\mathcal{L}_a + \mathcal{L}_{BerHu}$	12 M	50 fps	0 – 50 m	0,080	0,319	2,594	0,123	11,267	0,035	0,927	0,985	0,996
DSN 3	$\mathcal{L}_{BerHu}^+$	12 M	50 fps	0 – 50 m	0,078	0,312	2,596	0,122	11,143	0,035	0,927	0,986	0,996
DSN 3 + Pyramid ¹	$\mathcal{L}_{BerHu}$	12 M	39 fps	0 – 50 m	0,078	0,301	2,553	0,121	11,093	0,034	0,931	0,985	0,996
DSN 3 + Pyramid ¹ + Sem	$\mathcal{L}_{BerHu}$	12 M	39 fps	0 – 50 m	0,077	0,301	3,539	0,123	11,309	0,034	0,927	0,984	0,996
DSN 3 + Pyramid ¹ + CS	$\mathcal{L}_{BerHu}^+$	12 M	39 fps	0 – 50 m	0,074	0,302	2,437	0,116	10,579	0,032	0,937	0,988	0,996
DSN 3 + Pyramid ² + CS	$\mathcal{L}_{BerHu}^+$	12 M	32 fps	0 – 50 m	0,074	0,274	2,393	0,114	10,413	0,032	0,937	0,988	0,997

Tabela 9 – Avaliação da DSN com o módulo de normais de superfície no *dataset* KITTI Depth (UHRIG et al., 2017). A configuração DSN 3 + Pyramid¹ + CS + SN, bem como o detector de bordas Sobel e a função de custo $\mathcal{L}_{BerHu}^+$ é padrão para todos os experimentos. SN significa que o treinamento é realizado com o *ground truth* adicional de normais de superfície. A coluna Blur indica a presença ou ausência de uma convolução com kernel gaussiano. As constantes ψ_1 e ψ_2 formam a função de custo 2,5D e são empregadas nas etapas de pré-treinamento e treinamento respectivamente

Método	Tamanho	Velocidade	Blur	ψ_1	ψ_2	Cap	Abs Rel↓	Sqr Rel↓	RMSE↓	RMSE (log)↓	SILog↓	log10↓	$\delta < 1, 25^\uparrow$	$\delta < 1, 25^{2^\uparrow}$	$\delta < 1, 25^{3^\uparrow}$
DSN 3 + Pyramid ¹ + CS + SN	12 M	32 <i>fps</i>	✓	10 ⁶	10 ⁷	0 – 80 m	0,081	0,391	3,347	0,126	11,591	0,036	0,924	0,985	0,996
DSN 3 + Pyramid ¹ + CS + SN	12 M	32 <i>fps</i>	✓	10 ⁶	10 ⁶	0 – 80 m	0,075	0,365	3,264	0,120	10,947	0,033	0,934	0,986	0,996
DSN 3 + Pyramid ¹ + CS + SN	12 M	33 <i>fps</i>	✗	10 ⁶	10 ⁷	0 – 80 m	0,076	0,360	3,259	0,120	10,950	0,034	0,932	0,986	0,996
DSN 3 + Pyramid ¹ + CS + SN	12 M	33 <i>fps</i>	✗	10 ⁶	10 ⁶	0 – 80 m	0,075	0,363	3,253	0,119	10,886	0,033	0,934	0,986	0,996
DSN 3 + Pyramid ¹ + CS + SN	12 M	32 <i>fps</i>	✓	10 ⁶	10 ⁷	0 – 50 m	0,077	0,296	2,473	0,118	10,827	0,034	0,934	0,987	0,996
DSN 3 + Pyramid ¹ + CS + SN	12 M	32 <i>fps</i>	✓	10 ⁶	10 ⁶	0 – 50 m	0,072	0,268	2,355	0,111	10,113	0,031	0,944	0,989	0,997
DSN 3 + Pyramid ¹ + CS + SN	12 M	33 <i>fps</i>	✗	10 ⁶	10 ⁷	0 – 50 m	0,072	0,263	2,365	0,111	10,129	0,032	0,942	0,989	0,997
DSN 3 + Pyramid ¹ + CS + SN	12 M	33 <i>fps</i>	✗	10 ⁶	10 ⁶	0 – 50 m	0,071	0,267	2,351	0,111	10,053	0,031	0,944	0,989	0,997

Tabela 10 – Desempenho da FCN proposta e de outros trabalhos na literatura de SIDE de acordo com as métricas do KITTI Depth (UHRIG et al., 2017). Legenda: Cal. Cam. - câmera calibrada; CM - movimento da câmera; DM - mapa de disparidade; DSN † - configuração da DSN que produz os resultados gerais mais precisos; DSO - odometria esparsa direta; GAN - Rede Generativa Adversarial; GT - *ground truth*; MV - vídeo monocular; SGM - correspondência semi-global; SI - imagem única; SL - rótulos semânticos; SN - normais de superfície; SP - par estéreo; Synt. - sintético

Método	Entrada de Treinament/Info. Auxiliar	Cap	Abs Rel↓	Sqr Rel↓	RMSE↓	RMSE log↓	$\delta < 1,25^\dagger$	$\delta < 1,25^\dagger$	$\delta < 1,25^\dagger$
Saxena et al. (SAXENA; SUN; NG, 2008)	SI/GT de profundidade	0 – 80 m	0.280	3.012	8.734	0.361	0.601	0.820	0.926
Eigen et al. (EIGEN; PUHRSCH; FERUS, 2014)	SI/GT de profundidade	0 – 80 m	0.203	1.548	6.307	0.282	0.702	0.898	0.967
Liu et al. (LIU et al., 2015)	SI/GT de profundidade	0 – 80 m	0.201	1.584	6.471	0.273	0.680	0.898	0.967
Zhou et al. (ZHOU et al., 2017)	MV/-	0 – 80 m	0.198	1.836	6.565	0.275	0.718	0.901	0.960
UnDeepVO (LI et al., 2018a)	SP/Cal. Cam.	0 – 80 m	0.183	1.730	6.570	0.268	-	-	-
Yang et al. (YANG et al., 2017)	MV/-	0 – 80 m	0.182	1.481	6.501	0.267	0.725	0.906	0.963
AdaDepth (KUNDU et al., 2018)	SI e SI Synt./GT Synt. de profundidade (GAN)	0 – 80 m	0.167	1.257	5.578	0.237	0.771	0.922	0.971
Klodt et al. (KLODT; VEDALDI, 2018)	MV/-	0 – 80 m	0.166	1.490	5.988	-	0.778	0.919	0.966
Mahjourian et al. (MAHJOURIAN; WICKE; ANGELOVA, 2018)	MV/-	0 – 80 m	0.163	1.240	6.220	0.250	0.762	0.916	0.968
LEGO (YANG et al., 2018)	SP/-	0 – 80 m	0.162	1.352	6.276	0.252	-	-	-
Poggi et al. (POGGI et al., 2018)	SP/-	0 – 80 m	0.153	1.363	6.030	0.252	0.789	0.918	0.963
GeoNet (YIN; SHI, 2018)	MV/-	0 – 80 m	0.153	1.328	5.737	0.232	0.802	0.934	0.972
Pilzer et al. (PILZER et al., 2018)	SP/Cal. Cam.	0 – 80 m	0.152	1.388	6.016	0.247	0.789	0.918	0.965
DDVO (WANG et al., 2018)	MV/-	0 – 80 m	0.151	1.257	5.583	0.228	0.810	0.936	0.974
DF-Net (ZOU; LUO; HUANG, 2018)	MV/-	0 – 80 m	0.150	1.124	5.507	0.223	0.806	0.933	0.973
Ranjan et al. (RANJAN et al., 2019)	MV/-	0 – 80 m	0.148	1.149	5.464	0.226	0.815	0.935	0.973
MiniNet et al. (LIU et al., 2020)	MV/-	0 – 50 m	0.141	1.080	5.264	0.216	0.825	0.941	0.976
Struct2depth (CASSER et al., 2019a)	MV/Cal. Cam. (apenas a matriz intrínseca)	0 – 80 m	0.141	1.026	5.291	0.215	0.816	0.945	0.979
Elkerdawy et al. (ELKERDawy; ZHANG; RAY, 2019)	SP/Cal. Cam.	0 – 80 m	0.136	-	5.891	-	0.827	-	-
PHN et al. (ZHANG et al., 2018b)	SI/GT de profundidade	0 – 80 m	0.136	-	4.082	0.164	0.864	0.966	0.989
Zhan FullNYU (ZHAN et al., 2018)	SP/CM	0 – 80 m	0.135	1.132	5.585	0.229	0.820	0.933	0.971
Wong et al. (WONG; SOATTO, 2019)	SP/- (Cal. necessária no teste)	0 – 80 m	0.133	1.126	5.515	0.231	0.826	0.934	0.969
3Net (ResNet-50) (POGGI; TOSI; MATTOCCIA, 2018)	Setup Trinocular/-	0 – 80 m	0.129	0.996	5.281	0.223	0.831	0.939	0.974
StrAT (MEHTA; SAKURIKAR; NARAYANAN, 2018)	SI/Cal. Cam.	0 – 80 m	0.128	1.019	5.403	0.227	0.827	0.935	0.971
EPC++ (LUO et al., 2018a)	MV/Cal. Cam. (apenas a matriz intrínseca)	0 – 80 m	0.128	0.935	5.011	0.209	0.831	0.945	0.979
Gordon et al. (GORDON et al., 2019)	MV/-	0 – 80 m	0.124	0.930	5.120	0.206	0.851	0.950	0.978
Sparse-to-Continuous (ROSA; GUIZILINI; GRASSI, 2019)	SI/GT de profundidade	0 – 80 m	0.123	0.641	4.524	0.199	0.881	0.966	0.986
SA-Attention (ZHOU et al., 2019)	MV/Cal. Cam. (apenas a matriz intrínseca)	0 – 80 m	0.121	0.837	4.945	0.197	0.853	0.955	0.982
3Net (VGG) (POGGI; TOSI; MATTOCCIA, 2018)	Setup Trinocular/-	0 – 80 m	0.119	1.201	5.888	0.208	0.844	0.941	0.978
SceneNet (CHEN et al., 2019)	SI e SP/SL	0 – 80 m	0.118	0.905	5.096	0.211	0.839	0.945	0.977
Su et al. (SU et al., 2020)	SI/GT de profundidade	0 – 80 m	0.117	-	4.251	0.174	0.894	0.971	0.984
SDNet (OCHS; KRETZ; MESTER, 2019)	SI/GT de profundidade e SL	0 – 80 m	0.116	0.945	4.916	0.208	0.861	0.952	0.968
Cao et al. (CAO; WU; SHEN, 2017)	SI/GT de profundidade	0 – 80 m	0.115	0.912	4.712	0.198	0.887	0.963	0.982
Monodepth (GODARD; AODHA; BROSTOW, 2017)	SP/Cal. Cam.	0 – 80 m	0.114	0.898	4.935	0.206	0.861	0.949	0.976
LSIM (GOLDMAN; HASSNER; AVIDAN, 2019)	SP/Cal. Cam.	0 – 80 m	0.113	0.898	5.048	0.208	0.853	0.948	0.976
Kuznetsov et al. (KUZNETSOV; STUCKLER; LEIBE, 2017)	SP/Cal. Cam. e GT de profundidade	0 – 80 m	0.113	0.741	4.621	0.189	0.862	0.960	0.986
SuperDepth (PILLAI; AMBRUS; GAIDON, 2019)	SP/Cal. Cam.	0 – 80 m	0.112	0.875	4.958	0.207	0.852	0.947	0.977
DeepLabV3+ (F10) (GUR; WOLF, 2019)	SI/GT de profundidade e Cal. Cam.	0 – 80 m	0.110	0.666	4.186	0.168	0.880	0.966	0.988
Monodepth2 (GODARD et al., 2019)	SP ou MV ou ambos/Cal. Cam.	0 – 80 m	0.106	0.806	4.630	0.193	0.876	0.958	0.980
PackNet-SM (GUIZILINI et al., 2019)	MV/Termo de velocidade (opcional)	0 – 80 m	0.104	0.758	4.386	0.182	0.895	0.964	0.982
Guizilini et al. (GUIZILINI et al., 2020)	MV/Rede pré-treinada com SL	0 – 80 m	0.100	0.761	4.270	0.175	0.902	0.965	0.982
CFA (GURRAM et al., 2018)	SI/GT de profundidade e SL	0 – 80 m	0.100	0.601	4.298	0.174	0.874	0.966	0.989
RefineNet-Distill (PILZER et al., 2019)	SP/Cal. Cam.	0 – 80 m	0.098	0.831	4.656	0.202	0.882	0.948	0.973
DVSO (YANG et al., 2018b)	SP/GT de profundidade (DSO)	0 – 80 m	0.097	0.734	4.442	0.187	0.888	0.958	0.980
SOM (ZHU et al., 2019)	SI/GT de profundidade	0 – 80 m	0.097	0.398	3.007	0.133	0.913	0.985	0.997
Depth Hints (WATSON et al., 2019)	SP/Cal. Cam.	0 – 80 m	0.096	0.710	4.393	0.185	0.890	0.962	0.981
monoResMatch (TOSI et al., 2019)	SP/Proxy GT de profundidade com SGM	0 – 80 m	0.096	0.673	4.351	0.184	0.890	0.961	0.981
VGG16-UNet (GUO et al., 2018)	SI/SP-DM Synt. e Cal. Cam.	0 – 80 m	0.096	0.641	4.095	0.168	0.892	0.967	0.986
SemiDepth (AMIRI; LOO; ZHANG, 2019)	SP/Cal. Cam. e GT de profundidade	0 – 80 m	0.096	0.552	3.995	0.152	0.892	0.972	0.992
SVS (LUO et al., 2018b)	SP/Cal. Cam.	0 – 80 m	0.094	0.626	4.252	0.177	0.891	0.965	0.984
DenseDepth (ALHASHIM; WONKA, 2018)	SI/GT de profundidade	0 – 80 m	0.093	0.589	4.170	0.171	0.886	0.965	0.986
VOMonodepth (ANDRAGHETTI et al., 2019)	SP/Profundidades esparsas e Cal. Cam.	0 – 80 m	0.091	0.548	3.690	0.181	0.892	0.956	0.979
FAL-netB49 (GONZALEZBELLO; KIM, 2020)	SP/Cal. Cam.	0 – 80 m	0.088	0.547	4.004	0.175	0.898	0.966	0.984
Mendes et al. (2020)	SI/GT de profundidade	0 – 80 m	0.088	0.461	3.780	0.140	0.905	0.978	0.994
Monodepth2-Self (POGGI et al., 2020)	SP ou MV ou ambos/Cal. Cam.	0 – 80 m	0.083	-	3.682	-	0.919	-	-
BA-Net (TANG; TAN, 2018)	MV/Cal. Cam.	0 – 80 m	0.083	-	3.640	0.134	-	-	-
DSN†	SI/GT de profundidade e SN	0 – 80 m	0.075	0.363	3.253	0.119	0.934	0.986	0.996
VNL (YIN et al., 2019)	SI/GT de profundidade e VN	0 – 80 m	0.072	-	3.258	0.117	0.938	0.990	0.998
DORN (FU et al., 2018)	SI/GT de profundidade	0 – 80 m	0.072	0.307	2.727	0.120	0.932	0.984	0.994
BTS (LEE et al., 2019)	SI/GT de profundidade	0 – 80 m	0.059	0.245	2.756	0.096	0.956	0.993	0.998
Zhou et al. (ZHOU et al., 2017)	MV/-	0 – 50 m	0.190	1.436	4.975	0.258	0.735	0.915	0.968
Kim et al. (KIM et al., 2018)	SI/GT de profundidade	0 – 50 m	0.177	-	6.570	0.254	-	-	-
Garg et al. (GARG et al., 2016)	SP/CM	1 – 50 m	0.169	1.080	5.104	0.273	0.740	0.904	0.962
Mahjourian et al. (MAHJOURIAN; WICKE; ANGELOVA, 2018)	MV/-	0 – 50 m	0.155	0.927	4.549	0.231	0.781	0.931	0.975
GeoNet (YIN; SHI, 2018)	MV/-	0 – 50 m	0.147	0.936	4.348	0.218	0.810	0.941	0.977
Poggi et al. (POGGI et al., 2018)	SP/-	0 – 50 m	0.145	1.014	4.608	0.227	0.813	0.934	0.972
Pilzer et al. (PILZER et al., 2018)	SP/Cal. Cam.	0 – 50 m	0.144	1.007	4.660	0.240	0.793	0.923	0.968
MiniNet et al. (LIU et al., 2020)	MV/-	0 – 50 m	0.135	0.839	4.067	0.205	0.838	0.947	0.978
Wong et al. (WONG; SOATTO, 2019)	SP/- (Cal. necessária no teste)	0 – 50 m	0.126	1.832	4.172	0.217	0.840	0.941	0.973
Sparse-to-Continuous (ROSA; GUIZILINI; GRASSI, 2019)	SI/GT de profundidade	0 – 50 m	0.119	0.444	3.097	0.180	0.893	0.974	0.990
Monodepth (GODARD; AODHA; BROSTOW, 2017)	SP/Cal. Cam.	0 – 50 m	0.108	0.657	3.729	0.194	0.873	0.954	0.979
Kuznetsov et al. (KUZNETSOV; STUCKLER; LEIBE, 2017)	SP/Cal. Cam. e GT de profundidade	1 – 50 m	0.108	0.595	3.518	0.179	0.875	0.964	0.988
Cao et al. (CAO; WU; SHEN, 2017)	SI/GT de profundidade	0 – 50 m	0.107	-	3.605	0.187	0.898	0.966	0.984
Su et al. (SU et al., 2020)	SI/GT de profundidade	0 – 50 m	0.107	-	3.440	0.172	0.900	0.976	0.990
LSIM et al. (GOLDMAN; HASSNER; AVIDAN, 2019)	SP/Cal. Cam.	0 – 50 m	0.106	0.653	3.790	0.195	0.867	0.954	0.979
CFA (GURRAM et al., 2018)	SI/GT de profundidade e SL	0 – 50 m	0.096	0.482	3.338	0.166	0.886	0.980	0.995
Gan et al. (GAN et al., 2018)	SI/GT de profundidade	0 – 50 m	0.094	0.552	3.133	0.165	0.898	0.967	0.986
Mendes et al. (2020)	SI/GT de profundidade	0 – 50 m	0.083	0.335	2.701	0.129	0.919	0.983	0.996
DSN†	SI/GT de profundidade e SN	0 – 50 m	0.071	0.267	2.351	0.111	0.944	0.989	0.997
DORN (FU et al., 2018)	SI/GT de profundidade	0 – 50 m	0.071	0.268	2.271	0.116	0.936	0.985	0.995
BTS (LEE et al., 2019)	SI/GT de profundidade	0 – 50 m	0.056	0.169	1.925	0.087	0.964	0.994	0.999

Figura 42 – Comparação qualitativa entre as predições da melhor abordagem desenvolvida (DSN †) e de métodos auto-supervisionados recentes que abordam a tarefa de SIDE. As imagens de entrada e o *ground truth* pertencem ao conjunto de dados KITTI Depth (UHRIG et al., 2017). Os mapas de profundidade de referência são interpolados para fins de visualização. Devido a discrepância das paletas de cores utilizadas, a comparação entre as imagens deve ser realizada considerando-se a geometria e o *layout* das cenas.

