BRENO BATISTA DA SILVA

CRIAÇÃO DE UM MODELO DE REDE PROFUNDA PARA CALCULAR A SIMILARIDADE DE MAPAS DE PROFUNDIDADE COM IMAGENS MONOCULARES

ALEGRE

2018

CRIAÇÃO DE UM MODELO DE REDE PROFUNDA PARA CALCULAR A SIMILARIDADE DE MAPAS DE PROFUNDIDADE COM IMAGENS MONOCULARES

Trabalho apresentado como requisito parcial à obtenção do grau de Bacharel em Ciência da Computação, pela Universidade Federal do Espírito Santo.

por

BRENO BATISTA DA SILVA

Professor
Jacson Rodrigues Correia da Silva

Universidade Federal do Espírito Santo

ALEGRE

2018

Termo de Aprovação

BRENO BATISTA DA SILVA

CRIAÇÃO DE UM MODELO DE REDE PROFUNDA PARA CALCULAR A SIMILARIDADE DE MAPAS DE PROFUNDIDADE COM IMAGENS MONOCULARES

Trabalho aprovado como requisito parcial à obtenção do grau de Bacharel em Ciência da Computação, pela Universidade Federal do Espírito Santo, pela seguinte banca examinadora:

Prof. M. Sc. Jacson Rodrigues Correia da Silva UFES	Profa. Dra. Larice Nogueira de Andrade UFES

ALEGRE, 10 DE SETEMBRO DE 2018

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

ANN	Artificial Neural Network
DNN	Deep Neural Network
CNN	Convolutional Neural Network
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
API	Application Programming Interface

RESUMO

Elementos tridimensionais estão cada vez mais presentes em nosso meio, sendo largamente utilizados em jogos, robótica, indústria cinematográfica e arquitetura. Por este motivo, a comunidade tem aplicado muito esforço na reconstrução 3D, com diversas abordagens e algoritmos propostos para resolver o problema. Os algoritmos de reconstrução 3D precisam conhecer a distância dos objetos de uma imagem, por isso, gerar um mapa de profundidade de qualidade é de extrema importância.

Mapas de profundidade apresentam a distância de cada pixel de um objeto no mundo real a seu observador. Estes mapas são criados por diversos métodos, onde podem ser utilizadas, em geral, duas ou mais imagens (abordagem estéreo) ou uma única imagem (abordagem monocular). A abordagem monocular é um processo mais complexo, porque fornece uma quantidade menor de informações ao algoritmo de reconstrução utilizado. Através das Redes Neurais Convolucionais, a comunidade vem evoluindo a geração de mapas de profundidade, tanto em qualidade (mapa mais próximo ao real) quanto em tempo de treinamento.

Este projeto propõe a construção de um modelo de Rede Neural Profunda para mensurar a similaridade de uma imagem monocular e de seu mapa de profundidade, fornecendo uma forma de avaliação para outros trabalhos que efetuam a geração desses mapas de profundidade e que necessitam de comparar a similaridade de seus resultados, permitindo verificar se é possível melhorar seu comportamento.

Palavras-chave: rede neural profunda; rede neural convolucional; rede siamesa; mapa de profundidade; imagem monocular.

1 INTRODUÇÃO

A reconstrução 3D de cenas pode ser realizada de forma ativa, através de *hardware* especializado ou de forma passiva, através de algoritmos e imagens providas de câmeras comuns.

Para os métodos de reconstrução 3D passiva é essencial a geração de um mapa de profundidade fiel à realidade da cena. Estes mapas, geralmente em escala de cinza, apresentam a distância de cada pixel entre a câmera e o objeto no mundo real [1] e para sua criação são utilizadas, em geral, duas ou mais imagens (abordagem estéreo), onde é possível utilizar o cálculo de disparidade, fluxo óptico e/ou *parallax* [2].

Alguns métodos se propõem a gerar um mapa de profundidade a partir de uma única imagem (abordagem monocular) porque, apesar de relevante, devido a maior facilidade e menor custo para obter a imagem em relação a outra abordagem, tem um problema proporcionalmente mais complexo, por conta da quantidade menor de informações que uma única imagem fornece.

Com o auxílio do Aprendizado de Máquina, a Visão Computacional tem evoluído continuamente, principalmente com o uso das Redes Neurais Convolucionais que superam os métodos tradicionais para vários problemas, como: classificação de imagens [3], segmentação [4] e construção de mapas de profundidade na abordagem estéreo e monocular [5–7].

Dado os bons resultados apresentados por Redes Neurais Convolucionais na geração de mapas de profundidade com imagens estéreo e a necessidade desses mapas em diversas áreas, um modelo de Rede Neural capaz de mensurar a similaridade de uma imagem monocular e seu mapa de profundidade poderá ser utilizado como função de avaliação para métodos novos ou existentes, permitindo verificar o quanto se aproximaram da profundidade de imagem.

Dentre as Redes Neurais Convolucionais, as Redes Neurais Siamesas têm apresentado um bom resultado na obtenção de similaridade de imagens [8, 9], destacando-se

assim como um bom modelo de Rede Neural para ser explorado. Por isso, o presente projeto tem como hipótese que será possível gerar um modelo de Rede Neural Profunda capaz de fornecer a similaridade entre uma imagem monocular e seu mapa de profundidade. A similaridade refere-se ao quanto, em probabilidade, o mapa de profundidade refere-se à imagem monocular apresentada.

Este projeto está estruturado da seguinte forma: no restante deste capítulo temos uma descrição do problema, sua importância e os objetivos gerais e específicos deste projeto; em seguida, o Capítulo 2 descreve os métodos que serão objetos de estudo durante o projeto e os trabalhos relacionados; no Capítulo 3 uma metodologia para desenvolvimento deste projeto é apresentada; e no Capítulo 4 são apresentados quais os resultados esperados com o desenvolvimento deste projeto. Então, são apresentadas as referências e o cronograma para desenvolvimento deste trabalho.

1.1 O problema e a sua importância

Aplicações como reconstrução 3D de cenas e navegação de carros autônomos necessitam da informação de distância do observador até os objetos da imagem. Esta tarefa, apesar de fácil para os humanos, é complexa para o computador, devido a forma como ele enxerga uma imagem (Figura 1).

(a) O que o humano vê

(b) O que o computador vê

Figura 1: Diferenças da visão humana para o computador.

Fonte: Imagem adaptada de
<https://adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/>.

Diversos trabalhos tentaram gerar essa informação de profundidade com precisão para aumentar o sucesso das aplicações que precisam dela. Uma forma de repre-

sentar essa informação são os mapas de profundidade que, geralmente em escala de cinza, apresentam a distância de cada pixel de um objeto no mundo real a seu observador.