Fonte: adaptada de (MENDES et al., 2021)

oclusões.

Nos mapas de profundidade da Fig. 42f, nota-se uma maior quantidade de objetos inferidos, principalmente nas regiões mais distantes da cena, as quais são mais difíceis de serem compreendidas pelos outros métodos da literatura de SIDE. Como as paletas de cores são desiguais para os mapas da Fig. 42, a comparação qualitativa dessas predições deve abordar os aspectos geométricos e a disposição das cenas.

Por outro lado, a Fig. 43 compara as predições da DSN com aquelas produzidas por *frameworks* supervisionados recentes. Como pode ser visto em tal figura, a DSN produz mapas de profundidade de alta resolução com menos regiões desfocadas sob o limite do céu. Por meio da Fig. 44, é possível verificar que as nuvens de pontos 3D reconstruídas a partir das predições da abordagem proposta têm uma distribuição mais densa do que as nuvens de pontos criadas com o *ground truth* esparsa e ruidoso. Os mapas 3D da Fig. 44 também mostram que a FCN desenvolvida é robusta ao ruído dos dados de entrada e é capaz de modelar a forma dos objetos com poucos erros.

Figura 43 – Diferenças entre os mapas de profundidade retornados pela melhor configuração da DSN (DSN †) e por abordagens de SIDE no estado-da-arte com *pipelines* de treinamento supervisionados. As imagens RGB de entrada compõem o *dataset* KITTI Depth (UHRIG et al., 2017)

Fonte: adaptada de (MENDES et al., 2021)

Figura 44 – Análise qualitativa das nuvens de pontos reconstruídas a partir das previsões da DSN. DSN † indica a configuração da DSN que produz às métricas gerais mais precisas. A entrada RGB e as nuvens de pontos de referência com dados de profundidade são retiradas do conjunto de dados KITTI Depth (UHRIG et al., 2017). No entanto, as informações das normais da superfície de referência são geradas com o auxílio do algoritmo introduzido por Fouhey, Gupta e Hebert (2013)

Fonte: adaptada de (MENDES et al., 2021)

5.1.4 Estudos sobre densificação de profundidade

Com o intuito de avaliar o desempenho da DSN em aplicações de densificação de profundidade, empregou-se a técnica de amostragem com base na distribuição categórica da Eq. 4.1 para alimentar a CNN com mapas RGBD. Os resultados de tais experimentos,

Tabela 11 – Avaliação do *pipeline* de densificação de profundidade de acordo com Eigen Split aplicado ao conjunto de dados do KITTI Depth (UHRIG et al., 2017). A configuração DSN 3 + Pyramid¹ + CS, além da função $\mathcal{L}_{BerHu}^+$, é considerada como padrão para os primeiros experimentos. A configuração DSN 3 + Pyramid¹ + CS + SN, juntamente com o detector de borda, é analisada posteriormente

Método	Cap	Amostras	Abs Rel↓	RMSE↓	log10↓	$\delta < 1, 25^\uparrow$	$\delta < 1, 25^2^\uparrow$	$\delta < 1, 25^3^\uparrow$
DSN 3 + Pyramid ¹ + CS	80	20	0,049	2,829	0,021	0,961	0,991	0,997
DSN 3 + Pyramid ¹ + CS	80	50	0,035	2,397	0,015	0,976	0,994	0,998
DSN 3 + Pyramid ¹ + CS	80	100	0,029	2,184	0,013	0,981	0,995	0,998
DSN 3 + Pyramid ¹ + CS	80	200	0,024	1,926	0,010	0,987	0,996	0,999
DSN 3 + Pyramid ¹ + CS	80	500	0,019	1,672	0,008	0,991	0,997	0,999
DSN 3 + Pyramid ¹ + CS + SN	80	200	0,024	1,863	0,010	0,987	0,996	0,999
DSN 3 + Pyramid ¹ + CS + SN	80	500	0,019	1,588	0,008	0,991	0,997	0,999
DSN 3 + Pyramid ¹ + CS	50	20	0,045	1,922	0,019	0,970	0,993	0,998
DSN 3 + Pyramid ¹ + CS	50	50	0,032	1,595	0,014	0,982	0,995	0,999
DSN 3 + Pyramid ¹ + CS	50	100	0,026	1,432	0,011	0,986	0,996	0,999
DSN 3 + Pyramid ¹ + CS	50	200	0,021	1,252	0,009	0,990	0,997	0,999
DSN 3 + Pyramid ¹ + CS	50	500	0,017	1,075	0,007	0,993	0,998	0,999
DSN 3 + Pyramid ¹ + CS + SN	50	200	0,022	1,247	0,09	0,990	0,997	0,999
DSN 3 + Pyramid ¹ + CS + SN	50	500	0,018	1,051	0,008	0,993	0,998	0,999

expostos na Tabela 11, reforçam a teoria de que quanto mais pontos de LiDAR são apresentados à entrada da rede, mais precisos são os mapas de profundidade retornados.

Isso pode ser visualizado nos gráficos da Fig. 45, em que as métricas de erro diminuem e as métricas de precisão aumentam à medida que mais dados de profundidade esparsos são adicionados à entrada da rede. No entanto, pelos perfis dos gráficos da Fig. 45, infere-se que os valores dos erros e precisões atingem pontos de saturação, em que não é possível melhorar os resultados significativamente, ainda que mais amostras de LiDAR sejam alimentadas na rede.

Com base nas métricas da Tabela 11, pode-se concluir que as pistas no espaço 3D produzidas pelos mapas de referência de normais de superfície causam uma influência positiva no processo de densificação. Além disso, o melhor desempenho do *pipeline* de densificação de profundidade proposto é comparado aos resultados de outras abordagens na literatura através da Tabela 12.

Embora a DSN seja capaz de densificar mapas de profundidade em uma alta taxa de *frames*, ela também gera mapas densos com uma qualidade que supera a de outros trabalhos no estado-da-arte. Os resultados de densificação de profundidade mais precisos são expostos na Fig. 46, em que ambos os experimentos envolvendo 200 e 500 dados esparsos de profundidade, alimentados na DSN, geram mapas de profundidade densos de alta qualidade.

A partir dos gráficos da Fig. 47, analisou-se os efeitos causados na DSN quando ela é alimentada com dados esparsos de profundidade gerados pela Monodepth2 pré-treinada na configuração mono + estéreo. O comportamento da curva relacionada à abordagem Monodepth2 cresce para a métrica Abs Rel e diminui para a $\delta < 1, 25$ [%], pois quanto mais

Figura 45 – Evolução dos erros e da precisão das predições da DSN de acordo com o número de amostras esparsas de entrada. Utilizou-se o *ground truth* do *dataset* KITTI Depth (UHRIG et al., 2017) para apresentar os resultados

Fonte: adaptada de (MENDES et al., 2021)

Tabela 12 – Comparação quantitativa dos resultados do *framework* proposto e dos resultados de outras técnicas de densificação de profundidade. As métricas são calculadas considerando o Eigen Split e o conjunto de dados KITTI Depth (UHRIG et al., 2017). Aplicou-se o limite de 80 *m* em todas as análises

Método	Amostras	Abs Rel↓	RMSE↓	$\delta < 1, 25\uparrow$	$\delta < 1, 25^2\uparrow$	$\delta < 1, 25^3\uparrow$
full-MAE (CADENA; DICK; REID, 2016)	~650	0,179	7,14	0,709	0,888	0,956
Liao et al. (LIAO et al., 2017)	225	0,113	4,50	0,874	0,960	0,984
Ma et al. (MA; KARAMAN, 2018)	200	0,083	3,851	0,919	0,970	0,986
Ma et al. (MA; KARAMAN, 2018)	500	0,073	3,378	0,935	0,976	0,989
MC + SPN (CHENG; WANG; YANG, 2018)	500	0,059	3,248	0,944	0,977	0,989
SPN (LIU et al., 2017)	500	0,063	3,243	0,943	0,978	0,991
Mirror Connection (MC) (CHENG; WANG; YANG, 2018)	500	0,051	3,049	0,953	0,979	0,990
CSPN (CHENG; WANG; YANG, 2018)	500	0,049	3,029	0,955	0,980	0,990
MC + CSPN (CHENG; WANG; YANG, 2018)	500	0,044	2,977	0,957	0,980	0,991
MC + CSPN + ACSPF (CHENG; WANG; YANG, 2018)	500	0,042	2,843	0,961	0,982	0,992
DSN 3 + Pyramid ¹ + CS + SN	200	0,024	1,863	0,987	0,996	0,999
DSN 3 + Pyramid ¹ + CS + SN	500	0,019	1,588	0,991	0,997	0,999

dados esparsos são amostrados dos mapas de profundidade produzidos pela Monodepth2, mais os erros desses mapas influenciam os resultados da DSN.

No entanto, os gráficos da Fig. 47 também indicam que o desempenho da FCN proposta (sem o módulo de normais de superfície) é melhorado utilizando menos quantidades de dados de profundidade amostrados a partir dos resultados da Monodepth2. Portanto, a DSN pode ser efetivamente aplicada em situações reais, onde não há informações do

Figura 46 – Avaliação qualitativa do *framework* de densificação de profundidade com base no *ground truth* ruidoso e esparsos do *dataset* KITTI Depth (UHRIG et al., 2017). Legenda: Entrada - imagem RGB de entrada; GT - *ground truth*; d - dados de entrada esparsos

Fonte: adaptada de (MENDES et al., 2021)

ground truth disponíveis.

Figura 47 – Comparação entre os erros e a precisão obtidos pelo *framework* de densificação de profundidade (sem o módulo de normais de superfície), quando ele é apresentado a dados esparsos do *ground truth* e aos mapas de profundidade produzidos pela Monodepth2 (GODARD et al., 2019)

Fonte: adaptada de (MENDES et al., 2021)

5.1.5 Resultados de odometria visual

Uma vez que os dados de profundidade são estimados, eles podem ser incorporados a sistemas de VO/SLAM para aperfeiçoar a qualidade da odometria visual e do mapa reconstruído. Com relação aos métodos de Odometria Visual Semi-Direta (SVO), as informações precisas de profundidade obtidas a partir de abordagens de SIDE podem ser adicionadas para aumentar a invariância do algoritmo de *tracking* a mudanças de iluminação nas imagens superexpostas ou subexpostas de entrada (LOO et al., 2019). Ademais, com o auxílio de tais pistas adicionais, a etapa de mapeamento de pontos da estrutura SVO é realizada com uma incerteza menor, levando a um aumento na precisão da correspondência de características e a uma diminuição no tempo de convergência desse mecanismo de correspondência (LOO et al., 2019).

CNN-SLAM (TATENO et al., 2017) e CNN-SVO (LOO et al., 2019) são exemplos de trabalhos, nos quais a incorporação dos mapas de profundidade, retornados por uma rede neural profunda, no *pipeline* de mapeamento é realizada de maneira benéfica. Mais especificamente, o primeiro trabalho utilizou uma FCN (LAINA et al., 2016) baseada em ResNet, enquanto o último implementou a Monodepth (GODARD; AODHA; BROSTOW, 2017) para gerar as estimações de profundidade. Sabendo-se que as predições da DSN são mais precisas do que aquelas produzidas pelos métodos citados, alimentou-se a estrutura CNN-SVO com os mapas de profundidade gerados pela DSN a fim de melhor determinar as características correspondentes para as estimações de pose na etapa de localização. Usando diferentes configurações da CNN proposta, avaliou-se os avanços que essa modificação no mapeamento de pontos de quadros-chave consecutivos traz para a tarefa de SVO.

Particularmente, foram comparadas as métricas obtidas pela aplicação conjunta da CNN-SVO e variações da DSN com: a CNN-SVO sozinha, a Odometria Esparsa Direta (DSO) (ENGEL; KOLTUN; CREMERS, 2017), a SVO (FORSTER; PIZZOLI; SCARAMUZZA, 2014) e a ORB-SLAM sem fechamento de laço (MUR-ARTAL; TARDÓS, 2017) através da Tabela 13. Tais experimentos são conduzidos com o auxílio das 11 sequências com trajetórias de referência fornecidas pelo *dataset* KITTI Odometry (GEIGER; LENZ; URTASUN, 2012). O Erro Absoluto de Trajetória (ATE) e a métrica RMSE foram aplicados para expor os resultados mencionados. Na Tabela 13, os espaços não preenchidos com valores em metro estão relacionados às técnicas que não são capazes de computar toda a sequência devido a falhas no algoritmo de *tracking*. Conforme proposto no trabalho de Loo et al. (2019), os ATEs foram armazenados com uma mediana de 5 rodadas.

Analisando as métricas da Tabela 13, conclui-se que quando a técnica CNN-SVO, juntamente com o algoritmo de *Bundle Adjustment* (BA), emprega os mapas de profundidade retornados pela DSN, ela se torna capaz de rastrear a maioria das sequências abordadas de forma mais precisa. Isso ocorre devido à inicialização de novos pontos do mapa no *pipeline* da CNN-SVO com incertezas de profundidade menores, as quais

Tabela 13 – Avaliação das predições da DSN, aplicadas no *framework* da CNN-SVO, de acordo com a métrica RMSE. Foram utilizadas as primeiras 11 seqüências do *dataset* KITTI Odometry (GEIGER; LENZ; URTASUN, 2012) para gerar os resultados. Legenda: BA - Bundle Adjustment; DSN † - configuração DSN, sem o módulo de normais de superfície, que produz os melhores resultados; DSN ‡ - configuração de DSN, com o módulo de normais de superfície, que produz os resultados mais precisos; DSO - Odometria Esparsa Direta; SVO - Odometria Visual Semi-Direta

Seqüência	SVO	CNN-SVO + BA	CNN-SVO + BA + DSN‡	CNN-SVO + BA + DSN†	CNN-SVO + DSN†	DSO	ORB-SLAM sem fechamento de laço
00	-	17,5269	16,9438	19,7020	42,9432	113,1838	77,9502
01	-	-	-	-	-	-	-
02	-	50,5119	14,9513	15,7180	45,0136	116,8108	41,0064
03	-	3,4588	1,9309	3,4340	12,4293	1,3943	1,0182
04	58,3970	2,4414	2,3090	4,1912	5,6721	0,422	0,9302
05	-	8,1513	7,1224	9,4541	30,2511	47,4605	40,3542
06	-	11,5091	8,7421	11,4059	22,5910	55,6173	52,2282
07	-	6,5141	1,9292	2,3099	8,0324	16,7192	16,546
08	-	10,9755	8,3792	9,4699	13,3337	111,0832	51,6215
09	-	10,6873	10,9941	15,0341	28,6044	52,2251	58,1742
10	-	4,8354	2,3163	2,8371	4,1040	11,090	18,4765

possibilitam que o mapa convirja para as profundidades reais, de maneira mais rápida, por meio de algoritmos de *depth-filters*. Os resultados da Tabela 13 também indicam que quando o CNN-SVO é combinada com a configuração da DSN que explora pistas de normais de superfície, ela consegue fornecer os melhores valores gerais de métricas.

As trajetórias ilustradas na Fig. 48 representam os resultados qualitativos alinhados da abordagem de odometria visual mais precisa nas sequências do KITTI Odometry. Tais trajetórias confirmam as métricas expostas na Tabela 13. Na Fig. 48, aplicou-se a técnica de alinhamento de trajetória de Horn (forma fechada) (SALAS et al., 2015) para apresentar os resultados.