A maioria dos trabalhos da literatura em cálculo do mapa de profundidade, utilizam duas ou mais imagens (visão estéreo), fazendo uso do cálculo de disparidade, fluxo óptico e/ou *parallax*. O grau de dificuldade cresce ao estimar a profundidade a partir de uma única imagem (monocular), devido a menor quantidade de informação disponível.

Alguns métodos tradicionais se propuseram a resolver este problema, utilizando conhecimento da estrutura global da imagem, pré-conhecimento da cena e/ou propagação de desfoque [10–12].

Entretanto, estes métodos ainda não conseguiram alcançar uma assertividade completa (*ground truth*), ou seja, a perfeita profundidade da cena. Assim, vários trabalhos buscam melhorar a forma de encontrar a profundidade das imagens. Uma abordagem que tem apresentado excelentes resultados, superando os métodos tradicionais, é a utilização de Redes Neurais Convolucionais [5–7].

Assim, a criação de um modelo de Rede Neural Profunda capaz de mensurar a similaridade de uma imagem monocular e de seu mapa de profundidade, proverá uma forma de avaliação para outros trabalhos que efetuam a geração desses mapas de profundidade. Através dessa função de avaliação mais precisa, este projeto poderá ser capaz de melhorar o desempenho do estado-da-arte.

1.2 Objetivos

1.2.1 Objetivo geral

Gerar um modelo de Rede Neural Profunda capaz de mensurar a similaridade entre uma imagem monocular e seu mapa de profundidade.

1.2.2 Objetivos específicos

1. Gerar conjuntos de dados com imagens monoculares e mapas de profundidade.
2. Testar e explorar modelos existentes de Redes Neurais Profundas sobre os conjuntos de dados gerados.

3. Gerar um novo modelo de Rede Neural para mensurar a assertividade entre a imagem monocular e seu mapa de profundidade.
4. Testar o modelo de Rede Neural com bases de dados.
5. Comparar os resultados com trabalhos disponíveis na literatura.

2 REFERENCIAL TEÓRICO E REVISÃO DE LITERATURA

A visão computacional tem expandido seus métodos de trabalho com reconstrução de cenas 3D através da utilização da aprendizagem de máquina. É fundamental para os algoritmos de reconstrução 3D conhecer a distância dos objetos de uma imagem, por isso, gerar um mapa de profundidade de qualidade é de extrema importância.

Neste capítulo, serão abordados os principais conceitos sobre visão computacional, mapa de profundidade e aprendizado de máquina que serão objetos de estudo durante o projeto, assim como, serão referenciados o estado-da-arte destes conceitos.

2.1 Visão Computacional

A visão computacional é estudada desde 1970 e seu objetivo é desenvolver aplicações e modelos matemáticos que possibilitem um computador a ver e entender uma cena através da recuperação do máximo de informações possíveis, como a forma tridimensional e a aparência dos objetos da cena. Esses modelos matemáticos abrangem diversas áreas, como física, geometria euclidiana e projetiva, estatística e otimização [13].

Os estudos em visão computacional são divididos em vários campos, como: formação da imagem, processamento de imagens, detecção e casamento de características, segmentação, estrutura a partir do movimento (*structure from motion*), estimativa de movimento e profundidade, fotografia computacional, alinhamento baseado em características, correspondência estéreo, reconstrução 3D, renderização baseada em imagem e reconhecimento [13].

Um tópico importante e amplamente estudado na área de visão computacional é a reconstrução 3D de cenas a partir de imagens e/ou vídeos digitais, devido ao grande número de aplicações possíveis, como por exemplo, na medicina, na ajuda a deficien-

tes visuais, robótica, navegação, indústria cinematográfica, videogames, realidade 3D e realidade aumentada [2].

2.1.1 Reconstrução 3D

A reconstrução 3D é o processo de captura da forma e aparência de objetos reais a partir de imagens e/ou vídeos. Uma forma de efetuar a reconstrução 3D de cenas é denominada de abordagem ativa, onde é utilizado *hardware* especializado, que trazem bons resultados, mas que apresentam problemas de não operar muito bem a luz do sol ou em um ambiente descontrolado. Além disso, um *hardware* especialista, como o *kinect* e o *scanner* a laser, são caros e inacessíveis para a maioria das pessoas.

Outra forma de reconstrução 3D é denominada de abordagem passiva, onde a reconstrução é feita de imagens providas de câmeras comuns. Ela possui algumas vantagens, pois permite criar modelos 3D de objetos que não existem mais e são mais acessíveis devido ao baixo custo das câmeras digitais, que podem ser encontradas, por exemplo, em qualquer celular [2].

Além disso, abordagem passiva abre a possibilidade de utilização de imagens disponíveis na Internet, como no caso do projeto *Building Rome in a Day* [14], que utilizou mais de 150 mil imagens do site flickr[1] junto a um algoritmo de reconstrução 3D em um *cluster* com 500 *cores* para reconstruir partes da cidade em menos de um dia, conforme a Figura 2.

Os métodos tradicionais existentes para a abordagem passiva são: *shape from stereo-vision*, *shape from motion*, *shape from silhouette*, *shape from photo-consistency* e *shape from defocus* [2]. Esses métodos não serão utilizados neste trabalho, entretanto, faz-se necessário abordar alguns para embasamento de conceitos chave e futura comparação com os resultados.

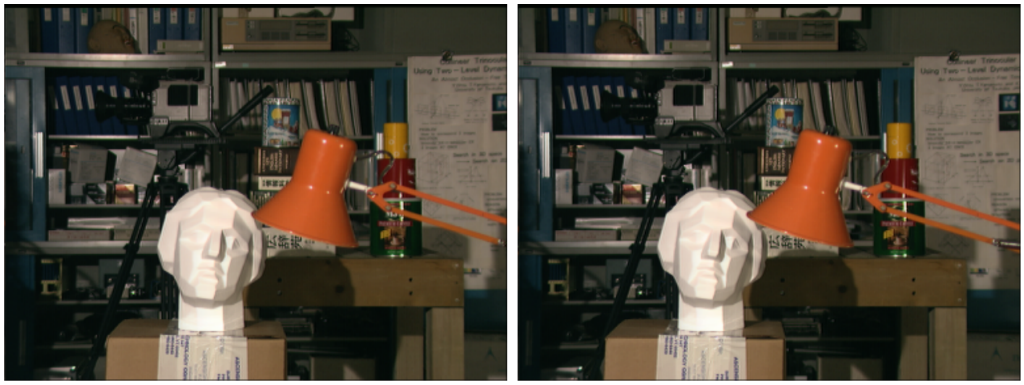
O ser humano compara as informações obtidas pelo olho esquerdo com o direito para determinar a diferença entre a profundidade relativa do objeto. Isso acontece porque os dois olhos possuem diferentes pontos de vista, conforme a Figura 3. Esse fenômeno é chamado de disparidade estereoscópica [15].

Através da forma como os humanos enxergam e processam as cenas, o *shape from stereo vision* tenta simular este processo, podendo ser definido como um método de cálculo de disparidade a partir de duas imagens de um sistema com duas câmeras. É

[1] Disponível em: <https://www.flickr.com>

Figura 2: Resultados do projeto Building Rome in a Day.
Fonte: Imagem adaptada de [14].

característico deste método que a distância das câmeras seja relativamente pequena e constante (as câmeras são conectadas por uma construção rígida e são calibradas quanto ao seu ponto de vista) e a aquisição das imagem ocorra no mesmo tempo para ambas as câmeras. Dessa forma, sabendo a distância de ponto de vista das duas imagens, é possível obter o mapa de profundidade através de alguns passos como: calibração das câmeras, estandardização das imagens, correspondência estéreo e a triangulação [2].



(a) Imagem pelo olho esquerdo (b) Imagem pelo olho direito

Figura 3: Disparidade da retina humana.
Fonte: Imagem adaptada de [16].

Já a abordagem do *shape from motion* trabalha com imagens obtidas a partir de câ-

meras diferentes e em diferentes posições, analisando as relações entre a posição dos pixels nas imagens, rotações e translações na câmera para calcular as coordenadas 3D da cena real. Os três principais métodos dessa abordagem são o *two-image algorithm*, o *factorisation* e o *bundle adjustment* [2]. Um excelente comparativo das técnicas baseadas em visão estéreo ou movimento pode ser encontrado em [17].

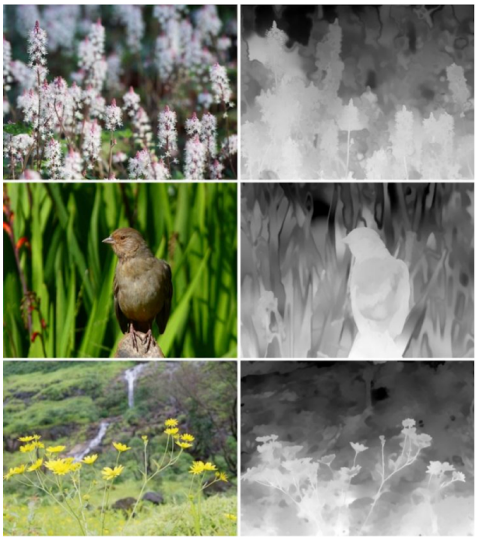


Figura 4: Imagem original e o mapa de profundidade estimado.
Fonte: Imagem adaptada de [11].

Para os algoritmos de reconstrução 3D de cenas é fundamental um mapa de profundidade, que é uma imagem em escalas de cinza que possui exatamente o mesmo tamanho da imagem original e indica a relativa ou aparente distância de cada pixel entre a câmera e o objeto no mundo real [1].

Para geração desses mapas são utilizados, normalmente, duas ou mais imagens (abordagem estéreo), como abordado pelos métodos *shape from stereo vision* e *shape from motion*, respectivamente e por outros trabalhos, como Seitz et al. [18]. Entretanto, alguns trabalhos se propõem a resolver o problema a partir de uma única imagem (abordagem monocular). Dentre eles, Zhuo e Sim [10] apresentou um método de desfocagem robusto e resistente a ruído, localização imprecisa das bordas e interferências das bordas vizinhas, capaz de gerar mapas de profundidade mais precisos em comparação com os métodos disponíveis na época. Akimov, Vatolin e Smirnov [11] construiu um método utilizando propagação de desfoque da imagem, alcançando um resultado superior ao trabalho de Zhuo e Sim [10], apresentado na Figura 4. Wang et al. [12] também teve uma contribuição, aprimorando o uso do desfoque introduzindo uma estratégia multi-escalar.

A literatura possui um extenso número de trabalhos que buscaram gerar mapas de profundidade, sendo superados por trabalhos que utilizam a Aprendizagem de Máquina, discutidos na subseção 2.2, e mesmo com essa evolução, a comunidade ainda busca atingir a geração exata (*ground truth*) do mapa de profundidade.

Uma forma de melhorar o resultado desses trabalhos, permitindo a construção de mapas de profundidade mais exatos, é melhorar o mecanismo de mensurar a similaridade entre o mapa de profundidade e a imagem de origem.

2.2 Aprendizado de Máquina

O aprendizado de máquina é um campo de pesquisa muito ativo atualmente, possuindo um grande número aplicações, como robótica, carros autônomos, *business intelligence*, mecanismos de busca, redes sociais, sistemas de recomendação e assistentes pessoais.

De acordo com Kohavi e Provost [19], a aprendizagem de máquina pode ser definida como o estudo e construção de algoritmos que podem aprender e fazer predições sobre um conjunto de dados. Segundo Russell e Norvig [20], existem três tipos principais de aprendizado que um sistema pode exercer, sendo eles: não supervisionado, supervisionado e por reforço.

No aprendizado não supervisionado, o objetivo é tentar estabelecer a existência de grupos ou similaridade nos dados (clusterização), uma vez que o agente aprende padrões nos exemplos de entrada mesmo sem ter nenhuma informação ou rótulo sobre a qual grupo aquele exemplo pertence [20].

Já no aprendizado supervisionado, o conjunto de exemplos de entrada são rotulados, isto é, já sabe-se a classe e/ou grupo que o exemplo pertence (Ex: $f(x_i) = y_j$, sendo x_i o i-ésimo exemplo do problema e y_j a classe j do problema). O objetivo é encontrar uma hipótese capaz de classificar novas observações entre as classes já existentes [20].

E no aprendizado por reforço, o sistema desenvolvido possui interações com o meio ambiente que o cerca e aprende uma política ótima de ação por experimentação direta com o meio. O agente é recompensado ou penalizado, dependendo da ação executada. Seu objetivo é desenvolver uma política que maximize a quantidade de recompensa recebida ao longo de sua execução [21].

Além desses, ainda existe o aprendizado semi-supervisionado, que é um meio termo entre o não supervisionado e o supervisionado, onde o sistema recebe dois conjuntos de exemplos de entrada, sendo um conjunto rotulado e o outro não [20].

O Aprendizado de Máquina possui diversos paradigmas já apresentados e estudados pela literatura, como o paradigma simbólico, estatístico, baseado em protótipo, conexionista e genético [21]. Porém, o escopo desse projeto é o paradigma conexionista, devido aos bons resultados alcançados sobre o problema.

Dentro da abordagem de aprendizado supervisionado e do paradigma conexionista se encontram as Redes Neurais Artificiais (ANN) que possuem como base modelos biológicos do sistema nervoso que se iniciaram com os estudos de McCulloch & Pitts [22].