Os valores de *scale drift* da Fig. 49 mostram que os resultados de localização obtidos têm uma escala próxima do *ground truth*. No entanto, as defasagens de escala ocorrem devido às distorções sofridas pelos mapas de profundidade de saída da DSN, os quais são redimensionados para serem aplicados na CNN-SVO.

5.2 Estudos de ablação em ambientes internos

5.2.1 Estudos estruturais

Com o intuito de analisar o comportamento da DSN 3 quando exposta a outros tipos de cenas, foram realizados estudos de ablação com os conjuntos de dados de ambientes internos, os quais são compostos por imagens com escalas diferentes das presentes nos *datasets outdoor*. Inicialmente, foram modificadas a função de custo e a estrutura do decodificador da FCN proposta a fim de determinar uma combinação otimizada.

Em relação à métrica RMSE, a Tabela 14 indica que tanto o termo de atenção quanto a modificação na constante κ da função BerHu não trazem benefícios para as predições da rede, pois, em comparação com o KITTI Depth (UHRIG et al., 2017), o *dataset* NYU Depth V2 (SILBERMAN et al., 2012) é mais denso em regiões mais próximas e mais distantes da cena. Devido ao perfil mais denso do *ground truth* dos conjuntos de dados de ambientes internos, funções de custo simples, como a $\mathcal{L}_1^+$ e a $\mathcal{L}_2^+$, são capazes de lidar com a distribuição de pixels dos mapas de referência e atualizar os pesos da rede de forma mais rápida e eficiente.

Como também pode ser inferido da Tabela 14, o módulo Pyramid¹ melhora o valor da maioria das métricas. No entanto, o bloco Pyramid² não é vantajoso para as predições da DSN 3, uma vez que, considerando o NYU Depth V2 (SILBERMAN et al., 2012), os mapas de características aumentados da convolução inicial do decodificador da rede não apresentam as pistas monoculares necessárias para aperfeiçoar a estimação do mapa de profundidade de saída.

Ademais, foram utilizados os mapas semânticos dos conjuntos de dados NYU Depth V2 (SILBERMAN et al., 2012) e SUNRGBD (SONG; LICHTENBERG; XIAO, 2015), bem

Figura 48 – Resultados qualitativos das estimações de trajetória da câmera obtidas nas sequências 00, 02, 03, 05, 07, 08 e 10 do KITTI Odometry (GEIGER; LENZ; URTASUN, 2012) pela CNN-SVO + BA em conjunto com a DSN e o módulo de normais de superfície. As trajetórias pretas pontilhadas e azuis sólidas referem-se ao *ground truth* e as predições respectivamente

Fonte: adaptada de (MENDES et al., 2021)

como as informações de profundidade de todos os *datasets indoor* cobertos neste trabalho, para pré-treinar a DSN. Assim como nos experimentos que abordam dados de ambientes externos, é possível inferir que os pesos pré-treinados para segmentação semântica ajudam a DSN a raciocinar sobre as regiões de alta frequência da imagem, porém as características aprendidas com o pré-treinamento da rede na mesma tarefa de SIDE leva a resultados mais precisos.

5.2.2 Análises de normais de superfície

Considerando as cenas de ambientes internos, os resultados obtidos pelos experimentos envolvendo a abordagem proposta, que é treinada com mapas de referência de profundidade e de normais de superfície, são apresentados na Tabela 15. A partir da

Tabela 14 – Desempenho do método de SIDE proposto para diferentes configurações de arquitetura e estratégias de treinamento. A rede de extração de características da DSN 3 é o Modelo 3 desenvolvido. Ambos os módulos da pirâmide, além da função de custo $\mathcal{L}_{BerHu}^+$, são implementados como nas análises envolvendo os conjuntos de dados de ambientes externos. Os termos Sem e Indoor indicam o pré-treinamento da DSN 3 com mapas semânticos e mapas de profundidade extra, respectivamente, fornecidos pelos *datasets* de ambientes internos

Método	Custo	Tamanho	Velocidade	Abs Rel↓	Sqr Rel↓	RMSE↓	RMSE (log)↓	SILog↓	log10↓	$\delta < 1,25^\uparrow$	$\delta < 1,25^2^\uparrow$	$\delta < 1,25^3^\uparrow$
DSN 3	$\mathcal{L}_{BerHu}$	12 M	51 <i>fps</i>	0,169	0,136	0,523	0,208	17,143	0,070	0,768	0,934	0,978
DSN 3	$\mathcal{L}_a + \mathcal{L}_{BerHu}$	12 M	51 <i>fps</i>	0,179	0,144	0,540	0,216	17,315	0,073	0,746	0,926	0,977
DSN 3	$\mathcal{L}_{BerHu}^+$	12 M	51 <i>fps</i>	0,179	0,151	0,538	0,215	17,277	0,073	0,750	0,924	0,976
DSN 3	$\mathcal{L}_1$	12 M	51 <i>fps</i>	0,180	0,151	0,548	0,217	17,588	0,073	0,746	0,927	0,978
DSN 3 + Pyramid ¹	$\mathcal{L}_{BerHu}$	12 M	41 <i>fps</i>	0,171	0,133	0,521	0,207	16,605	0,070	0,758	0,936	0,982
DSN 3 + Pyramid ¹	$\mathcal{L}_2$	12 M	41 <i>fps</i>	0,175	0,138	0,520	0,207	16,462	0,071	0,757	0,934	0,981
DSN 3 + Pyramid ¹	$\mathcal{L}_1^+$	12 M	41 <i>fps</i>	0,167	0,130	0,519	0,207	16,859	0,069	0,763	0,933	0,981
DSN 3 + Pyramid ¹	$\mathcal{L}_1^+$	12 M	32 <i>fps</i>	0,174	0,143	0,529	0,210	16,896	0,071	0,756	0,933	0,979
DSN 3 + Pyramid ²	$\mathcal{L}_1^+$	12 M	32 <i>fps</i>	0,174	0,139	0,522	0,211	17,231	0,071	0,760	0,932	0,977
DSN 3 + Pyramid ¹ + Sem	$\mathcal{L}_1^+$	12 M	41 <i>fps</i>	0,142	0,098	0,445	0,177	14,197	0,059	0,823	0,954	0,987
DSN 3 + Pyramid ¹ + Indoor	$\mathcal{L}_1^+$	12 M	41 <i>fps</i>	0,142	0,098	0,445	0,177	14,197	0,059	0,823	0,954	0,987

Figura 49 – Perfil de *scale drift* dos resultados de *tracking* gerados pela CNN-SVO + BA + DSN (com o módulo de normais de superfície). As sequências são retiradas do conjunto de dados KITTI Odometry (GEIGER; LENZ; URTASUN, 2012)

Fonte: adaptada de (MENDES et al., 2021)

métrica RMSE, é possível constatar que a aplicação do detector de bordas, implementado no módulo de normais de superfície, melhora os resultados, uma vez que ele dá suporte para a CNN desenvolvida no processo de predição de contornos mais precisos. A operação de detecção de borda reduz a velocidade de predição da rede em 1 *fps* e não altera o número de parâmetros treináveis.

Quando a DSN 3 é pré-treinada com os *ground truths* de normais de superfície adicionais, por meio da penalidade geométrica 2, 5D, os resultados também são aprimorados. Isso ocorre, pois o método baseado em aprendizagem é apresentado a uma maior quantidade de características geométricas 3D, as quais o ajudam no processo de generalização. Embora a presença do filtro Gaussiano leve a valores mais precisos para as métricas Sqr Rel e $\delta < 1,25^3$, sua ausência está relacionada à configuração do *framework* proposto que produziu os melhores resultados gerais. Isso prova a eficiência e robustez do módulo de normais de superfície, o qual pode ser aplicado a DSN de modo a manter uma alta taxa de *frames*.

Comparando os resultados mais precisos da DSN, no *dataset* NYU Depth V2 (SILBERMAN et al., 2012), com as métricas obtidas por diferentes trabalhos no estado-da-arte, a Tabela 16 mostra que o método proposto atinge o top-3 do ranking de referência. No entanto, entre as abordagens no top-3, a DSN possui os menores tempo de execução e

Tabela 15 – Análise do *pipeline* que explora pistas de normais de superfície. A configuração DSN 3 + Pyramid¹ + Indoor + SN, juntamente com a função de custo $\mathcal{L}_1^+$, é mantida em todos os experimentos. Legenda: Blur - convolução com kernel gaussiano; Borda - detector de bordas Sobel; DSN 3 - DSN com o Modelo 3 como *encoder*; Indoor - pré-treinamento com kernel gaussiano; ψ_1 e ψ_2 - constantes da função de custo 2, 5D utilizada nas etapas de pré-treinamento e treinamento, respectivamente; Pyramid¹ - módulo do *decoder* em formato de pirâmide; SN - *ground truth* de normais de superfície

Método	Tamanho	Velocidade	Borda	Blur	ψ_1	ψ_2	Abs Rel	Sqr Rel	RMSE	RMSE (log)	SILog	log10	$\delta < 1, 25$	$\delta < 1, 25^2$	$\delta < 1, 25^3$
DSN 3 + Pyramid ¹ + Indoor + SN	12 M	35 fps	✗	✗	✗	10 ⁰	0,138	0,096	0,444	0,177	14,227	0,058	0,825	0,954	0,986
DSN 3 + Pyramid ¹ + Indoor + SN	12 M	35 fps	✗	✗	✗	10 ⁶	0,137	0,096	0,441	0,176	14,171	0,058	0,825	0,954	0,986
DSN 3 + Pyramid ¹ + Indoor + SN	12 M	34 fps	✓	✗	✗	10 ⁶	0,137	0,094	0,439	0,175	14,134	0,058	0,826	0,954	0,986
DSN 3 + Pyramid ¹ + Indoor + SN	12 M	34 fps	✓	✗	✗	10 ⁷	0,138	0,095	0,442	0,176	14,171	0,058	0,825	0,956	0,986
DSN 3 + Pyramid ¹ + Indoor + SN	12 M	34 fps	✓	✗	✗	10 ⁵	0,139	0,096	0,443	0,177	14,289	0,059	0,822	0,953	0,986
DSN 3 + Pyramid ¹ + Indoor + SN	12 M	34 fps	✓	✗	✗	10 ⁶	0,134	0,091	0,435	0,173	13,917	0,057	0,831	0,958	0,986
DSN 3 + Pyramid ¹ + Indoor + SN	12 M	34 fps	✓	✗	✗	10 ⁷	0,134	0,089	0,434	0,172	13,841	0,057	0,831	0,957	0,987
DSN 3 + Pyramid ¹ + Indoor + SN	12 M	34 fps	✓	✗	✗	10 ⁶	0,132	0,088	0,429	0,171	13,732	0,056	0,834	0,959	0,987
DSN 3 + Pyramid ¹ + Indoor + SN	12 M	33 fps	✓	✓	10 ⁶	10 ⁷	0,132	0,086	0,431	0,171	13,793	0,056	0,833	0,959	0,988

Tabela 16 – Comparação quantitativa entre o método desenvolvido e outros que abordam a tarefa de estimação monocular de profundidade. Os resultados são avaliados no *dataset* NYU Depth V2 (SILBERMAN et al., 2012). DSN † refere-se à configuração da DSN que fornece os resultados gerais mais precisos

Método	Abs Rel↓	RMSE↓	log10↓	$\delta < 1, 25^\uparrow$	$\delta < 1, 25^{2^\uparrow}$	$\delta < 1, 25^{3^\uparrow}$
Saxena et al. (SAXENA; SUN; NG, 2008)	0,349	1,214	-	0,447	0,745	0,897
Karsch et al. (KARSCH; LIU; KANG, 2014)	0,349	1,210	1,131	-	-	-
Liu et al. (LIU; SALZMANN; HE, 2014)	0,335	1,060	1,127	-	-	-
Ladicky et al. (LADICKY; SHI; POLLEFEYS, 2014)	-	-	-	0,542	0,829	0,941
Eigen et al. (EIGEN; PUHRSCHE; FERGUS, 2014)	0,214	0,877	0,285	0,611	0,887	0,971
Wang et al. (WANG et al., 2015)	0,220	0,824	-	0,605	0,890	0,970
HCRF (LI et al., 2015)	0,232	0,821	0,094	0,621	0,886	0,968
DCNF (LIU et al., 2015)	0,213	0,759	0,087	0,650	0,906	0,976
NR forest (ROY; TODOROVIC, 2016)	0,187	0,744	0,078	-	-	-
MS-CNN (EIGEN; FERGUS, 2015)	0,158	0,641	-	0,769	0,950	0,988
Chakrabarti et al. (CHAKRABARTI; SHAO; SHAKHAROVICH, 2016)	0,149	0,620	-	0,806	0,958	0,987
Li et al. (LI; KLEIN; YAO, 2017)	0,152	0,611	0,064	0,789	0,955	0,988
SOM (ZHU et al., 2019)	0,136	0,604	0,067	0,814	0,959	0,990
Xu et al. (XU et al., 2018)	0,125	0,593	-	0,806	0,952	0,986
MS-CRF (XU et al., 2017)	0,121	0,586	0,052	0,811	0,954	0,987
PAD-Net (XU et al., 2018)	0,120	0,582	-	0,817	0,954	0,987
DeepLabV3+ (F10) (GUR; WOLF, 2019)	0,162	0,575	-	0,772	0,942	0,984
FCRN (LAINA et al., 2016)	0,127	0,573	0,055	0,811	0,953	0,988
Lee et al. (LEE et al., 2018)	0,139	0,572	-	0,815	0,963	0,991
Qi et al. (QI et al., 2018)	0,128	0,569	0,057	0,834	0,960	0,990
Index Network (LU et al., 2019)	-	0,565	-	0,786	-	-
RF-LW (NEKRASOV et al., 2019)	0,149	0,565	0,105	0,790	0,955	0,990
Cao et al. (CAO; WU; SHEN, 2017)	0,141	0,540	0,060	0,790	0,955	0,990
RelativeDepth (LEE; KIM, 2019)	0,131	0,538	0,087	0,837	0,971	0,994
SC-SfMLearner-ResNet18 (BIAN et al., 2020)	0,147	0,536	0,062	0,804	0,950	0,986
SENet-154 (HU et al., 2019)	0,115	0,530	0,050	0,866	0,975	0,993
Kim et al. (KIM et al., 2018)	0,117	0,525	-	0,825	0,976	0,993
SARPN (CHEN; CHEN; ZHA, 2019)	0,111	0,514	0,048	0,846	0,968	0,994
DORN (FU et al., 2018)	0,115	0,509	0,051	0,828	0,965	0,992
AdaDepth (KUNDU et al., 2018)	0,114	0,506	-	0,856	0,966	0,991
TRL (ZHANG et al., 2018)	0,144	0,501	0,181	0,815	0,962	0,992
Zhang et al. (ZHANG et al., 2018)	0,144	0,501	0,181	0,815	0,962	0,992
PHN et al. (ZHANG et al., 2018b)	0,144	0,501	-	0,835	0,962	0,992
Su et al. (SU et al., 2020)	0,137	0,498	0,058	0,826	0,967	0,995
SDC-Depth (WANG et al., 2020)	0,128	0,497	0,174	0,845	0,966	0,990
PAP (ZHANG et al., 2019)	0,121	0,497	0,175	0,846	0,968	0,994
ACAN (CHEN; ZHAO; HU, 2019)	0,138	0,496	0,101	0,826	0,964	0,990
SharpNet (RAMAMONJISOA; LEPETIT, 2019)	0,139	0,495	0,047	0,888	0,979	0,995
DenseDepth (ALHASHIM; WONKA, 2018)	0,123	0,465	0,053	0,846	0,974	0,994
DSN†	0,132	0,429	0,056	0,834	0,959	0,987
VNL (YIN et al., 2019)	0,108	0,416	0,048	0,875	0,976	0,994
BTS (LEE et al., 2019)	0,110	0,392	0,047	0,885	0,978	0,994

tamanho total, conforme exposto na Tabela 1.

Ademais, foram avaliadas as predições do módulo de normais de superfície da DSN, cujas métricas resultantes foram comparadas aquelas adquiridas por trabalhos recentes através da Tabela 17. Desconsiderando os pixels inválidos dos mapas de normais de superfície de referência, alcançou-se os resultados mais precisos apresentados na Tabela 17.

Por outro lado, considerando todos os pixels do *ground truth* de normais de superfície, ainda obtêm-se os melhores resultados da Mediana e das métricas de precisão. Portanto, isso demonstra que o módulo de normais de superfície introduzido é mais vantajoso que as outras técnicas, dado que, além de não consistir em um ramo inteiro ou outro modelo de CNN, ele trabalha mutuamente com a etapa de estimação monocular de profundidade.