Em 1962, Frank Rosenblatt apresentou o Perceptron [23] que é a base dos neurônios artificiais utilizados nos trabalhos atuais. Um perceptron recebe um conjunto de entradas binárias (x_1, x_2, x_3, ..., x_n) que, após serem multiplicadas pelos valores das arestas conectadas, definidos como pesos (w_1, w_2, w_3, ..., w_n), produz uma única saída (Figura 5).

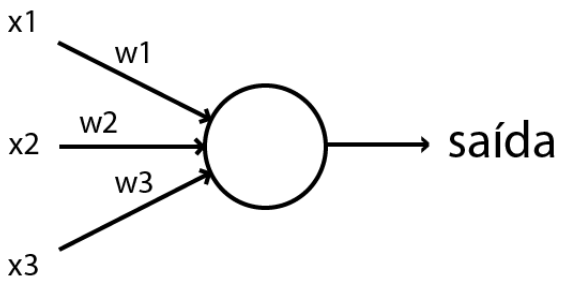


Figura 5: Ilustração do Perceptron.

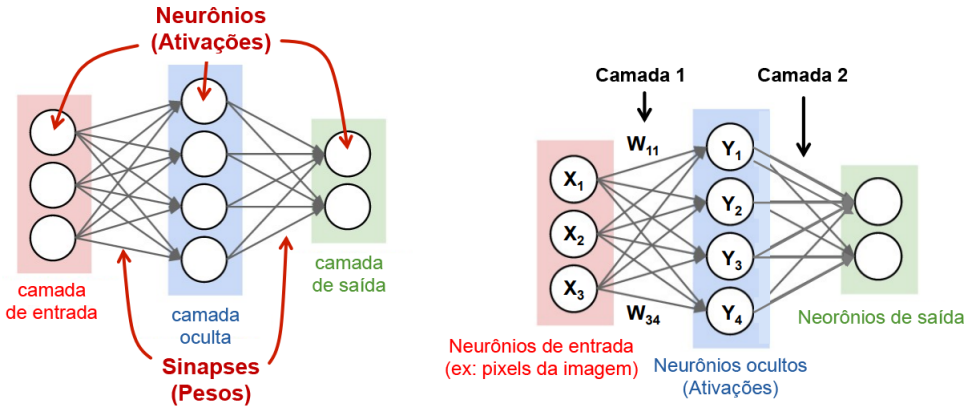
Fonte: Imagem adaptada de [24].

Rosenblatt propôs a correção dos pesos que representam a importância da respectiva entrada com a saída, ou seja, o conhecimento do neurônio. Segundo (Nielsen [24]), o resultado do neurônio é binário e determinado por: se a soma das entradas multiplicada com os pesos é menor ou igual, ou maior que um valor limitante (Equação 1).

Existem diversos modelos de Redes Neurais Artificiais, como a Rede Neural Recorrente [25] e a Rede Neural sem Peso [26], porém a mais utilizada é a Rede Neural *Feedforward*, que recebe esse nome porque a informação flui através das conexões das camadas da esquerda para direita, sem nenhuma conexão de retorno [27].

$$output = \begin{cases} 0 & if \ \sum_j w_j x_j \ \leq \ threshold \\ 1 & if \ \sum_j w_j x_j \ > \ threshold \end{cases} \tag{1}$$

Uma Rede Neural *Feedforward* são formadas por duas ou três camadas de neurônios (Figura 6), onde cada neurônio dessas camadas recebe entradas, processa essa informação, passa pela função de ativação e emite sua saída para os neurônios da próxima camada [27]. O aprendizado consiste na alteração dos parâmetros w_{ij} (pesos dos neurônios), objetivando minimizar a diferença entre o valor de saída e o valor esperado [28].



(a) Neurônios e sinapses

(b) Pesos das camadas

Figura 6: Arquitetura de uma Rede Neural Artificial.

Fonte: Imagem adaptada de [29].

Funções de ativação são um importante recurso das redes neurais artificiais, porque são elas que definem a saída do neurônio, ou seja, se a informação que o neurônio está recebendo é relevante [27]. Similar ao Perceptron, pode-se definir o processo de propagação da informação como a Equação 2, onde Y representa a saída do neurônio de acordo com a função de ativação F.

$$Y \ = \ F(\sum(peso \ * \ entrada)) \tag{2}$$

A função de ativação (F), apresentada na Equação 2, é uma transformação linear ou não linear que pode variar de acordo com a função escolhida. A Figura 7 mostra as principais funções de ativação não lineares encontradas na literatura, outras funções e mais informações em Goodfellow, Bengio e Courville [27].

Buscando aumentar a precisão e generalização das Redes Neurais Artificiais, pes-

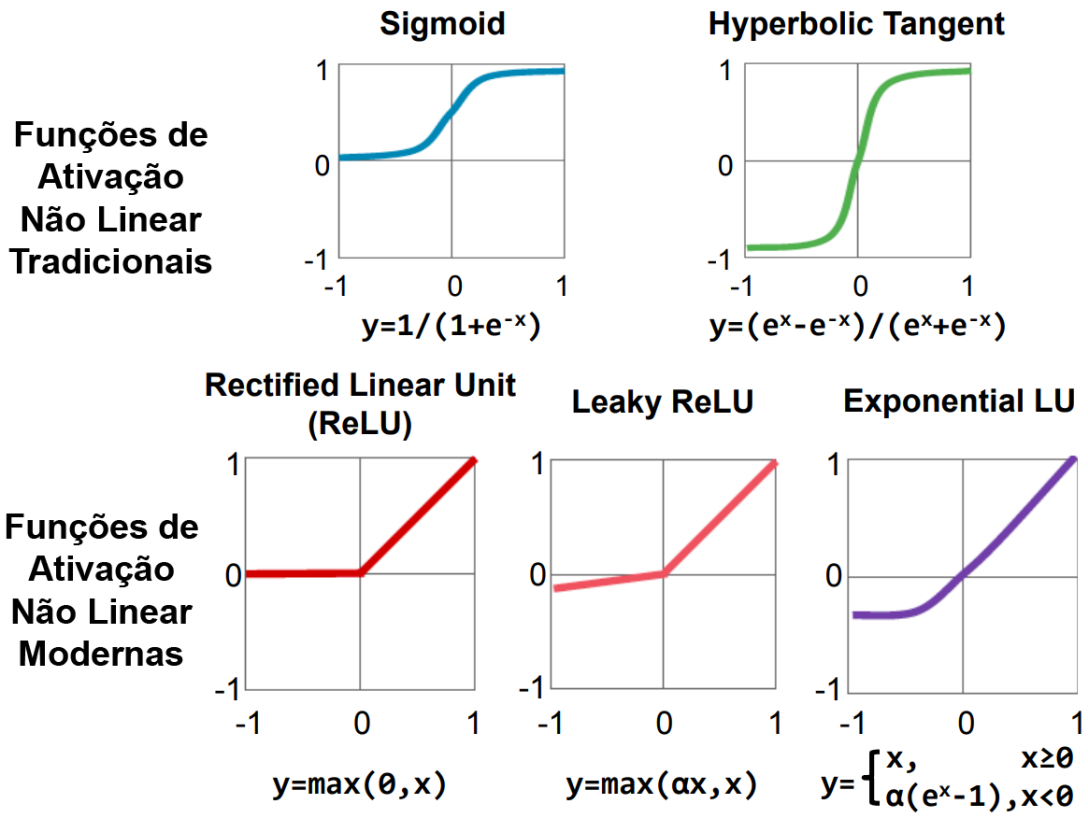


Figura 7: Funções de ativação não lineares.

Fonte: Imagem adaptada de [29].

quisadores tentaram adicionar mais camadas ocultas as redes, como [30]. Devido ao maior número de camadas ocultas, estas redes ficaram conhecidas como Redes Neurais Profundas (DNN).

As DNNs conseguem representar funções de alta complexidade [27] e um tipo específico delas, denominado Redes Neurais Convolucionais (CNN), são atualmente a base para muitas aplicações modernas de Inteligência Artificial, com aplicações em reconhecimento de fala, reconhecimento de imagem, carros autônomos, para detectar câncer e jogar jogos complexos [29].

2.2.1 Redes Neurais Convolucionais

A primeira CNN foi apresentada em 1989, por Yann Lecun [31], e utilizou a convolução, um tipo especializado de operação matemática, para reconhecer dígitos manuscritos. A chamada LeNet é reconhecida como a primeira Rede Neural Convolucional e foi aprimorada ao longo dos anos [32, 33].

A convolução permite que redes profundas aprendam funções em dados espaciais estruturados, como imagens, vídeo e texto. Matematicamente, as redes convolucionais fornecem cálculos para explorar efetivamente a estrutura local dos dados. Por exemplo, uma imagem é representada como uma matriz bidimensional de pixels e partes desta imagem que estão próximas umas das outras na matriz de pixels tendem a variar juntas. Desta forma, as Redes Neurais Convolucionais aprendem a explorar essa estrutura de covariância natural para aprender efetivamente e extrair o melhor conjunto de características das imagens [34].

Após o trabalho de LeCun, novos modelos de Redes Neurais Convolucionais foram apresentados e tiveram destaques em uma das competições mais importantes da área, o ILSVRC[2], como: AlexNet [35], que possui uma arquitetura parecida com a LeNet, porém mais profunda, maior e com mais camadas convolucionais; ZFNet [36], campeã do ILSVRC 2013, que aumentou a acurácia da AlexNet através do uso de mais camadas intermediárias; GoogleNet [3], campeã do ILSVRC 2014, que apresentou um novo módulo que reduziu drasticamente o número de parâmetros na rede; VGGNet [6], vice campeã do ILSVRC 2014, que demonstrou a importância da profundidade da rede em relação ao desempenho; ResNet [37], campeã do ILSVRC 2015,que apresentou novos tipos de conexões, de normalização e da estrutura final da rede; GDB-Net [38], campeã do ILSVRC 2016, que apresentou um *framework* composto de rede convolucionais; SENet [39], campeã do ILSVRC 2017, que propôs uma nova unidade na arquitetura da rede chamada de *Squeeze-and-Excitation* com o objetivo de recalibrar as características extraídas da convolução.

A arquitetura de uma CNN apresenta três camadas principais, conforme a Figura 8, comuns a todos os seus modelos, sendo: camada convolucional, camada *pooling* e camada totalmente conectada [27].

A camada convolucional utiliza o conceito de campo receptivo local dos neurônios, que é a parte da percepção sensorial do corpo humano que afeta o disparo do neurônio. Esse campo de visão pode corresponder a um pedaço de pele ou a um segmento do campo visual de uma pessoa [34]. Aplicado a imagens, cada campo receptivo corresponde a um pedaço de pixels, conforme a representação da Figura 9.

Para cada posicionamento do campo receptivo local, a camada convolucional aplica uma função não linear para retornar um único número correspondente aquele pedaço da imagem (Figura 10). Essa função não linear é a multiplicação do campo recep-

[2]Disponível em: <https://www.image-net.org/challenges/LSVRC/>

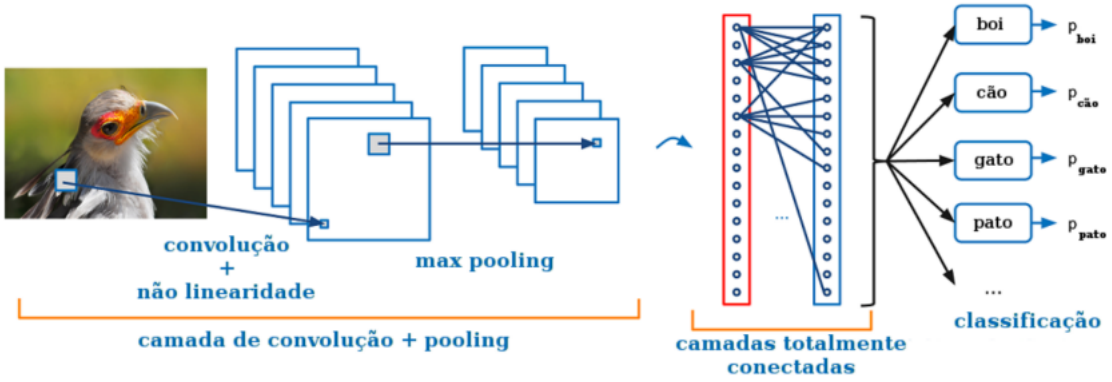


Figura 8: Camadas principais de uma CNN.

Fonte: Imagem adaptada de
<https://adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/>.

tivo por uma matriz chamada *kernel* convolucional e pode ser definida conforme a Equação 3, onde I é a imagem e K o *kernel* [34].

$$(K * I)(i,\ j)\ =\ \sum_{m}\sum_{n} K(m,\ n)\ I(i\ -\ m,\ j\ -\ n) \tag{3}$$

A camada de *pooling* recebe as características extraídas pela camada convolucional e executa uma transformação não linear para reduzir o tamanho destas características, juntamente com o custo computacional do treinamento da CNN [27, 34]. Uma operação utilizada, por exemplo, é o *max pooling*, definido na Equação 4, que fornece como saída o valor máximo de uma vizinhança retangular r do mapa de entrada, conforme a Figura 11.

$$Pooling(a,b) = max\{(K * I)(a^{'},b^{'}),\ a^{'}\ \epsilon\ [a,\ a+r],\ b^{'}\ \epsilon\ [b,\ b+r]\} \tag{4}$$

Após várias execuções das camadas convolucionais e *pooling* as características extraídas são passadas para uma rede neural totalmente conectada com uma ou mais camadas. Essa rede é chamada de totalmente conectada porque todos os neurônios de uma camada recebem os valores de todos os neurônios da camada anterior para gerar uma informação semântica [40].