Por meio das cenas de ambientes internos ilustradas nas Figs. 50, 51 e 52, comparou-se visualmente as predições de profundidade e de normais de superfície mais precisas da

Tabela 17 – Desempenho da CNN desenvolvida, bem como de outras técnicas, no *ground truth* do NYU Depth V2 (SILBERMAN et al., 2012). Legenda: DSN † - configuração da DSN que fornece os resultados gerais mais precisos; DSN ‡ - mesma configuração da DSN †, porém com o filtro Gaussiano; Mask - desconsideração dos pixels inválidos relacionados ao *ground truth* de normais de superfície

Método	Média↓	Mediana↓	RMSE↓	11, 25°↑	22, 50°↑	30°↑
Fouhey et al. (FOUHEY; GUPTA; HEBERT, 2013)	36,3	19,2	-	16,4	36,6	48,2
UIOW (FOUHEY; GUPTA; HEBERT, 2014)	35,2	17,9	-	40,5	54,1	58,9
Ladicky et al. (LADICKÝ; ZEISL; POLLEFEYS, 2014)	33,5	23,1	-	27,7	49,0	58,7
MS-CNN (EIGEN; FERGUS, 2015)	23,7	15,5	-	39,2	62,0	71,1
Deep3D (WANG; FOUHEY; GUPTA, 2015)	26,9	14,8	-	42,0	61,2	68,2
SkipNet (BANSAL; RUSSELL; GUPTA, 2016)	19,8	12,0	28,2	47,9	70,0	77,8
SURGE (WANG et al., 2016)	20,6	12,2	-	47,3	68,9	76,6
Qi et al. (QI et al., 2018)	19,0	11,8	26,9	48,4	71,5	79,5
PAP (ZHANG et al., 2019)	18,6	11,7	25,5	48,8	72,2	79,8
DSN† + Mask	8,6	5,5	12,2	72,2	91,0	96,8
DSN†	18,7	6,9	33,7	63,2	79,7	84,8
DSN‡ + Mask	8,6	5,5	12,2	72,3	91,0	96,9
DSN‡	18,7	6,8	33,7	63,3	79,7	84,8

CNN proposta com aquelas obtidas por outros métodos supervisionados da literatura de SIDE. Pela Fig. 50f, é possível notar que a DSN é capaz de estimar melhor a profundidade de objetos mais finos e a estrutura geométrica dos elementos que compõem as cenas, tornando as predições mais próximas do *ground truth*.

Comparando os mapas de profundidade da Fig. 51b com os mapas ilustrados nas Figs. 51b e 51c, pode-se concluir que os resultados da DSN apresentam maiores detalhes tanto para os artefatos mais próximos quanto para os mais distantes da câmera. Além disso, a Fig. 52 mostra que a FCN proposta é capaz de completar os pixels sem informações do *ground truth* e retornar vetores de normais de superfície mais consistentes com o *layout* local e global da cena.

As nuvens de pontos expostas nas Figs. 53c e 53e revelam que as predições da técnica desenvolvida são mais densas do que as nuvens de pontos de referência e que os pixels estimados não obedecem a uma distribuição distorcida das estruturas 3D que retratam os ambientes internos da Fig. 53a.

5.2.3 Estudos de densificação de profundidade

A fim de analisar o desempenho do *framework* de densificação de profundidade em cenários internos, conduziu-se estudos com o *dataset* NYU Depth V2 (SILBERMAN et al., 2012), em que a DSN é alimentada com dados de profundidade esparsos, amostrados a partir do *ground truth*, e com imagens RGB para poder retornar mapas de profundidade densos. Os resultados da FCN introduzida são apresentados na Tabela 18. Nessa tabela, pode-se observar um comportamento dos erros e métricas de precisão semelhante aqueles adquiridos nos estudos *outdoor* com a mesma rede de densificação de profundidade. A Fig. 54 ilustra a resposta da DSN à aplicação de uma quantidade crescente de amostras

Figura 50 – Comparação entre as predições de profundidade de cenas *indoor* do *pipeline* de SIDE proposto e de outros métodos supervisionados no estado-da-arte. A abordagem DSN $\dagger$ está relacionada à configuração da DSN que produz os resultados gerais mais precisos. As imagens de entrada e o *ground truth* pertencem ao *dataset* NYU Depth V2 (SILBERMAN et al., 2012). O *ground truth* é interpolado para fins de visualização

Fonte: adaptada de (MENDES et al., 2021)

esparsas em sua entrada.

Comparou-se os resultados gerados pela melhor configuração da DSN (DSN $\dagger$) com as métricas adquiridas de métodos de densificação de profundidade recentes por meio da Tabela 19. Considerando as métricas Abs Rel e RMSE, alcançou-se o top-2 na classificação de benchmark da Tabela 19. No entanto, a técnica Non-local CSPN (PARK et al., 2020) possui um número maior de parâmetros treináveis (25 M) e aplica uma CNN de extração de características mais complexa (ResNet-34).

A Figura 55 mostra as predições produzidas pela DSN e por técnicas no estado-da-arte que abordam a tarefa de densificação de profundidade. Por meio dessa figura, é possível inferir que os resultados obtidos são os que mais se aproximam da referência. Nesse figura, apresentou-se também os mapas de normais de superfície estimados pela DSN, os quais são visualmente consistentes com os mapas de profundidade de saída. Ademais, as métricas apresentadas na Tabela 20 indicam que, assim como ocorre nas predições de profundidade, a qualidade dos mapas de normais de superfície aumenta à medida que mais dados de profundidade esparsos são adicionados à entrada da rede.

Tabela 18 – Métricas do NYU Depth V2 (SILBERMAN et al., 2012) obtidas pelo *pipeline* de densificação de profundidade. O padrão DSN 3 + Pyramid¹ + Indoor, juntamente com a função $\mathcal{L}_1^+$, é selecionado para todos os experimentos que não envolvem o *ground truth* de normais de superfície. DSN 3 + Pyramid¹ + Indoor + SN † é a configuração da DSN com o módulo de normais de superfície que produz os resultados mais precisos

Método	Cap	Amostras	Abs Rel↓	RMSE↓	log10↓	$\delta < 1, 25^\uparrow$	$\delta < 1, 25^{2^\uparrow}$	$\delta < 1, 25^{3^\uparrow}$
DSN 3 + Pyramid ¹ + Indoor	10	20	0,046	0,221	0,020	0,964	0,992	0,998
DSN 3 + Pyramid ¹ + Indoor	10	50	0,032	0,179	0,014	0,979	0,995	0,998
DSN 3 + Pyramid ¹ + Indoor	10	100	0,024	0,151	0,010	0,986	0,996	0,999
DSN 3 + Pyramid ¹ + Indoor	10	200	0,018	0,132	0,008	0,990	0,997	0,999
DSN 3 + Pyramid ¹ + Indoor	10	500	0,013	0,106	0,006	0,994	0,998	0,999
DSN 3 + Pyramid ¹ + Indoor + SN†	10	200	0,017	0,128	0,007	0,991	0,998	0,999
DSN 3 + Pyramid ¹ + Indoor + SN†	10	500	0,012	0,102	0,005	0,994	0,999	1,000
DSN 3 + Pyramid ¹ + Indoor	5	20	0,045	0,191	0,019	0,965	0,992	0,998
DSN 3 + Pyramid ¹ + Indoor	5	50	0,031	0,153	0,013	0,980	0,995	0,999
DSN 3 + Pyramid ¹ + Indoor	5	100	0,023	0,129	0,010	0,987	0,997	0,999
DSN 3 + Pyramid ¹ + Indoor	5	200	0,018	0,111	0,008	0,990	0,998	0,999
DSN 3 + Pyramid ¹ + Indoor	5	500	0,013	0,090	0,005	0,994	0,998	1,000
DSN 3 + Pyramid ¹ + Indoor + SN†	5	200	0,017	0,107	0,007	0,991	0,998	0,999
DSN 3 + Pyramid ¹ + Indoor + SN†	5	500	0,012	0,086	0,005	0,994	0,999	1,000

Tabela 19 – Avaliação quantitativa do melhor *framework* de densificação de profundidade (DSN †) e de outras técnicas no estado-da-arte. O NYU Depth V2 (SILBERMAN et al., 2012) é utilizado para calcular todas as métricas

Método	Amostras	Abs Rel↓	RMSE↓	$\delta < 1, 25^\uparrow$	$\delta < 1, 25^{2^\uparrow}$	$\delta < 1, 25^{3^\uparrow}$
Ma <i>et al.</i> (MA; KARAMAN, 2018)	200	0,044	0,230	0,971	0,994	0,998
NConv (ELDESOKEY; FELSBERG; KHAN, 2019)	200	0,027	0,173	0,982	0,996	0,999
GuideNet (TANG et al., 2019)	200	0,024	0,142	0,988	0,998	1,000
DSN†	200	0,017	0,128	0,991	0,998	0,999
Silberman <i>et al.</i> (SILBERMAN et al., 2012)	500	0,084	0,479	0,924	0,976	0,989
TGV (FERSTL et al., 2013)	500	0,123	0,635	0,819	0,930	0,968
Zhang <i>et al.</i> (ZHANG; FUNKHOUSER, 2018)	500	0,042	0,228	0,971	0,993	0,997
Ma <i>et al.</i> (MA; KARAMAN, 2018)	500	0,043	0,204	0,978	0,996	0,999
Ma <i>et al.</i> (MA; KARAMAN, 2018) + Bilateral (BARRON; POOLE, 2016)	500	0,084	0,479	0,924	0,976	0,989
Ma <i>et al.</i> (MA; KARAMAN, 2018) + SPN (LIU et al., 2017)	500	0,031	0,172	0,983	0,997	0,999
NConv (ELDESOKEY; FELSBERG; KHAN, 2019)	500	0,018	0,129	0,990	0,998	1,000
CSPN (CHENG; WANG; YANG, 2018)	500	0,016	0,117	0,992	0,999	1,000
Imran <i>et al.</i> (IMRAN et al., 2019)	500	0,013	0,118	0,994	0,999	-
CSPN++ (CHENG et al., 2019)	500	-	0,116	-	-	-
DeepLiDAR (QIU et al., 2019)	500	0,022	0,115	0,993	0,999	1,000
Xu <i>et al.</i> (XU et al., 2019)	500	0,018	0,112	0,995	0,999	1,000
GuideNet (TANG et al., 2019)	500	0,015	0,101	0,995	0,999	1,000
DSN†	500	0,012	0,102	0,994	0,999	1,000
NLSPN (PARK et al., 2020)	500	0,012	0,092	0,996	0,999	1,000

Tabela 20 – Análise das predições de normais de superfície da melhor configuração da FCN de densificação de profundidade (DSN †) quando recebe diferentes quantidades de amostras esparsas como entrada. Mask diz respeito a operação de desconsiderar os pixels inválidos correspondentes ao *ground truth* de normais de superfície

Método	Amostras	Média↓	Mediana↓	RMSE↓	11, 25°↑	22, 50°↑	30°↑
DSN† + Mask	200	7,6	3,5	12,2	77,1	91,0	95,6
DSN†	200	17,8	4,6	33,7	67,6	79,7	83,7
DSN† + Mask	500	6,1	3,0	9,6	82,9	94,9	98,1
DSN†	500	16,5	3,9	33,0	72,6	83,2	86

Figura 51 – Avaliação detalhada das predições do método introduzido e daquelas recuperadas por abordagens supervisionadas recentes. DSN † refere-se à melhor configuração da DSN e as entradas RGB compõem o conjunto de dados NYU Depth V2 (SILBERMAN et al., 2012)

Fonte: adaptada de (MENDES et al., 2021)

Figura 52 – Comparação qualitativa dos mapas de normais de superfície retornados pelo *framework* desenvolvido e por técnicas no estado-da-arte. DSN $\dagger$ corresponde à melhor configuração da DSN. As entradas RGB fazem parte do *dataset* NYU Depth V2 (SILBERMAN et al., 2012). Determinou-se as imagens do *ground truth* através do método de Fouhey, Gupta e Hebert (2013)

Fonte: adaptada de (MENDES et al., 2021)

Figura 53 – Avaliação qualitativa das nuvens de pontos reconstruídas a partir das previsões mais precisas da DSN. As imagens de entrada e o *ground truth* de profundidade compõem o *dataset* NYU Depth V2 (SILBERMAN et al., 2012). No entanto, gerou-se os dados de normais de superfície de referência com o auxílio do algoritmo desenvolvido por Fouhey, Gupta e Hebert (2013)

Fonte: adaptada de (MENDES et al., 2021)

Figura 54 – Evolução das métricas de densificação de profundidade obtidas pela DSN. Empregou-se os mapas de referência de profundidade do conjunto de dados NYU Depth V2 (SILBERMAN et al., 2012) para traçar os gráficos

Fonte: adaptada de (MENDES et al., 2021)

Figura 55 – Comparação entre os mapas de profundidade densos retornados pela melhor configuração da CNN de densificação de profundidade (DSN $\dagger$) e por outros métodos na literatura. Têm-se também os mapas de normais de superfície estimados pela DSN $\dagger$. O NYU Depth V2 (SILBERMAN et al., 2012) é empregado por todas as abordagens. Legenda: RGB - imagem de entrada; 500d - dados de entrada esparsos; GT - *ground truth*

Fonte: adaptada de (MENDES et al., 2021)

6 CONCLUSÃO

Por meio das pesquisas realizadas, é possível inferir que as áreas de estimação monocular de profundidade e densificação de profundidade avançaram significativamente nos últimos anos devido ao surgimento de métodos de Aprendizagem Profunda. Esses métodos podem superar o desempenho de técnicas clássicas de Visão Computacional e também gerar melhores previsões para outras tarefas fundamentais, como segmentação semântica, estimação de normais de superfície e VO.

Particularmente, considerando todas as análises experimentais realizadas, projetou-se uma arquitetura de CNN simples, leve e rápida, cujo número de parâmetros treináveis varia de 2 M até 12 M . O *framework* desenvolvido também pode ser utilizado em aplicações reais, uma vez que ele atinge velocidades de previsão de 32 fps até 88 fps .

Com relação aos testes envolvendo cenários *outdoor*, a configuração da DenseSIDE-Net que inclui o módulo de normais de superfície proposto, a função de custo geométrica $2,5D$, o bloco da pirâmide e o pré-treinamento com o *dataset* do Cityscapes (CORDTS et al., 2016) forneceu resultados no estado-de-arte para as tarefas SIDE ($Abs\ Rel = 0,075$), densificação de profundidade ($Abs\ Rel = 0,019$) e VO ($seq00\ RMSE = 16,9438$), em relação aos conjuntos de dados do KITTI Benchmark (UHRIG et al., 2017; GEIGER; LENZ; URTASUN, 2012; ALHAIJA et al., 2018).

Por outro lado, nos experimentos envolvendo ambientes *indoor*, a mesma configuração da DSN mencionada anteriormente, com exceção da etapa de pré-treinamento que é realizada com o auxílio dos *datasets indoor* (SONG; LICHTENBERG; XIAO, 2015; JANOCH et al., 2013; XIAO; OWENS; TORRALBA, 2013), produziu resultados promissores para previsão de profundidade ($RMSE = 0,429$), densificação de profundidade ($RMSE = 0,102$) e estimação de normais de superfície ($RMSE = 12,2$) nos conjuntos de dados do NYU Benchmark (SILBERMAN et al., 2012).

Portanto, esse trabalho visou abordar a tarefa de estimação monocular de profundidade e outras aplicações correlatas que influenciam e que são influenciadas pelos mapas de profundidade estimados. Foram introduzidas novas estruturas de FCN com alta capacidade de generalização e de atingir um *trade-off* entre precisão e velocidade suficiente para viabilizar suas implementações em sistemas embarcados, como veículos autônomos. Com o auxílio dos resultados experimentais *outdoor* e *indoor* de SIDE alcançados, verificou-se que o sistema proposto obtém previsões com qualidade e precisão semelhantes aos resultados de outros métodos no estado-da-arte. No entanto, até onde este trabalho pôde abranger, a técnica desenvolvida é a que melhor se adapta as demandas dos diferentes ambientes por complexidade e exatidão, o que mostra a eficácia do balanceamento otimizado das

configurações da DenseSIDENet.

Além disso, foi possível determinar como a incorporação de múltiplas técnicas de Aprendizagem Profunda e de Visão Computacional, como os blocos de pirâmide e de normais de superfície, influenciam nos processos de treinamento e de teste da rede de SIDE proposta. Os efeitos benéficos das funções de custo baseadas em atenção a geometria da cena (função de custo $2,5D$) e aos valores de profundidade dos pixels mais distantes e mais próximos do *layout* da imagem também foram avaliados durante os estudos de ablação. Logo, buscou-se condições para potencializar o comportamento da DSN, por meio de diferentes técnicas de complementação e de atenção, as quais permitiram a geração de predições de alta qualidade em tempo de processamento reduzido.

Como trabalhos futuros, pretende-se incorporar novos métodos de atenção na estrutura da DSN, como o modelo proposto por Dosovitskiy et al. (2020), além de outras formas de fusão sensorial com dados de varreduras de *laser* (YAN et al., 2021; HU et al., 2021) e mais vistas da cena analisada para treinar a rede, como nos trabalhos de Godard et al. (2019) e de Casser et al. (2019a). Ademais, planeja-se implementar tal CNN em sistemas reais como no carro autônomo em desenvolvimento no Instituto Nacional de Ciência e Tecnologia para Sistemas Autônomos Cooperativos (InSAC) e no manipulador robótico Kinova Gen3 (KINOVA ROBOTICS, 2019), disponível no Laboratório de Sistemas Inteligentes (LASI) do Departamento de Engenharia Elétrica e de Computação (SEL) da EESC/USP, para auxiliar outras tarefas como preensão robótica e controle servo visual. A mais curto prazo, novas pesquisas estão sendo conduzidas nas áreas de SLAM monocular e VO direta a fim de melhorar o desempenho desses sistemas por meio dos mapas produzidos pela DSN.