Eigen, Puhrsch e Fergus [5] desenvolveram uma CNN com uma fase de refinamento, sendo capaz de calcular o mapa de profundidade a partir de imagens monoculares, atingindo melhor resultado que os métodos da época. Já Liu et al. [41] apresentou

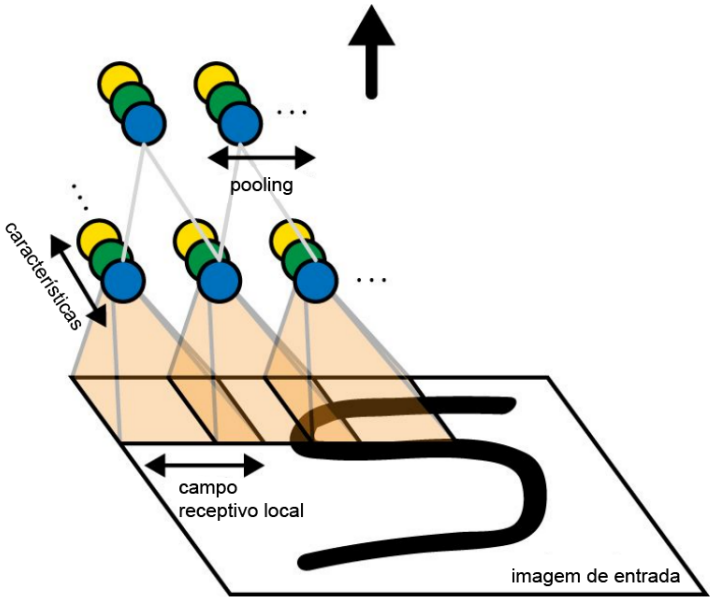


Figura 9: Campo receptivo local.

Fonte: Imagem adaptada de [34].

uma CNN que explorava campos condicionais aleatórios, superando o trabalho de Eigen. Cheng, Wang e Yang [7] propôs uma rede convolucional de propagação espacial alcançando o estado-da-arte em qualidade (30% de redução no erro de profundidade) e em velocidade (de 2% a 5%), de acordo com o autor.

A geração de um modelo de Rede Neural consiste em criar uma arquitetura capaz de trabalhar com o problema. Seu treinamento consiste em ajustar seus parâmetros (pesos) para trabalhar com o problema referente. Segundo Yosinski et al. [42], quando essas redes são treinadas com imagens, as primeiras camadas aprendem características gerais, enquanto as últimas camadas aprendem características específicas.

Por esse motivo, esses modelos de Rede Neural possuem a capacidade de transferência de aprendizado. Assim, uma vez treinado, ele pode ser ajustado para trabalhar em outra base de dados. Para isso, deve-se manter as camadas gerais e modificar, ou aumentar, as camadas específicas, permitindo aproveitar o conhecimento que o modelo já possuía e realizar melhorias e/ou adaptações para trabalhar com um novo problema [27].

No processo de otimização (*finetuning*) do novo modelo de Rede Neural, podem-se fazer pequenos ajustes nos pesos das camadas gerais. Isso também provê uma diminuição do tempo necessário de treinamento em relação ao treinamento iniciado de

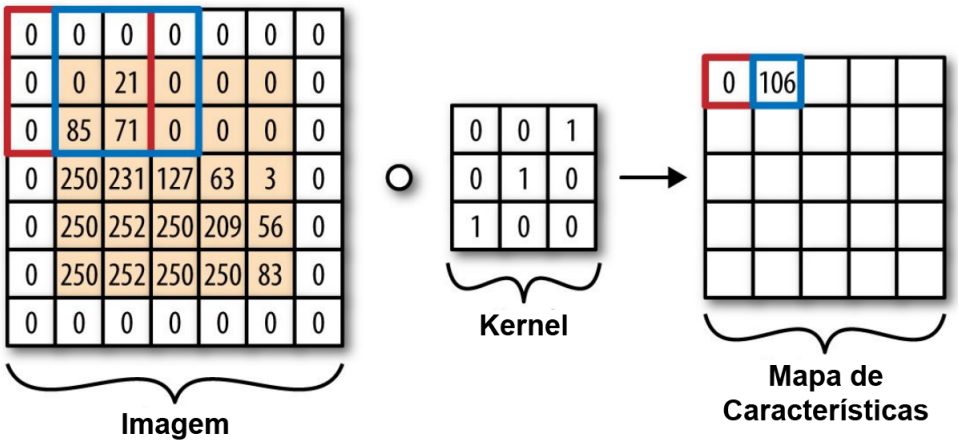


Figura 10: Processo de convolução.

Fonte: Imagem adaptada de [34].

pesos aleatórios [27, 42].

É necessário para este projeto que o modelo neural seja capaz de avaliar a similaridade da imagem original e seu mapa de disparidade. Para isso, uma arquitetura que apresenta bons resultados em cálculo de similaridade são as Redes Siamesas. Estas redes são compostas por dois modelos irmãos de Rede Neural mais uma camada final que recebe a saída de cada um desses modelos irmãos e produz uma única saída.

O estudo das Redes Siamesas iniciaram em 1993 com Baldi e Chauvin [43] e Bromley et al. [44] e trabalhos atuais apresentam Redes Siamesas compostas por CNNs que devido a arquitetura em Y, são adaptadas para trabalhar com duas imagens de entrada (Figura 12).

Zagoruyko e Komodakis [45], construíram uma rede siamesa capaz de comparar fragmentos de imagens, onde cada uma das redes convolucionais internas processavam um fragmento, extraindo suas características, e suas saídas eram comparadas pela camada final, que por sua vez, gerava como saída a similaridade dos fragmentos; Ye et al. [8], apresentaram uma rede siamesa capaz de gerar mapas de profundidade para auxiliar robôs cirúrgicos, onde cada uma das redes convolucionais internas geram um mapa de profundidade e esses mapas são comparados pela camada final para escolher o melhor; e Appalaraju e Chaoji [9] apresentaram a SimNet capaz de calcular a similaridade de imagens de forma similar ao trabalho de Zagoruyko e Komodakis [45], adicionado a aprendizagem por currículo, onde os dados de treinamento são ordenados, antes do treinamento, da menor dificuldade para a maior. Um estudo sobre as redes siamesas e um comparativo dos métodos apresentados pode ser encontrado

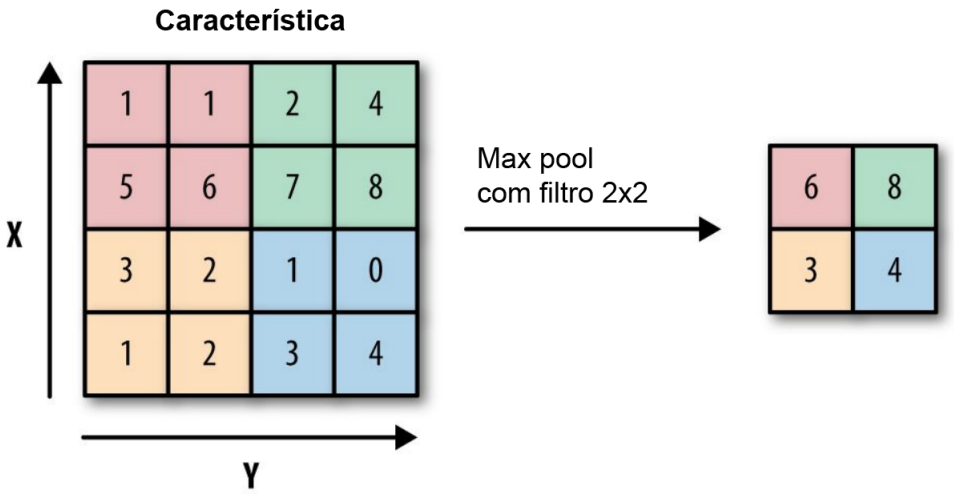


Figura 11: Processo de *pooling*.

Fonte: Imagem adaptada de [34].

em Pflugfelder [46].

As CNNs, são construídas com APIs, como o Keras, TensorFlow e o Caffe, que fornecem mecanismos para facilitar e otimizar a construção de Redes Neurais Profundas. O TensorFlow [47] e o Caffe [48] são bibliotecas de código aberto para aprendizado de máquina, fornecendo APIs de alto nível que facilitam a criação e o treinamento dos seus modelos, bem como o controle de baixo nível para máxima, flexibilidade e desempenho. O Keras [49] também é uma API de alto nível focada em redes neurais e faz uso do TensorFlow, fornecendo meios de construir modelos e realizar experimentos com menor quantidade de código.

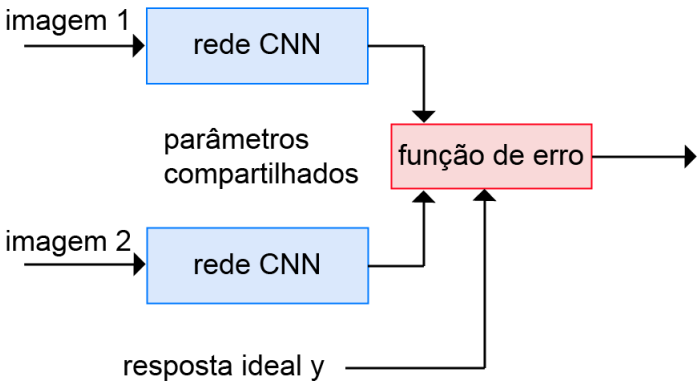


Figura 12: Arquitetura de uma Rede Siamesa.

Fonte: Imagem adaptada de [9].

3 METODOLOGIA

Neste capítulo é abordada a metodologia do trabalho, explicando-se todos os passos necessários e também as base de dados que serão utilizadas para alcançar a hipótese do trabalho, sendo como etapas: geração dos conjuntos de dados de treinamento e de testes; exploração e geração de modelos; treinamento e testes do modelo; operação e avaliação do modelo.

Conforme Goodfellow, Bengio e Courville [27], para conseguir uma alta assertividade no modelo neural, as redes precisam de um conjunto grande de dados para realizarem o treinamento. Nesta etapa, será gerado um conjunto de tuplas do tipo <*imagem original*, *mapa de profundidade*> rotulados, seu resultado esperado é o grau de assertividade, sendo o intervalo [0, 1], onde 0 (zero) indica a total falta de similaridade e 1 (um) a similaridade exata.

Para a geração do conjunto, serão utilizados algumas base de imagens já fornecidas na literatura e utilizados nos trabalhos citados que obtiveram o estado-da-arte. Dentre eles, pode-se destacar o KITTI [50] e o NYU Depth [51]. Além disso, também será utilizado o Middlebury [52]. A união dessas bases permitirá criar um conjunto grande o suficiente para testar os modelos neurais explorados neste projeto.

O mapa de profundidade fornecidos pelas bases de imagens representam a profundidade exata (*ground truth*) em relação a sua imagem. Sendo assim, todas as tuplas da base deverão ter a saída 1 (um) no modelo neural a ser desenvolvido. Além disso, faz-se necessário gerar tuplas com a imagem e/ou mapas de profundidade distorcidas com transformações, como: rotação, corte, inclinação e ruídos. Dado um percentual de distorção x aplicado a uma tupla, o modelo neural deve ser treinado para que sua saída ao avaliar esta tupla seja $(1 - x)$.

O conjunto de imagens será dividido em dois conjuntos, sendo dois terços para um conjunto de treinamento ($DS_{treinamento}$) e um terço para um conjunto alvo (DS_{testes}). O conjunto $DS_{treinamento}$, será dividido ainda mais, sendo dois terços para o conjunto de

treino (DS_{treino}) e um terço para o de testes ($DS_{validacao}$). Todas essas divisões serão de forma aleatória mantendo a proporção de cada conjunto de imagens.

Vários modelos de Redes Neurais são fornecidos publicamente para *finetuning* na Internet. O Model Zoo[1] reúne um conjunto destes modelos neurais construídos por vários pesquisadores e engenheiros, já treinados para os mais diversos problemas, como: regressão simples, classificação visual em larga escala, redes siamesas para similaridade de imagens, robótica e reconhecimento de fala. Através da transferência de aprendizado, vários desses modelos neurais já foram aprimorados, atingindo o estado-da-arte, sendo capazes até mesmo de ganhar competições, como Donahue et al. [53], Zeiler e Fergus [36] e Sermanet et al. [54].

Estes modelos neurais podem ser obtidos e modificados para se adaptar a um novo problema, para isso, uma API deve ser utilizada. O Keras é utilizado atualmente em diversos trabalhos, por melhor uso dos recursos de hardware e fornecer mais recursos para a criação de modelos de redes neurais modificáveis com poucas linhas de código. Por estes motivos, o Keras será utilizado para criar os modelos que serão utilizados neste trabalho. Além disso, os modelos neurais fornecidos pelo Model Zoo que estão sob o framework Caffe serão portados para o Keras.

Inicialmente, a SimNet e os modelos de Redes Siamesas fornecidos publicamente pelo Model Zoo serão escolhidos para experimentação, realizando o *finetuning* com o DS_{treino} do problema. Estes modelos neurais deverão ser adaptados ao problema, a camada de entrada da rede deve ser modificada para o tamanho (altura, largura, profundidade) das imagens do conjunto ou as imagens do conjunto devem ser geradas com o tamanho compatível aos modelos neurais e a camada de saída também deve ser modificada, criando uma nova camada substituta para o problema abordado. Caso necessário, novas camadas também poderão ser adicionadas ou retiradas desses modelos neurais. Sua exploração possivelmente permitirá a criação de um novo modelo de Rede Neural que forneça o resultado esperado para alcançar a hipótese desse trabalho.