6.1 Publicação

Este trabalho levou a publicação dos seguintes artigos:

- (MENDES et al., 2020);
- (MENDES et al., 2021);
- (RIBEIRO; MENDES; JR, 2021).

REFERÊNCIAS

- ABADI, M.; BARHAM, P.; CHEN, J.; CHEN, Z.; DAVIS, A.; DEAN, J.; DEVIN, M.; GHEMAWAT, S.; IRVING, G.; ISARD, M. et al. Tensorflow: a system for large-scale machine learning. In: **USENIX. 12th USENIX Symposium on Operating Systems Design and Implementation**. Savannah, GA, USA, 2016. v. 16, p. 265–283.
- ADAM-BOURDARIOS, C.; COWAN, G.; GERMAIN, C.; GUYON, I.; KÉGL, B.; ROUSSEAU, D. The higgs boson machine learning challenge. In: **NIPS 2014 Workshop on High-energy Physics and Machine Learning**. [S.l.: s.n.], 2015. p. 19–55.
- AFSHARI, H. H.; GADSDEN, S. A.; HABIBI, S. Gaussian filters for parameter and state estimation: A general review of theory and recent trends. **Signal Processing**, Elsevier, v. 135, p. 218–238, 2017.
- ALEOTTI, F.; TOSI, F.; POGGI, M.; MATTOCCIA, S. Generative adversarial networks for unsupervised monocular depth prediction. In: **Proceedings of the European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2018. p. 0–0.
- ALHAIJA, H.; MUSTIKOVELA, S.; MESCHEDER, L.; GEIGER, A.; ROTHER, C. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. **International Journal of Computer Vision (IJCV)**, 2018.
- ALHASHIM, I.; WONKA, P. High quality monocular depth estimation via transfer learning. **arXiv preprint arXiv:1812.11941**, 2018.
- AMIRI, A. J.; LOO, S. Y.; ZHANG, H. Semi-supervised monocular depth estimation with left-right consistency using deep neural network. In: IEEE. **2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)**. [S.l.], 2019. p. 602–607.
- ANDRAGHETTI, L.; MYRIOKEFALITAKIS, P.; DOVESI, P. L.; LUQUE, B.; POGGI, M.; PIEROPAN, A.; MATTOCCIA, S. Enhancing self-supervised monocular depth estimation with traditional visual odometry. In: IEEE. **2019 International Conference on 3D Vision (3DV)**. [S.l.], 2019. p. 424–433.
- BABU, V. M.; DAS, K.; MAJUMDAR, A.; KUMAR, S. Undemon: Unsupervised deep network for depth and ego-motion estimation. In: IEEE. **2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.], 2018. p. 1082–1088.
- BADINO, H.; HUBER, D.; PARK, Y.; KANADE, T. Fast and accurate computation of surface normals from range images. In: IEEE. **2011 IEEE International Conference on Robotics and Automation**. [S.l.], 2011. p. 3084–3091.
- 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.
- BANSAL, A.; RUSSELL, B.; GUPTA, A. Marr revisited: 2d-3d alignment via surface normal prediction. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 5965–5974.

- BARRON, J. T.; POOLE, B. The fast bilateral solver. In: SPRINGER. **European Conference on Computer Vision**. [S.l.], 2016. p. 617–632.
- BAY, H.; TUYTELAARS, T.; GOOL, L. V. Surf: Speeded up robust features. In: SPRINGER. **European conference on computer vision**. [S.l.], 2006. p. 404–417.
- BELLMAN, R. **An introduction to artificial intelligence: Can computers think?** [S.l.]: Thomson Course Technology, 1978.
- BERGSTRA, J.; BREULEUX, O.; BASTIEN, F.; LAMBLIN, P.; PASCANU, R.; DESJARDINS, G.; TURIAN, J.; WARDE-FARLEY, D.; BENGIO, Y. Theano: A cpu and gpu math compiler in python. In: SCIPY. **Proc. 9th Python in Science Conf**. Austin, USA, 2010. v. 1.
- BHAT, S. F.; ALHASHIM, I.; WONKA, P. Adabins: Depth estimation using adaptive bins. **arXiv preprint arXiv:2011.14141**, 2020.
- BIAN, J.-W.; ZHAN, H.; WANG, N.; CHIN, T.-J.; SHEN, C.; REID, I. Unsupervised depth learning in challenging indoor video: Weak rectification to rescue. **arXiv preprint arXiv:2006.02708**, 2020.
- BIEDERMAN, I. Recognition-by-components: a theory of human image understanding. **Psychological review**, American Psychological Association, v. 94, n. 2, p. 115, 1987.
- BINFORD, T. **Visual Perception by Computer**. **IEEE Cont. on Systems and Control**. [S.l.]: Miami, 1971.
- BISHOP, C. M. **Pattern recognition and machine learning**. [S.l.]: springer, 2006.
- CADENA, C.; DICK, A. R.; REID, I. D. Multi-modal auto-encoders as joint estimators for robotics scene understanding. In: **Robotics: Science and Systems**. [S.l.: s.n.], 2016. v. 5, p. 1.
- CALONDER, M.; LEPETIT, V.; STRECHA, C.; FUA, P. Brief: Binary robust independent elementary features. In: SPRINGER. **European conference on computer vision**. [S.l.], 2010. p. 778–792.
- CAO, Y.; WU, Z.; SHEN, C. Estimating depth from monocular images as classification using deep fully convolutional residual networks. **IEEE Transactions on Circuits and Systems for Video Technology**, IEEE, v. 28, n. 11, p. 3174–3182, 2017.
- CARVALHO, M.; SAUX, B. L.; TROUVÉ-PELOUX, P.; ALMANSA, A.; CHAMPAGNAT, F. On regression losses for deep depth estimation. In: IEEE. **2018 25th IEEE International Conference on Image Processing (ICIP)**. [S.l.], 2018. p. 2915–2919.
- CASSER, V.; PIRK, S.; MAHJOURIAN, R.; ANGELOVA, A. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In: **Proceedings of the AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2019. v. 33, p. 8001–8008.
- CASSER, V.; PIRK, S.; MAHJOURIAN, R.; ANGELOVA, A. Unsupervised monocular depth and ego-motion learning with structure and semantics. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops**. [S.l.: s.n.], 2019. p. 0–0.

- CHAKRABARTI, A.; SHAO, J.; SHAKHNAROVICH, G. Depth from a single image by harmonizing overcomplete local network predictions. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2016. p. 2658–2666.
- CHANG, J.; WETZSTEIN, G. Deep optics for monocular depth estimation and 3d object detection. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2019. p. 10193–10202.
- CHARBONNIER, P.; BLANC-FERAUD, L.; AUBERT, G.; BARLAUD, M. Two deterministic half-quadratic regularization algorithms for computed imaging. In: **IEEE. Proceedings of 1st International Conference on Image Processing**. [S.l.], 1994. v. 2, p. 168–172.
- CHARNIAK, E. **Introduction to artificial intelligence**. [S.l.]: Pearson Education India, 1985.
- CHEN, L.; TANG, W.; WAN, T. R.; JOHN, N. W. Self-supervised monocular image depth learning and confidence estimation. **Neurocomputing**, Elsevier, v. 381, p. 272–281, 2020.
- CHEN, L.-C.; PAPANDREOU, G.; KOKKINOS, I.; MURPHY, K.; YUILLE, A. L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 40, n. 4, p. 834–848, 2017.
- CHEN, P.; CHEN, G.; ZHANG, S. Log hyperbolic cosine loss improves variational auto-encoder. 2018.
- CHEN, P.-Y.; LIU, A. H.; LIU, Y.-C.; WANG, Y.-C. F. Towards scene understanding: Unsupervised monocular depth estimation with semantic-aware representation. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2019. p. 2624–2632.
- CHEN, R.; MAHMOOD, F.; YUILLE, A.; DURR, N. J. Rethinking monocular depth estimation with adversarial training. **arXiv preprint arXiv:1808.07528**, 2018.
- CHEN, T.; LI, M.; LI, Y.; LIN, M.; WANG, N.; WANG, M.; XIAO, T.; XU, B.; ZHANG, C.; ZHANG, Z. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. **arXiv preprint arXiv:1512.01274**, 2015.
- CHEN, W.; YUE, H.; WANG, J.; WU, X. An improved edge detection algorithm for depth map inpainting. **Optics and Lasers in Engineering**, Elsevier, v. 55, p. 69–77, 2014.
- CHEN, X.; CHEN, X.; ZHA, Z.-J. Structure-aware residual pyramid network for monocular depth estimation. **arXiv preprint arXiv:1907.06023**, 2019.
- CHEN, Y.; SCHMID, C.; SMINCHISESCU, C. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2019. p. 7063–7072.

- CHEN, Y.; YANG, B.; LIANG, M.; URTASUN, R. Learning joint 2d-3d representations for depth completion. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2019. p. 10023–10032.
- CHEN, Y.; ZHAO, H.; HU, Z. Attention-based context aggregation network for monocular depth estimation. **arXiv preprint arXiv:1901.10137**, 2019.
- CHENG, X.; WANG, P.; GUAN, C.; YANG, R. Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion. **arXiv preprint arXiv:1911.05377**, 2019.
- CHENG, X.; WANG, P.; YANG, R. Depth estimation via affinity learned with convolutional spatial propagation network. In: **Proceedings of the European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2018. p. 103–119.
- CHODOSH, N.; WANG, C.; LUCEY, S. Deep convolutional compressed sensing for lidar depth completion. In: SPRINGER. **Asian Conference on Computer Vision**. [S.l.], 2018. p. 499–513.
- CHOI, W.; CHAO, Y.-W.; PANTOFARU, C.; SAVARESE, S. Understanding indoor scenes using 3d geometric phrases. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2013. p. 33–40.
- CHOLLET, F. Xception: Deep learning with depthwise separable convolutions. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 1251–1258.
- CHOLLET, F. et al. **Keras**. 2015.
- CIREŞAN, D.; MEIER, U.; MASCI, J.; SCHMIDHUBER, J. Multi-column deep neural network for traffic sign classification. **Neural networks**, Elsevier, v. 32, p. 333–338, 2012.
- CLEVERT, D.-A.; UNTERTHINER, T.; HOCHREITER, S. Fast and accurate deep network learning by exponential linear units (elus). **arXiv preprint arXiv:1511.07289**, 2015.
- CORDTS, M.; OMRAN, M.; RAMOS, S.; REHFELD, T.; ENZWEILER, M.; BENENSON, R.; FRANKE, U.; ROTH, S.; SCHIELE, B. 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.
- COUGHLAN, J. M.; YUILLE, A. L. The manhattan world assumption: Regularities in scene statistics which enable bayesian inference. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2001. p. 845–851.
- DAHL, G. E.; SAINATH, T. N.; HINTON, G. E. Improving deep neural networks for lvcsr using rectified linear units and dropout. In: IEEE. **2013 IEEE international conference on acoustics, speech and signal processing**. [S.l.], 2013. p. 8609–8613.
- DIMITRIEVSKI, M.; VEELAERT, P.; PHILIPS, W. Learning morphological operators for depth completion. In: SPRINGER. **International Conference on Advanced Concepts for Intelligent Vision Systems**. [S.l.], 2018. p. 450–461.

- DOSOVITSKIY, A.; BEYER, L.; KOLESNIKOV, A.; WEISSENBORN, D.; ZHAI, X.; UNTERTHINER, T.; DEHGHANI, M.; MINDERER, M.; HEIGOLD, G.; GELLY, S. et al. An image is worth 16x16 words: Transformers for image recognition at scale. **arXiv preprint arXiv:2010.11929**, 2020.
- DOZAT, T. Incorporating nesterov momentum into adam. 2016.
- DUCHI, J.; HAZAN, E.; SINGER, Y. Adaptive subgradient methods for online learning and stochastic optimization. **Journal of machine learning research**, v. 12, n. Jul, p. 2121–2159, 2011.
- DUMOULIN, V.; VISIN, F. A guide to convolution arithmetic for deep learning. **arXiv preprint arXiv:1603.07285**, 2016.
- EIGEN, D.; FERGUS, R. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: **Proceedings of the IEEE international conference on computer vision**. [S.l.: s.n.], 2015. p. 2650–2658.
- EIGEN, D.; PUHRSCHE, C.; FERGUS, R. Depth map prediction from a single image using a multi-scale deep network. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2014. p. 2366–2374.
- ELDESOKEY, A.; FELSBURG, M.; KHAN, F. S. Confidence propagation through cnns for guided sparse depth regression. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, 2019.
- ELKERDAWY, S.; ZHANG, H.; RAY, N. Lightweight monocular depth estimation model by joint end-to-end filter pruning. In: IEEE. **2019 IEEE International Conference on Image Processing (ICIP)**. [S.l.], 2019. p. 4290–4294.
- ENGEL, J.; KOLTUN, V.; CREMERS, D. Direct sparse odometry. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 40, n. 3, p. 611–625, 2017.
- FAN, R.; WANG, H.; XUE, B.; HUANG, H.; WANG, Y.; LIU, M.; PITAS, I. Three-filters-to-normal: An accurate and ultrafast surface normal estimator. **IEEE Robotics and Automation Letters**, IEEE, 2021.
- FARABET, C.; COUPRIE, C.; NAJMAN, L.; LECUN, Y. Learning hierarchical features for scene labeling. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 35, n. 8, p. 1915–1929, 2012.
- FERSTL, D.; REINBACHER, C.; RANFTL, R.; RÜTHER, M.; BISCHOF, H. Image guided depth upsampling using anisotropic total generalized variation. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2013. p. 993–1000.
- FORSTER, C.; PIZZOLI, M.; SCARAMUZZA, D. Svo: Fast semi-direct monocular visual odometry. In: IEEE. **2014 IEEE international conference on robotics and automation (ICRA)**. [S.l.], 2014. p. 15–22.
- FORSTER, C. Q.; TOZZI, C. L. Towards 3d reconstruction of endoscope images using shape from shading. In: IEEE. **Proceedings 13th Brazilian Symposium on Computer Graphics and Image Processing (Cat. No. PR00878)**. [S.l.], 2000. p. 90–96.

- FOUHEY, D. F.; GUPTA, A.; HEBERT, M. Data-driven 3d primitives for single image understanding. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2013. p. 3392–3399.
- FOUHEY, D. F.; GUPTA, A.; HEBERT, M. Unfolding an indoor origami world. In: SPRINGER. **European Conference on Computer Vision**. [S.l.], 2014. p. 687–702.
- FREEMAN, W. T.; JONES, T. R.; PASZTOR, E. C. Example-based super-resolution. **IEEE Computer graphics and Applications**, IEEE, v. 22, n. 2, p. 56–65, 2002.
- FU, H.; GONG, M.; WANG, C.; BATMANGHELICH, K.; TAO, D. Deep ordinal regression network for monocular depth estimation. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 2002–2011.
- FU, J.; WANG, S.; LU, Y.; LI, S.; ZENG, W. Kinect-like depth denoising. In: IEEE. **2012 IEEE International Symposium on Circuits and Systems**. [S.l.], 2012. p. 512–515.
- FUKUSHIMA, K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. **Biological cybernetics**, Springer, v. 36, n. 4, p. 193–202, 1980.
- FUSIELLO, A.; TRUCCO, E.; VERRI, A. A compact algorithm for rectification of stereo pairs. **Machine vision and applications**, Springer, v. 12, n. 1, p. 16–22, 2000.
- GAN, Y.; XU, X.; SUN, W.; LIN, L. Monocular depth estimation with affinity, vertical pooling, and label enhancement. In: **Proceedings of the European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2018. p. 224–239.
- GANSBEKE, W. V.; NEVEN, D.; BRABANDERE, B. D.; GOOL, L. V. Sparse and noisy lidar completion with rgb guidance and uncertainty. In: IEEE. **2019 16th International Conference on Machine Vision Applications (MVA)**. [S.l.], 2019. p. 1–6.
- GARG, R.; BG, V. K.; CARNEIRO, G.; REID, I. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In: SPRINGER. **European Conference on Computer Vision**. [S.l.], 2016. p. 740–756.
- GEIGER, A.; LENZ, P.; STILLER, C.; URTASUN, R. 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.
- GEIGER, A.; LENZ, P.; URTASUN, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In: **Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2012.
- GENESERETH, M. R.; NILSSON, N. J. **Logical foundations of artificial intelligence**. [S.l.]: Morgan Kaufmann, 2012.
- GIRSHICK, R. Fast r-cnn. In: **Proceedings of the IEEE international conference on computer vision**. [S.l.: s.n.], 2015. p. 1440–1448.
- GODARD, C.; AODHA, O. M.; BROSTOW, G. J. Unsupervised monocular depth estimation with left-right consistency. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2017. p. 270–279.

- GODARD, C.; AODHA, O. M.; FIRMAN, M.; BROSTOW, G. J. Digging into self-supervised monocular depth estimation. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2019. p. 3828–3838.
- GOLDMAN, M.; HASSNER, T.; AVIDAN, S. Learn stereo, infer mono: Siamese networks for self-supervised, monocular, depth estimation. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops**. [S.l.: s.n.], 2019. p. 0–0.
- GONZALEZBELLO, J. L.; KIM, M. Forget about the lidar: Self-supervised depth estimators with med probability volumes. **Advances in Neural Information Processing Systems**, v. 33, 2020.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A.; BENGIO, Y. **Deep learning**. Cambridge, USA: MIT Press, 2016. v. 1.
- GOODFELLOW, I. J.; WARDE-FARLEY, D.; LAMBLIN, P.; DUMOULIN, V.; MIRZA, M.; PASCANU, R.; BERGSTRA, J.; BASTIEN, F.; BENGIO, Y. Pylearn2: a machine learning research library. **arXiv preprint arXiv:1308.4214**, 2013.
- GORDON, A.; LI, H.; JONSKOWSKI, R.; ANGELOVA, A. Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2019. p. 8977–8986.
- GOULD, S.; FULTON, R.; KOLLER, D. Decomposing a scene into geometric and semantically consistent regions. In: IEEE. **2009 IEEE 12th international conference on computer vision**. [S.l.], 2009. p. 1–8.
- GUIZILINI, V.; AMBRUS, R.; PILLAI, S.; GAIDON, A. Packnet-sfm: 3d packing for self-supervised monocular depth estimation. **arXiv preprint arXiv:1905.02693**, 2019.
- GUIZILINI, V.; HOU, R.; LI, J.; AMBRUS, R.; GAIDON, A. Semantically-guided representation learning for self-supervised monocular depth. **arXiv preprint arXiv:2002.12319**, 2020.
- GUO, X.; LI, H.; YI, S.; REN, J.; WANG, X. Learning monocular depth by distilling cross-domain stereo networks. In: **Proceedings of the European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2018. p. 484–500.
- GUPTA, A.; EFROS, A. A.; HEBERT, M. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In: SPRINGER. **European Conference on Computer Vision**. [S.l.], 2010. p. 482–496.
- GUPTA, A.; HEBERT, M.; KANADE, T.; BLEI, D. M. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2010. p. 1288–1296.
- GUPTA, H.; MITRA, K. Unsupervised single image underwater depth estimation. In: IEEE. **2019 IEEE International Conference on Image Processing (ICIP)**. [S.l.], 2019. p. 624–628.

- GUPTA, S.; GIRSHICK, R.; ARBELÁEZ, P.; MALIK, J. Learning rich features from rgb-d images for object detection and segmentation. In: SPRINGER. **European conference on computer vision**. [S.l.], 2014. p. 345–360.
- GUR, S.; WOLF, L. Single image depth estimation trained via depth from defocus cues. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2019. p. 7683–7692.
- GURRAM, A.; URFALIOGLU, O.; HALFAOUI, I.; BOUZARAA, F.; LÓPEZ, A. M. Monocular depth estimation by learning from heterogeneous datasets. In: IEEE. **2018 IEEE Intelligent Vehicles Symposium (IV)**. [S.l.], 2018. p. 2176–2181.
- HANE, C.; LADICKY, L.; POLLEFEYS, M. Direction matters: Depth estimation with a surface normal classifier. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2015. p. 381–389.
- HAO, Z.; LI, Y.; YOU, S.; LU, F. Detail preserving depth estimation from a single image using attention guided networks. In: IEEE. **2018 International Conference on 3D Vision (3DV)**. [S.l.], 2018. p. 304–313.
- HARTLEY, R.; ZISSERMAN, A. **Multiple view geometry in computer vision**. [S.l.]: Cambridge university press, 2003.
- HAUGELAND, J. **Artificial intelligence: The very idea**. [S.l.]: MIT press, 1989.
- HE, K.; GKIOXARI, G.; DOLLÁR, P.; GIRSHICK, R. Mask r-cnn. In: **Proceedings of the IEEE international conference on computer vision**. [S.l.: s.n.], 2017. p. 2961–2969.
- HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: IEEE. **Proceedings of the IEEE conference on computer vision and pattern recognition**. Las Vegas, USA, 2016. p. 770–778.
- HEDAU, V.; HOIEM, D.; FORSYTH, D. Recovering the spatial layout of cluttered rooms. In: IEEE. **2009 IEEE 12th international conference on computer vision**. [S.l.], 2009. p. 1849–1856.
- HEDAU, V.; HOIEM, D.; FORSYTH, D. Thinking inside the box: Using appearance models and context based on room geometry. In: SPRINGER. **European Conference on Computer Vision**. [S.l.], 2010. p. 224–237.
- HEKMATIAN, H.; AL-STOUHI, S.; JIN, J. Conf-net: Predicting depth completion error-map for high-confidence dense 3d point-cloud. **arXiv preprint arXiv:1907.10148**, 2019.
- HELMKE, U.; HÜPER, K.; LEE, P.; MOORE, J. et al. Essential matrix estimation via newton-type methods. In: **Proceedings of the MTNS**. [S.l.: s.n.], 2004.
- HERRERA, D.; KANNALA, J.; HEIKKILÄ, J. et al. Depth map inpainting under a second-order smoothness prior. In: SPRINGER. **Scandinavian Conference on Image Analysis**. [S.l.], 2013. p. 555–566.

- HINTON, G.; DENG, L.; YU, D.; DAHL, G. E.; MOHAMED, A.-r.; JAITLEY, N.; SENIOR, A.; VANHOUCKE, V.; NGUYEN, P.; SAINATH, T. N. et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. **IEEE Signal processing magazine**, IEEE, v. 29, n. 6, p. 82–97, 2012.
- HIRSCHMULLER, H. Accurate and efficient stereo processing by semi-global matching and mutual information. In: IEEE. **2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)**. [S.l.], 2005. v. 2, p. 807–814.
- HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.
- HOIEM, D.; EFROS, A. A.; HEBERT, M. Automatic photo pop-up. In: **ACM SIGGRAPH 2005 Papers**. [S.l.: s.n.], 2005. p. 577–584.
- HOIEM, D.; EFROS, A. A.; HEBERT, M. Geometric context from a single image. In: IEEE. **Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1**. [S.l.], 2005. v. 1, p. 654–661.
- HOIEM, D.; EFROS, A. A.; HEBERT, M. Recovering surface layout from an image. **International Journal of Computer Vision**, Springer, v. 75, n. 1, p. 151–172, 2007.
- HOIEM, D.; EFROS, A. A.; HEBERT, M. Closing the loop in scene interpretation. In: IEEE. **2008 IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.], 2008. p. 1–8.
- HOIEM, D.; EFROS, A. A.; HEBERT, M. Recovering occlusion boundaries from an image. **International Journal of Computer Vision**, Springer, v. 91, n. 3, p. 328–346, 2011.
- HOWARD, A. G.; ZHU, M.; CHEN, B.; KALENICHENKO, D.; WANG, W.; WEYAND, T.; ANDREETTO, M.; ADAM, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. **arXiv preprint arXiv:1704.04861**, 2017.
- HU, J.; OZAY, M.; ZHANG, Y.; OKATANI, T. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In: IEEE. **2019 IEEE Winter Conference on Applications of Computer Vision (WACV)**. [S.l.], 2019. p. 1043–1051.
- HU, J.; SHEN, L.; SUN, G. Squeeze-and-excitation networks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2018. p. 7132–7141.
- HU, M.; WANG, S.; LI, B.; NING, S.; FAN, L.; GONG, X. Penet: Towards precise and efficient image guided depth completion. **arXiv preprint arXiv:2103.00783**, 2021.
- HU, W.; LI, X.; CHEUNG, G.; AU, O. Depth map denoising using graph-based transform and group sparsity. In: IEEE. **2013 IEEE 15th International Workshop on Multimedia Signal Processing (MMSP)**. [S.l.], 2013. p. 001–006.
- HUANG, G.; LIU, Z.; MAATEN, L. V. D.; WEINBERGER, K. Q. Densely connected convolutional networks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 4700–4708.

HUANG, Z.; FAN, J.; CHENG, S.; YI, S.; WANG, X.; LI, H. Hms-net: Hierarchical multi-scale sparsity-invariant network for sparse depth completion. **IEEE Transactions on Image Processing**, IEEE, 2019.

IMRAN, S.; LONG, Y.; LIU, X.; MORRIS, D. Depth coefficients for depth completion. In: IEEE. **2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.], 2019. p. 12438–12447.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. **arXiv preprint arXiv:1502.03167**, 2015.

JANOCH, A.; KARAYEV, S.; JIA, Y.; BARRON, J. T.; FRITZ, M.; SAENKO, K.; DARRELL, T. A category-level 3d object dataset: Putting the kinect to work. In: **Consumer depth cameras for computer vision**. [S.l.]: Springer, 2013. p. 141–165.

JARITZ, M.; CHARETTE, R. D.; WIRBEL, E.; PERROTTON, X.; NASHASHIBI, F. Sparse and dense data with cnns: Depth completion and semantic segmentation. In: IEEE. **2018 International Conference on 3D Vision (3DV)**. [S.l.], 2018. p. 52–60.

JIA, Y.; SHELHAMER, E.; DONAHUE, J.; KARAYEV, S.; LONG, J.; GIRSHICK, R.; GUADARRAMA, S.; DARRELL, T. Caffe: Convolutional architecture for fast feature embedding. In: ACM. **Proceedings of the 22nd ACM international conference on Multimedia**. Orlando, USA, 2014. p. 675–678.

JIAO, J.; CAO, Y.; SONG, Y.; LAU, R. Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss. In: **Proceedings of the European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2018. p. 53–69.

JORDAN, K.; MORDOHAJ, P. A quantitative evaluation of surface normal estimation in point clouds. In: IEEE. **2014 IEEE/RSJ International Conference on Intelligent Robots and Systems**. [S.l.], 2014. p. 4220–4226.

KARSCH, K.; LIU, C.; KANG, S. B. Depth transfer: Depth extraction from video using non-parametric sampling. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 36, n. 11, p. 2144–2158, 2014.

KIEFER, J.; WOLFOWITZ, J. et al. Stochastic estimation of the maximum of a regression function. **The Annals of Mathematical Statistics**, Institute of Mathematical Statistics, v. 23, n. 3, p. 462–466, 1952.

KIM, Y.; JUNG, H.; MIN, D.; SOHN, K. Deep monocular depth estimation via integration of global and local predictions. **IEEE transactions on Image Processing**, IEEE, v. 27, n. 8, p. 4131–4144, 2018.

KINGMA, D. P.; BA, J. **Adam: A Method for Stochastic Optimization**. 2014.

KINOVA ROBOTICS. **KINOVA Gen3 Ultra lightweight robot User Guide**. 2019.

KLASING, K.; ALTHOFF, D.; WOLLHERR, D.; BUSS, M. Comparison of surface normal estimation methods for range sensing applications. In: IEEE. **2009 IEEE International Conference on Robotics and Automation**. [S.l.], 2009. p. 3206–3211.

- KLASING, K.; WOLLHERR, D.; BUSS, M. Realtime segmentation of range data using continuous nearest neighbors. In: IEEE. **2009 IEEE International Conference on Robotics and Automation**. [S.l.], 2009. p. 2431–2436.
- KLODT, M.; VEDALDI, A. Supervising the new with the old: learning sfm from sfm. In: **Proceedings of the European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2018. p. 698–713.
- KONG, S.; FOWLKES, C. Pixel-wise attentional gating for scene parsing. In: IEEE. **2019 IEEE Winter Conference on Applications of Computer Vision (WACV)**. [S.l.], 2019. p. 1024–1033.
- KOPIER, A. A.; SILVA, V.; OLIVEIRA, L.; LINDEN, R.; SILVA, L.; FONSECA, B. L. d. C. Redes neurais artificiais e suas aplicações no setor elétrico. **Revista de Engenharias da Faculdade Salesiana**, n. 9, p. 27–33, 2019.
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2012. p. 1097–1105.
- KROGH, A.; HERTZ, J. A. A simple weight decay can improve generalization. In: **Advances in neural information processing systems**. [S.l.: s.n.], 1992. p. 950–957.
- KU, J.; HARAKEH, A.; WASLANDER, S. L. In defense of classical image processing: Fast depth completion on the cpu. In: IEEE. **2018 15th Conference on Computer and Robot Vision (CRV)**. [S.l.], 2018. p. 16–22.
- KULKARNI, M.; RAJAGOPALAN, A. N. Depth inpainting by tensor voting. **JOSA A**, Optical Society of America, v. 30, n. 6, p. 1155–1165, 2013.
- KUMAR, A. C.; BHANDARKAR, S. M.; PRASAD, M. Depthnet: A recurrent neural network architecture for monocular depth prediction. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops**. [S.l.: s.n.], 2018. p. 283–291.
- KUMAR, A. C.; BHANDARKAR, S. M.; PRASAD, M. Monocular depth prediction using generative adversarial networks. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops**. [S.l.: s.n.], 2018. p. 300–308.
- KUNDU, J. N.; UPPALA, P. K.; PAHUJA, A.; BABU, R. V. Adadepth: Unsupervised content congruent adaptation for depth estimation. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 2656–2665.
- KURZWEIL, R.; RICHTER, R.; KURZWEIL, R.; SCHNEIDER, M. L. **The age of intelligent machines**. [S.l.]: MIT press Cambridge, 1990. v. 579.
- KUZNIETSOV, Y.; STUCKLER, J.; LEIBE, B. Semi-supervised deep learning for monocular depth map prediction. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 6647–6655.

- LADICKY, L.; SHI, J.; POLLEFEYS, M. Pulling things out of perspective. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2014. p. 89–96.
- LADICKY, L.; ZEISL, B.; POLLEFEYS, M. Discriminatively trained dense surface normal estimation. In: **ECCV**. [S.l.: s.n.], 2014. v. 2, n. 3, p. 4.
- LAINA, I.; RUPPRECHT, C.; BELAGIANNIS, V.; TOMBARI, F.; NAVAB, N. Deeper depth prediction with fully convolutional residual networks. In: IEEE. **2016 Fourth international conference on 3D vision (3DV)**. [S.l.], 2016. p. 239–248.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **nature**, Nature Publishing Group, v. 521, n. 7553, p. 436, 2015.
- LECUN, Y. et al. Generalization and network design strategies. **Connectionism in perspective**, Citeseer, p. 143–155, 1989.
- LEE, D. C.; HEBERT, M.; KANADE, T. Geometric reasoning for single image structure recovery. In: IEEE. **2009 IEEE conference on computer vision and pattern recognition**. [S.l.], 2009. p. 2136–2143.
- LEE, J. H.; HAN, M.-K.; KO, D. W.; SUH, I. H. From big to small: Multi-scale local planar guidance for monocular depth estimation. **arXiv preprint arXiv:1907.10326**, 2019.
- LEE, J.-H.; HEO, M.; KIM, K.-R.; KIM, C.-S. Single-image depth estimation based on fourier domain analysis. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 330–339.
- LEE, J.-H.; KIM, C.-S. Monocular depth estimation using relative depth maps. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2019. p. 9729–9738.
- LEE, S.; LEE, J.; KIM, D.; KIM, J. Deep architecture with cross guidance between single image and sparse lidar data for depth completion. **IEEE Access**, IEEE, v. 8, p. 79801–79810, 2020.
- LENSSEN, J. E.; OSENDORFER, C.; MASCI, J. Deep iterative surface normal estimation. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2020. p. 11247–11256.
- LI, A.; YUAN, Z.; LING, Y.; CHI, W.; ZHANG, C. et al. A multi-scale guided cascade hourglass network for depth completion. In: **The IEEE Winter Conference on Applications of Computer Vision**. [S.l.: s.n.], 2020. p. 32–40.
- LI, B.; DAI, Y.; HE, M. Monocular depth estimation with hierarchical fusion of dilated cnns and soft-weighted-sum inference. **Pattern Recognition**, Elsevier, v. 83, p. 328–339, 2018.
- LI, B.; SHEN, C.; DAI, Y.; HENGEL, A. V. D.; HE, M. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 1119–1127.

- LI, J.; KLEIN, R.; YAO, A. A two-streamed network for estimating fine-scaled depth maps from single rgb images. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2017. p. 3372–3380.
- LI, R.; WANG, S.; LONG, Z.; GU, D. Undeepvo: Monocular visual odometry through unsupervised deep learning. In: IEEE. **2018 IEEE international conference on robotics and automation (ICRA)**. [S.l.], 2018. p. 7286–7291.
- LI, R.; XIAN, K.; SHEN, C.; CAO, Z.; LU, H.; HANG, L. Deep attention-based classification network for robust depth prediction. In: SPRINGER. **Asian Conference on Computer Vision**. [S.l.], 2018. p. 663–678.
- LIAO, Y.; HUANG, L.; WANG, Y.; KODAGODA, S.; YU, Y.; LIU, Y. Parse geometry from a line: Monocular depth estimation with partial laser observation. In: IEEE. **2017 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.], 2017. p. 5059–5066.
- LIU, B.; GOULD, S.; KOLLER, D. Single image depth estimation from predicted semantic labels. In: IEEE. **2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition**. [S.l.], 2010. p. 1253–1260.
- LIU, F.; SHEN, C.; LIN, G. Deep convolutional neural fields for depth estimation from a single image. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 5162–5170.
- LIU, F.; SHEN, C.; LIN, G.; REID, I. Learning depth from single monocular images using deep convolutional neural fields. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 38, n. 10, p. 2024–2039, 2015.
- LIU, J.; LI, Q.; CAO, R.; TANG, W.; QIU, G. Mininet: An extremely lightweight convolutional neural network for real-time unsupervised monocular depth estimation. **ISPRS Journal of Photogrammetry and Remote Sensing**, Elsevier, v. 166, p. 255–267, 2020.
- LIU, L.-K.; CHAN, S. H.; NGUYEN, T. Q. Depth reconstruction from sparse samples: Representation, algorithm, and sampling. **IEEE Transactions on Image Processing**, IEEE, v. 24, n. 6, p. 1983–1996, 2015.
- LIU, M.; SALZMANN, M.; HE, X. Discrete-continuous depth estimation from a single image. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2014. p. 716–723.
- LIU, S.; MELLO, S. D.; GU, J.; ZHONG, G.; YANG, M.-H.; KAUTZ, J. Learning affinity via spatial propagation networks. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2017. p. 1520–1530.
- LOO, S. Y.; AMIRI, A. J.; MASHOHOR, S.; TANG, S. H.; ZHANG, H. Cnn-svo: Improving the mapping in semi-direct visual odometry using single-image depth prediction. In: IEEE. **2019 International Conference on Robotics and Automation (ICRA)**. [S.l.], 2019. p. 5218–5223.
- LOWE, D. G. Distinctive image features from scale-invariant keypoints. **International journal of computer vision**, Springer, v. 60, n. 2, p. 91–110, 2004.

- LU, H.; DAI, Y.; SHEN, C.; XU, S. Index network. **arXiv preprint arXiv:1908.09895**, 2019.
- LU, J.; FORSYTH, D. Sparse depth super resolution. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2015. p. 2245–2253.
- LU, J.; MIN, D.; PAHWA, R. S.; DO, M. N. A revisit to mrf-based depth map super-resolution and enhancement. In: IEEE. **2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.], 2011. p. 985–988.
- LU, K.; BARNES, N.; ANWAR, S.; ZHENG, L. From depth what can you see? depth completion via auxiliary image reconstruction. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2020. p. 11306–11315.
- LUGER, G.; STUBBLEFIELD, W. Artificial intelligence: its roots and scope. **Artificial intelligence: structures and strategies for**, p. 1–34, 1993.
- LUO, C.; YANG, Z.; WANG, P.; WANG, Y.; XU, W.; NEVATIA, R.; YUILLE, A. Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding. **arXiv preprint arXiv:1810.06125**, 2018.
- LUO, Y.; REN, J.; LIN, M.; PANG, J.; SUN, W.; LI, H.; LIN, L. Single view stereo matching. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 155–163.
- MA, F.; CAVALHEIRO, G. V.; KARAMAN, S. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In: IEEE. **2019 International Conference on Robotics and Automation (ICRA)**. [S.l.], 2019. p. 3288–3295.
- MA, F.; KARAMAN, S. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In: IEEE. **2018 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.], 2018. p. 1–8.
- MA, J.; SHERIDAN, R. P.; LIAW, A.; DAHL, G. E.; SVETNIK, V. Deep neural nets as a method for quantitative structure–activity relationships. **Journal of chemical information and modeling**, ACS Publications, v. 55, n. 2, p. 263–274, 2015.
- MAHJOURIAN, R.; WICKE, M.; ANGELOVA, A. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 5667–5675.
- MAHMOOD, F.; DURR, N. J. Deep learning and conditional random fields-based depth estimation and topographical reconstruction from conventional endoscopy. **Medical image analysis**, Elsevier, v. 48, p. 230–243, 2018.
- MALLICK, S. P.; ZICKLER, T. E.; KRIEGMAN, D. J.; BELHUMEUR, P. N. Beyond lambert: Reconstructing specular surfaces using color. In: IEEE. **2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)**. [S.l.], 2005. v. 2, p. 619–626.

- MALLYA, A.; LAZEBNIK, S. Learning informative edge maps for indoor scene layout prediction. In: **Proceedings of the IEEE international conference on computer vision**. [S.l.: s.n.], 2015. p. 936–944.
- MANCINI, M.; COSTANTE, G.; VALIGI, P.; CIARFUGLIA, T. A.; DELMERICO, J.; SCARAMUZZA, D. Toward domain independence for learning-based monocular depth estimation. **IEEE Robotics and Automation Letters**, IEEE, v. 2, n. 3, p. 1778–1785, 2017.
- 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.
- MEHTA, I.; SAKURIKAR, P.; NARAYANAN, P. Structured adversarial training for unsupervised monocular depth estimation. In: IEEE. **2018 International Conference on 3D Vision (3DV)**. [S.l.], 2018. p. 314–323.
- MENDES, R. de Q.; RIBEIRO, E. G.; ROSA, N. dos S.; JR, V. G. On deep learning techniques to boost monocular depth estimation for autonomous navigation. **Robotics and Autonomous Systems**, Elsevier, v. 136, p. 103701, 2021.
- MENDES, R. Q.; RIBEIRO, E. G.; ROSA, N. S.; GRASSI, V. Depth completion with morphological operations: An intermediate approach to enhance monocular depth estimation. In: IEEE. **2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and 2020 Workshop on Robotics in Education (WRE)**. [S.l.], 2020. p. 1–6.
- MOUSAVIAN, A.; PIRSIAVASH, H.; KOŠECKÁ, J. Joint semantic segmentation and depth estimation with deep convolutional networks. In: IEEE. **2016 Fourth International Conference on 3D Vision (3DV)**. [S.l.], 2016. p. 611–619.
- MUR-ARTAL, R.; TARDÓS, J. D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. **IEEE Transactions on Robotics**, IEEE, v. 33, n. 5, p. 1255–1262, 2017.
- MURDOCK, C.; CHANG, M.; LUCEY, S. Deep component analysis via alternating direction neural networks. In: **Proceedings of the European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2018. p. 820–836.
- MURPHY, K. P. **Machine learning: a probabilistic perspective**. [S.l.]: MIT press, 2012.
- NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: **Proceedings of the 27th international conference on machine learning (ICML-10)**. [S.l.: s.n.], 2010. p. 807–814.
- NEKRASOV, V.; DHARMASIRI, T.; SPEK, A.; DRUMMOND, T.; SHEN, C.; REID, I. Real-time joint semantic segmentation and depth estimation using asymmetric annotations. In: IEEE. **2019 International Conference on Robotics and Automation (ICRA)**. [S.l.], 2019. p. 7101–7107.
- NILSSON, N. J. **Principles of artificial intelligence**. [S.l.]: Morgan Kaufmann, 2014.

- NILSSON, N. J.; NILSSON, N. J. **Artificial intelligence: a new synthesis**. [S.l.]: Morgan Kaufmann, 1998.
- NIXON, M.; AGUADO, A. **Feature extraction and image processing for computer vision**. [S.l.]: Academic press, 2019.
- OCHS, M.; KRETZ, A.; MESTER, R. Sdnet: Semantically guided depth estimation network. In: SPRINGER. **German Conference on Pattern Recognition**. [S.l.], 2019. p. 288–302.
- PAGLIARI, D.; PINTO, L. Calibration of kinect for xbox one and comparison between the two generations of microsoft sensors. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 15, n. 11, p. 27569–27589, 2015.
- PAPADAKIS, J.; WILLIS, A. R. Real-time surface fitting to rgb-d sensor data. In: IEEE. **SoutheastCon 2017**. [S.l.], 2017. p. 1–7.
- PARK, J.; JOO, K.; HU, Z.; LIU, C.-K.; KWEON, I. S. Non-local spatial propagation network for depth completion. **arXiv preprint arXiv:2007.10042**, 2020.
- PASZKE, A.; GROSS, S.; MASSA, F.; LERER, A.; BRADBURY, J.; CHANAN, G.; KILLEEN, T.; LIN, Z.; GIMELSHEIN, N.; ANTIGA, L. et al. Pytorch: An imperative style, high-performance deep learning library. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2019. p. 8026–8037.
- PELUSO, V.; CIPOLLETTA, A.; CALIMERA, A.; POGGI, M.; TOSI, F.; MATTOCCIA, S. Enabling energy-efficient unsupervised monocular depth estimation on armv7-based platforms. In: IEEE. **2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)**. [S.l.], 2019. p. 1703–1708.
- PIASCO, N.; SIDIBÉ, D.; DEMONCEAUX, C.; GOUET-BRUNET, V. Perspective-n-learned-point: Pose estimation from relative depth. In: . [S.l.: s.n.], 2019.
- PILLAI, S.; AMBRUŞ, R.; GAIDON, A. Superdepth: Self-supervised, super-resolved monocular depth estimation. In: IEEE. **2019 International Conference on Robotics and Automation (ICRA)**. [S.l.], 2019. p. 9250–9256.
- PILZER, A.; LATHUILIERE, S.; SEBE, N.; RICCI, E. Refine and distill: Exploiting cycle-inconsistency and knowledge distillation for unsupervised monocular depth estimation. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2019. p. 9768–9777.
- PILZER, A.; XU, D.; PUSCAS, M.; RICCI, E.; SEBE, N. Unsupervised adversarial depth estimation using cycled generative networks. In: IEEE. **2018 International Conference on 3D Vision (3DV)**. [S.l.], 2018. p. 587–595.
- POGGI, M.; ALEOTTI, F.; TOSI, F.; MATTOCCIA, S. Towards real-time unsupervised monocular depth estimation on cpu. In: IEEE. **2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.], 2018. p. 5848–5854.
- POGGI, M.; ALEOTTI, F.; TOSI, F.; MATTOCCIA, S. On the uncertainty of self-supervised monocular depth estimation. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2020. p. 3227–3237.

- POGGI, M.; TOSI, F.; MATTOCCIA, S. Learning monocular depth estimation with unsupervised trinocular assumptions. In: IEEE. **2018 International Conference on 3D Vision (3DV)**. [S.l.], 2018. p. 324–333.
- PONTI, M. A.; COSTA, G. B. P. da. Como funciona o deep learning. **arXiv preprint arXiv:1806.07908**, 2018.
- QI, X.; LIAO, R.; LIU, Z.; URTASUN, R.; JIA, J. Geonet: Geometric neural network for joint depth and surface normal estimation. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 283–291.
- QIAN, N. On the momentum term in gradient descent learning algorithms. **Neural networks**, Elsevier, v. 12, n. 1, p. 145–151, 1999.
- QIU, J.; CUI, Z.; ZHANG, Y.; ZHANG, X.; LIU, S.; ZENG, B.; POLLEFEYS, M. Deeplidar: Deep surface normal guided depth prediction for outdoor scene from sparse lidar data and single color image. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2019. p. 3313–3322.
- RAMAMONJISOA, M.; LEPETIT, V. Sharpnet: Fast and accurate recovery of occluding contours in monocular depth estimation. In: **Proceedings of the IEEE International Conference on Computer Vision Workshops**. [S.l.: s.n.], 2019. p. 0–0.
- RAMIREZ, P. Z.; POGGI, M.; TOSI, F.; MATTOCCIA, S.; STEFANO, L. D. Geometry meets semantics for semi-supervised monocular depth estimation. In: SPRINGER. **Asian Conference on Computer Vision**. [S.l.], 2018. p. 298–313.
- RANFTL, R.; BOCHKOVSKIY, A.; KOLTUN, V. Vision transformers for dense prediction. **arXiv preprint arXiv:2103.13413**, 2021.
- RANJAN, A.; JAMPANI, V.; BALLE, L.; KIM, K.; SUN, D.; WULFF, J.; BLACK, M. J. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2019. p. 12240–12249.
- REN, X.; BO, L.; FOX, D. Rgb-(d) scene labeling: Features and algorithms. In: IEEE. **2012 IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.], 2012. p. 2759–2766.
- RIBEIRO, E. G.; GRASSI, V. Fast convolutional neural network for real-time robotic grasp detection. In: IEEE. **2019 19th International Conference on Advanced Robotics (ICAR)**. [S.l.], 2019. p. 49–54.
- RIBEIRO, E. G.; MENDES, R. de Q.; JR, V. G. Real-time deep learning approach to visual servo control and grasp detection for autonomous robotic manipulation. **Robotics and Autonomous Systems**, Elsevier, v. 139, p. 103757, 2021.
- ROBBINS, H.; MONRO, S. A stochastic approximation method. **The annals of mathematical statistics**, JSTOR, p. 400–407, 1951.
- ROBERTS, L. G. **Machine perception of three-dimensional solids**. 1963. Tese (Doutorado) — Massachusetts Institute of Technology, 1963.

- RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: SPRINGER. **International Conference on Medical image computing and computer-assisted intervention**. [S.l.], 2015. p. 234–241.
- ROSA, N. dos S.; GUIZILINI, V.; GRASSI, V. Sparse-to-continuous: Enhancing monocular depth estimation using occupancy maps. In: IEEE. **2019 19th International Conference on Advanced Robotics (ICAR)**. [S.l.], 2019. p. 793–800.
- ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, American Psychological Association, v. 65, n. 6, p. 386, 1958.
- ROY, A.; TODOROVIC, S. Monocular depth estimation using neural regression forest. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 5506–5514.
- RUBLEE, E.; RABAUD, V.; KONOLIGE, K.; BRADSKI, G. Orb: An efficient alternative to sift or surf. In: IEEE. **2011 International conference on computer vision**. [S.l.], 2011. p. 2564–2571.
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. **Learning internal representations by error propagation**. [S.l.], 1985.
- RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATY, A.; KHOSLA, A.; BERNSTEIN, M. et al. Imagenet large scale visual recognition challenge. **International journal of computer vision**, Springer, v. 115, n. 3, p. 211–252, 2015.
- RUSSELL, S. J.; NORVIG, P. **Artificial intelligence: a modern approach**. [S.l.]: Malaysia; Pearson Education Limited,, 2016.
- SALAS, M.; LATIF, Y.; REID, I. D.; MONTIEL, J. Trajectory alignment and evaluation in slam: Horns method vs alignment on the manifold. In: **Robotics: Science and Systems Workshop: The problem of mobile sensors**. [S.l.: s.n.], 2015.
- SANDLER, M.; HOWARD, A.; ZHU, M.; ZHMOGINOV, A.; CHEN, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2018. p. 4510–4520.
- SAXENA, A.; CHUNG, S. H.; NG, A. Y. Learning depth from single monocular images. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2006. p. 1161–1168.
- SAXENA, A.; SCHULTE, J.; NG, A. Y. et al. Depth estimation using monocular and stereo cues. In: **IJCAI**. [S.l.: s.n.], 2007. v. 7, p. 2197–2203.
- SAXENA, A.; SUN, M.; NG, A. Y. Learning 3-d scene structure from a single still image. In: IEEE. **2007 IEEE 11th International Conference on Computer Vision**. [S.l.], 2007. p. 1–8.
- SAXENA, A.; SUN, M.; NG, A. Y. Make3d: Depth perception from a single still image. In: **AAAI**. [S.l.: s.n.], 2008. v. 3, p. 1571–1576.

- SCHALKOFF, R. J. **Artificial intelligence: an engineering approach**. [S.l.]: McGraw-Hill New York, 1990.
- SCHNEIDER, N.; SCHNEIDER, L.; PINGGERA, P.; FRANKE, U.; POLLEFEYS, M.; STILLER, C. Semantically guided depth upsampling. In: SPRINGER. **German conference on pattern recognition**. [S.l.], 2016. p. 37–48.
- SCHWING, A. G.; FIDLER, S.; POLLEFEYS, M.; URTASUN, R. Box in the box: Joint 3d layout and object reasoning from single images. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2013. p. 353–360.
- SCHWING, A. G.; URTASUN, R. Efficient exact inference for 3d indoor scene understanding. In: SPRINGER. **European Conference on Computer Vision**. [S.l.], 2012. p. 299–313.
- SEIDE, F.; AGARWAL, A. Cntk: Microsoft’s open-source deep-learning toolkit. In: **Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. [S.l.: s.n.], 2016. p. 2135–2135.
- SERMANET, P.; EIGEN, D.; ZHANG, X.; MATHIEU, M.; FERGUS, R.; LECUN, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. **arXiv preprint arXiv:1312.6229**, 2013.
- SHANKARANARAYANA, S. M.; RAM, K.; MITRA, K.; SIVAPRAKASAM, M. Fully convolutional networks for monocular retinal depth estimation and optic disc-cup segmentation. **IEEE journal of biomedical and health informatics**, IEEE, v. 23, n. 4, p. 1417–1426, 2019.
- SHARMA, A.; VENTURA, J. Unsupervised learning of depth and ego-motion from cylindrical panoramic video. **arXiv preprint arXiv:1901.00979**, 2019.
- SHEN, J.; CHEUNG, S.-C. S. Layer depth denoising and completion for structured-light rgb-d cameras. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2013. p. 1187–1194.
- SHIVAKUMAR, S. S.; NGUYEN, T.; MILLER, I. D.; CHEN, S. W.; KUMAR, V.; TAYLOR, C. J. Dfusenet: Deep fusion of rgb and sparse depth information for image guided dense depth completion. In: IEEE. **2019 IEEE Intelligent Transportation Systems Conference (ITSC)**. [S.l.], 2019. p. 13–20.
- SILBERMAN, N.; FERGUS, R. Indoor scene segmentation using a structured light sensor. In: IEEE. **2011 IEEE international conference on computer vision workshops (ICCV workshops)**. [S.l.], 2011. p. 601–608.
- SILBERMAN, N.; HOIEM, D.; KOHLI, P.; FERGUS, R. Indoor segmentation and support inference from rgb-d images. In: SPRINGER. **European conference on computer vision**. [S.l.], 2012. p. 746–760.
- SILVA, I. N. D.; SPATTI, D. H.; FLAUZINO, R. A. Redes neurais artificiais para engenharia e ciências aplicadas curso prático. **São Paulo: Artliber**, 2010.
- SONG, S.; LICHTENBERG, S. P.; XIAO, J. Sun rgb-d: A rgb-d scene understanding benchmark suite. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 567–576.

SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. **The Journal of Machine Learning Research**, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014.

SU, W.; ZHANG, H.; ZHOU, Q.; YANG, W.; WANG, Z. Monocular depth estimation using information exchange network. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, 2020.

SUGIHARA, K. **Machine interpretation of line drawings**. [S.l.]: MIT press Cambridge, 1986. v. 1.

SUN, D.; ROTH, S.; BLACK, M. J. A quantitative analysis of current practices in optical flow estimation and the principles behind them. **International Journal of Computer Vision**, Springer, v. 106, n. 2, p. 115–137, 2014.

SUTSKEVER, I.; MARTENS, J.; DAHL, G.; HINTON, G. On the importance of initialization and momentum in deep learning. In: NSF. **International conference on machine learning**. Atlanta, USA, 2013. p. 1139–1147.

SUTSKEVER, I.; VINYALS, O.; LE, Q. V. Sequence to sequence learning with neural networks. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2014. p. 3104–3112.

SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; RABINOVICH, A. Going deeper with convolutions. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 1–9.

SZELISKI, R.; SCHARSTEIN, D. Symmetric sub-pixel stereo matching. In: SPRINGER. **European Conference on Computer Vision**. [S.l.], 2002. p. 525–540.

TAN, M.; LE, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In: PMLR. **International Conference on Machine Learning**. [S.l.], 2019. p. 6105–6114.

TANG, C.; TAN, P. Ba-net: Dense bundle adjustment network. **arXiv preprint arXiv:1806.04807**, 2018.

TANG, J.; TIAN, F.-P.; FENG, W.; LI, J.; TAN, P. Learning guided convolutional network for depth completion. **arXiv preprint arXiv:1908.01238**, 2019.

TATENO, K.; TOMBARI, F.; LAINA, I.; NAVAB, N. Cnn-slam: Real-time dense monocular slam with learned depth prediction. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2017. p. 6243–6252.

TEED, Z.; DENG, J. Deepv2d: Video to depth with differentiable structure from motion. **arXiv preprint arXiv:1812.04605**, 2018.

TIELEMAN, T.; HINTON, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. **COURSERA: Neural networks for machine learning**, v. 4, n. 2, p. 26–31, 2012.

- TOMPSON, J. J.; JAIN, A.; LECUN, Y.; BREGLER, C. Joint training of a convolutional network and a graphical model for human pose estimation. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2014. p. 1799–1807.
- TOSI, F.; ALEOTTI, F.; POGGI, M.; MATTOCCIA, S. Learning monocular depth estimation infusing traditional stereo knowledge. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2019. p. 9799–9809.
- TRUCCO, E.; VERRI, A. **Introductory techniques for 3-D computer vision**. [S.l.]: Prentice Hall Englewood Cliffs, 1998. v. 201.
- TURING, A. Mind. **Mind**, v. 59, n. 236, p. 433–460, 1950.
- UHRIG, J.; SCHNEIDER, N.; SCHNEIDER, L.; FRANKE, U.; BROX, T.; GEIGER, A. Sparsity invariant cnns. In: IEEE. **2017 International Conference on 3D Vision (3DV)**. [S.l.], 2017. p. 11–20.
- UMMENHOFER, B.; ZHOU, H.; UHRIG, J.; MAYER, N.; ILG, E.; DOSOVITSKIY, A.; BROX, T. Demon: Depth and motion network for learning monocular stereo. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2017. p. 5038–5047.
- VASILJEVIC, I.; KOLKIN, N.; ZHANG, S.; LUO, R.; WANG, H.; DAI, F. Z.; DANIELE, A. F.; MOSTAJABI, M.; BASART, S.; WALTER, M. R. et al. Diode: A dense indoor and outdoor depth dataset. **arXiv preprint arXiv:1908.00463**, 2019.
- VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, L.; POLOSUKHIN, I. Attention is all you need. **arXiv preprint arXiv:1706.03762**, 2017.
- WALTZ, D. Understanding line drawings of scenes with shadows. In: CITESEER. **The psychology of computer vision**. [S.l.], 1975.
- WANG, B.; AN, J. Fis-nets: Full-image supervised networks for monocular depth estimation. **arXiv preprint arXiv:2001.11092**, 2020.
- WANG, C.; BUENAPOSADA, J. M.; ZHU, R.; LUCEY, S. Learning depth from monocular videos using direct methods. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 2022–2030.
- WANG, L.; ZHANG, J.; WANG, O.; LIN, Z.; LU, H. Sdc-depth: Semantic divide-and-conquer network for monocular depth estimation. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2020. p. 541–550.
- WANG, P.; SHEN, X.; LIN, Z.; COHEN, S.; PRICE, B.; YUILLE, A. L. Towards unified depth and semantic prediction from a single image. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2015. p. 2800–2809.
- WANG, P.; SHEN, X.; RUSSELL, B.; COHEN, S.; PRICE, B.; YUILLE, A. L. Surge: Surface regularized geometry estimation from a single image. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2016. p. 172–180.

- WANG, R.; GERAGHTY, D.; MATZEN, K.; SZELISKI, R.; FRAHM, J.-M. Vplnet: Deep single view normal estimation with vanishing points and lines. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2020. p. 689–698.
- WANG, R.; PIZER, S. M.; FRAHM, J.-M. Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2019. p. 5555–5564.
- WANG, R.; SCHWORER, M.; CREMERS, D. Stereo dso: Large-scale direct sparse visual odometry with stereo cameras. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2017. p. 3903–3911.
- WANG, X.; FOUHEY, D.; GUPTA, A. Designing deep networks for surface normal estimation. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2015. p. 539–547.
- WANG, Y.; SUN, Y.; LIU, Z.; SARMA, S. E.; BRONSTEIN, M. M.; SOLOMON, J. M. Dynamic graph cnn for learning on point clouds. **Acm Transactions On Graphics (tog)**, ACM New York, NY, USA, v. 38, n. 5, p. 1–12, 2019.
- WARDE-FARLEY, D.; GOODFELLOW, I. J.; COURVILLE, A.; BENGIO, Y. An empirical analysis of dropout in piecewise linear networks. **arXiv preprint arXiv:1312.6197**, 2013.
- WATSON, J.; FIRMAN, M.; BROSTOW, G. J.; TURMUKHAMBETOV, D. Self-supervised monocular depth hints. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2019. p. 2162–2171.
- WIDROW, B.; HOFF, M. E. **Adaptive switching circuits**. [S.l.], 1960.
- WINSTON, P. H. **Artificial intelligence. 1992**. [S.l.]: Addison-Wesley, 1992.
- WOFK, D.; MA, F.; YANG, T.-J.; KARAMAN, S.; SZE, V. Fastdepth: Fast monocular depth estimation on embedded systems. In: IEEE. **2019 International Conference on Robotics and Automation (ICRA)**. [S.l.], 2019. p. 6101–6108.
- WONG, A.; FEI, X.; TSUEI, S.; SOATTO, S. Unsupervised depth completion from visual inertial odometry. **IEEE Robotics and Automation Letters**, IEEE, v. 5, n. 2, p. 1899–1906, 2020.
- WONG, A.; SOATTO, S. Bilateral cyclic constraint and adaptive regularization for unsupervised monocular depth prediction. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2019. p. 5644–5653.
- WU, Y.; BOOMINATHAN, V.; CHEN, H.; SANKARANARAYANAN, A.; VEERARAGHAVAN, A. Phasecam3d—learning phase masks for passive single view depth estimation. In: IEEE. **2019 IEEE International Conference on Computational Photography (ICCP)**. [S.l.], 2019. p. 1–12.
- WU, Y.; HE, K. Group normalization. In: **Proceedings of the European conference on computer vision (ECCV)**. [S.l.: s.n.], 2018. p. 3–19.

- XIANG, Y.; SAVARESE, S. Estimating the aspect layout of object categories. In: **IEEE. 2012 IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.], 2012. p. 3410–3417.
- XIAO, J.; OWENS, A.; TORRALBA, A. Sun3d: A database of big spaces reconstructed using sfm and object labels. In: **Proceedings of the IEEE international conference on computer vision**. [S.l.: s.n.], 2013. p. 1625–1632.
- XIAO, J.; RUSSELL, B.; TORRALBA, A. Localizing 3d cuboids in single-view images. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2012. p. 746–754.
- XIE, S.; GIRSHICK, R.; DOLLÁR, P.; TU, Z.; HE, K. Aggregated residual transformations for deep neural networks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 1492–1500.
- XINGJIAN, S.; CHEN, Z.; WANG, H.; YEUNG, D.-Y.; WONG, W.-K.; WOO, W.-c. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2015. p. 802–810.
- XIONG, H. Y.; ALIPANAHI, B.; LEE, L. J.; BRETSCHNEIDER, H.; MERICO, D.; YUEN, R. K.; HUA, Y.; GUEROUSSOV, S.; NAJAFABADI, H. S.; HUGHES, T. R. et al. The human splicing code reveals new insights into the genetic determinants of disease. **Science**, American Association for the Advancement of Science, v. 347, n. 6218, p. 1254806, 2015.
- XU, B.; WANG, N.; CHEN, T.; LI, M. Empirical evaluation of rectified activations in convolutional network. **arXiv preprint arXiv:1505.00853**, 2015.
- XU, D.; OUYANG, W.; WANG, X.; SEBE, N. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 675–684.
- XU, D.; RICCI, E.; OUYANG, W.; WANG, X.; SEBE, N. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2017. p. 5354–5362.
- XU, D.; WANG, W.; TANG, H.; LIU, H.; SEBE, N.; RICCI, E. Structured attention guided convolutional neural fields for monocular depth estimation. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 3917–3925.
- XU, Y.; ZHU, X.; SHI, J.; ZHANG, G.; BAO, H.; LI, H. Depth completion from sparse lidar data with depth-normal constraints. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2019. p. 2811–2820.
- XU, Z.; WANG, Y.; YAO, J. Deformable spatial propagation network for depth completion. **arXiv preprint arXiv:2007.04251**, 2020.
- YAN, L.; LIU, K.; BELYAEV, E. Revisiting sparsity invariant convolution: A network for image guided depth completion. **IEEE Access**, IEEE, 2020.

- YAN, Z.; WANG, K.; LI, X.; ZHANG, Z.; XU, B.; LI, J.; YANG, J. Rignet: Repetitive image guided network for depth completion. **arXiv preprint arXiv:2107.13802**, 2021.
- YANG, G.; TANG, H.; DING, M.; SEBE, N.; RICCI, E. Transformers solve the limited receptive field for monocular depth prediction. **arXiv preprint arXiv:2103.12091**, 2021.
- YANG, M.; YU, K.; ZHANG, C.; LI, Z.; YANG, K. Denseaspp for semantic segmentation in street scenes. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 3684–3692.
- YANG, N.; WANG, R.; STUCKLER, J.; CREMERS, D. Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In: **Proceedings of the European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2018. p. 817–833.
- YANG, X.; GAO, Y.; LUO, H.; LIAO, C.; CHENG, K.-T. Bayesian denet: Monocular depth prediction and frame-wise fusion with synchronized uncertainty. **IEEE Transactions on Multimedia**, IEEE, v. 21, n. 11, p. 2701–2713, 2019.
- YANG, Y.; WONG, A.; SOATTO, S. Dense depth posterior (ddp) from single image and sparse range. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2019. p. 3353–3362.
- YANG, Z.; WANG, P.; WANG, Y.; XU, W.; NEVATIA, R. Lego: Learning edge with geometry all at once by watching videos. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 225–234.
- YANG, Z.; WANG, P.; XU, W.; ZHAO, L.; NEVATIA, R. Unsupervised learning of geometry with edge-aware depth-normal consistency. **arXiv preprint arXiv:1711.03665**, 2017.
- YANG, Z.; WANG, P.; XU, W.; ZHAO, L.; NEVATIA, R. Unsupervised learning of geometry from videos with edge-aware depth-normal consistency. In: **Thirty-Second AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2018.
- YIN, W.; LIU, Y.; SHEN, C.; YAN, Y. Enforcing geometric constraints of virtual normal for depth prediction. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2019. p. 5684–5693.
- YIN, X.; WANG, X.; DU, X.; CHEN, Q. Scale recovery for monocular visual odometry using depth estimated with deep convolutional neural fields. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2017. p. 5870–5878.
- YIN, Z.; SHI, J. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 1983–1992.
- YU, D.; DENG, L.; DAHL, G. Roles of pre-training and fine-tuning in context-dependent dbn-hmms for real-world speech recognition. In: **Proc. NIPS Workshop on Deep Learning and Unsupervised Feature Learning**. [S.l.: s.n.], 2010.

- YU, L.-F.; YEUNG, S.-K.; TAI, Y.-W.; LIN, S. Shading-based shape refinement of rgb-d images. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2013. p. 1415–1422.
- YUAN, Y.; CHEN, X.; WANG, J. Object-contextual representations for semantic segmentation. **arXiv preprint arXiv:1909.11065**, 2019.
- ZEILER, M. D. Adadelta: an adaptive learning rate method. **arXiv preprint arXiv:1212.5701**, 2012.
- ZENG, J.; TONG, Y.; HUANG, Y.; YAN, Q.; SUN, W.; CHEN, J.; WANG, Y. Deep surface normal estimation with hierarchical rgb-d fusion. In: **Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2019. p. 6153–6162.
- ZHAN, H.; GARG, R.; WEERASEKERA, C. S.; LI, K.; AGARWAL, H.; REID, I. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 340–349.
- ZHANG, G.; LI, H. Effectiveness of scaled exponentially-regularized linear units (serlus). **arXiv preprint arXiv:1807.10117**, 2018.
- ZHANG, H.-T.; YU, J.; WANG, Z.-F. Probability contour guided depth map inpainting and superresolution using non-local total generalized variation. **Multimedia Tools and Applications**, Springer, v. 77, n. 7, p. 9003–9020, 2018.
- ZHANG, J.; LI, W.; GOU, H.; FANG, L.; YANG, R. Lead: Lidar extender for autonomous driving. **arXiv preprint arXiv:2102.07989**, 2021.
- ZHANG, R.; TSAI, P.-S.; CRYER, J. E.; SHAH, M. Shape-from-shading: a survey. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 21, n. 8, p. 690–706, 1999.
- ZHANG, X.; WU, R. Fast depth image denoising and enhancement using a deep convolutional network. In: IEEE. **2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.], 2016. p. 2499–2503.
- ZHANG, Y.; FUNKHOUSER, T. Deep depth completion of a single rgb-d image. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2018. p. 175–185.
- ZHANG, Y.; NGUYEN, T.; MILLER, I. D.; SHIVAKUMAR, S. S.; CHEN, S.; TAYLOR, C. J.; KUMAR, V. Dfnet: Ego-motion estimation and depth refinement from sparse, noisy depth input with rgb guidance. **arXiv preprint arXiv:1903.06397**, 2019.
- ZHANG, Z.; CUI, Z.; XU, C.; JIE, Z.; LI, X.; YANG, J. Joint task-recursive learning for semantic segmentation and depth estimation. In: **Proceedings of the European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2018. p. 235–251.
- ZHANG, Z.; CUI, Z.; XU, C.; YAN, Y.; SEBE, N.; YANG, J. Pattern-affinitive propagation across depth, surface normal and semantic segmentation. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2019. p. 4106–4115.

- ZHANG, Z.; XU, C.; YANG, J.; TAI, Y.; CHEN, L. Deep hierarchical guidance and regularization learning for end-to-end depth estimation. **Pattern Recognition**, Elsevier, v. 83, p. 430–442, 2018.
- ZHANG, Z.; XU, C.; YANG, J.; GAO, J.; CUI, Z. Progressive hard-mining network for monocular depth estimation. **IEEE Transactions on Image Processing**, IEEE, v. 27, n. 8, p. 3691–3702, 2018.
- ZHAO, Y.; ZHU, S.-C. Scene parsing by integrating function, geometry and appearance models. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2013. p. 3119–3126.
- ZHOU, J.; WANG, Y.; QIN, K.; ZENG, W. Unsupervised high-resolution depth learning from videos with dual networks. In: **Proceedings of the IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2019. p. 6872–6881.
- ZHOU, T.; BROWN, M.; SNAVELY, N.; LOWE, D. G. Unsupervised learning of depth and ego-motion from video. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**. [S.l.: s.n.], 2017. p. 1851–1858.
- ZHU, J.; SHI, Y.; REN, M.; FANG, Y.; LIEN, K.-C.; GU, J. Structure-attentioned memory network for monocular depth estimation. **arXiv preprint arXiv:1909.04594**, 2019.
- ZOPH, B.; VASUDEVAN, V.; SHLENS, J.; LE, Q. V. Learning transferable architectures for scalable image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2018. p. 8697–8710.
- ZOU, Y.; LUO, Z.; HUANG, J.-B. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In: **Proceedings of the European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2018. p. 36–53.
- ZWALD, L.; LAMBERT-LACROIX, S. The berhu penalty and the grouped effect. **arXiv preprint arXiv:1207.6868**, 2012.