O treinamento de cada modelo neural será realizado com o conjuntos DS_{treino} e $DS_{validacao}$. Após o treinamento, o modelo de Rede Neural será testado com o conjunto DS_{testes}. Dentre os modelos testados e explorados, o melhor modelo neural será escolhido (M_{alvo}). A métrica utilizada será o percentual de similaridade atingido pelo modelo sobre o conjunto de testes.

[1] Disponível em: <http://caffe.berkeleyvision.org/model_zoo.html>

Uma necessidade para o treinamento de DNNs é o processamento de alto desempenho com processadores gráficos, devido a grande quantidade de imagens. Neste projeto será utilizada a Placa Nvidia GeForce GTX 1080 com 8 GB GDDR5X de memória RAM.

Os resultados apresentados por M_{alvo} serão comparados com resultados de modelos de Rede Neural já existentes na literatura sobre o conjunto de dados DS_{testes}. E além dos testes propostos, também será realizado um experimento adicional: o M_{alvo} será utilizado como função de perda de outros trabalhos que já efetuam os mapas de profundidade e que necessitam de comparar a similaridade de seus resultados, permitindo verificar se é possível melhorar seu comportamento.

Todos os resultados serão comparados e apresentados em tabelas e gráficos quantitativos, sendo realizada então sua análise específica e sua discussão geral.

4 RESULTADOS ESPERADOS

Mapas de profundidade, como foi apresentado, fornecem a distância (profundidade) dos objetos de uma imagem e são essenciais para diversas aplicações, como robótica, auxiliando no sistema de navegação de robôs e carros autônomos, assim como na reconstrução 3D de objetos.

Construir um modelo de Rede Neural que consiga fornecer a assertividade da similaridade entre uma imagem e seu mapa de profundidade com taxa de acerto maior que o estado-da-arte em cálculo de similaridade. Acredita-se que este modelo de Rede Neural ajudará outros trabalhos que se proponham a resolver os problemas citados no subseção 1.1, uma vez que, o modelo de Rede Neural proposto pode ser utilizado como função de avaliação para outros algoritmos, como, por exemplo, algoritmos genéticos.

Dessa forma, através da substituição da função de avaliação de outros trabalhos por este modelo de Rede Neural, espera-se aumentar a taxa de acerto desses trabalhos e do estado-da-arte do problema.

Outro resultado esperado é a publicação de um artigo com os resultados obtidos.

5 CRONOGRAMA

As atividades necessárias para o desenvolvimento deste projeto estão descritas conforme a Tabela 1.

Atividade	**Descrição**
1	Gerar conjuntos de dados com imagens monoculares e mapas de profundidade.
2	Testar e explorar modelos de redes neurais profundas sobre os conjuntos de dados gerados.
3	Gerar um modelo para mensurar a assertividade entre a imagem original e seu mapa de profundidade.
4	Testar o modelo com bases de dados disponíveis na literatura.
5	Escrever monografia.
6	Efetuar correções (após a banca).

Tabela 1: Lista de atividades

O cronograma para a execução destas atividades é apresentado na Tabela 2.

Atividade	**Setembro**	**Outubro**	**Novembro**	**Dezembro**
1				
2				
3				
4				
5				
6				

Tabela 2: Cronograma de atividades

REFERÊNCIAS

1 KULKARNI, J. B.; SHEELARANI, C. Generation of depth map based on depth from focus: A survey. In: *2015 International Conference on Computing Communication Control and Automation*. IEEE, 2015. p. 716–720. ISBN 978-1-4799-6892-3. Disponível em: <http://ieeexplore.ieee.org/document/7155941/>.

2 SIUDAK, M.; ROKITA, P. A survey of passive 3D reconstruction methods on the basis of more than one image. *Machine Graphics and Vision*, v. 23, n. 3/4, p. 57–117, 2014.

3 SZEGEDY, C. et al. Going deeper with convolutions. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015. p. 1–9. ISBN 978-1-4673-6964-0. Disponível em: <http://ieeexplore.ieee.org/document/7298594/>.

4 BADRINARAYANAN, V.; HANDA, A.; CIPOLLA, R. Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *arXiv preprint arXiv:1505.07293*, 2015.

5 EIGEN, D.; PUHRSCH, C.; FERGUS, R. Depth map prediction from a single image using a multi-scale deep network. *CoRR*, abs/1406.2283, 2014. Disponível em: <http://arxiv.org/abs/1406.2283>.

6 LIU, S.; DENG, W. Very deep convolutional neural network based image classification using small training sample size. In: *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*. IEEE, 2015. p. 730–734. ISBN 978-1-4799-6100-9. Disponível em: <http://ieeexplore.ieee.org/document/7486599/>.

7 CHENG, X.; WANG, P.; YANG, R. Depth estimation via affinity learned with convolutional spatial propagation network. *arXiv preprint arXiv:1808.00150*, 2018.

8 YE, M. et al. Self-supervised siamese learning on stereo image pairs for depth estimation in robotic surgery. may 2017.

9 APPALARAJU, S.; CHAOJI, V. Image similarity using deep CNN and curriculum learning. *arXiv preprint arXiv:1709.08761*, 2017.

10 ZHUO, S.; SIM, T. On the recovery of depth from a single defocused image. In: *International Conference on Computer Analysis of Images and Patterns*. [S.l.: s.n.], 2009. p. 889–897.

11 AKIMOV, D.; VATOLIN, D.; SMIRNOV, M. Single-image depth map estimation using blur information. In: *21st GraphiCon International Conference on Computer Graphics and Vision. Conference Paper. Moscow*. [S.l.: s.n.], 2011. p. 12–15.

12 WANG, H. et al. Depth estimation from a single defocused image using multi-scale kernels. In: *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*. IEEE, 2014. p. 1524–1527. ISBN 978-1-4799-5199-4. Disponível em: <http://ieeexplore.ieee.org/document/7064542/>.

13 SZELISKI, R. *Computer vision: algorithms and applications*. [S.l.]: Springer Science & Business Media, 2010.

14 AGARWAL, S. et al. Building rome in a day. *Commun ACM*, v. 54, n. 10, p. 105, oct 2011. ISSN 00010782. Disponível em: <http://dl.acm.org/citation.cfm?doid=2001269.2001293>.

15 FU, X.; LI, Y. A survey of image-based 3D reconstruction. In: *Proceedings of the 2017 International Conference on Mechanical, Electronic, Control and Automation Engineering (MECAE 2017)*. Paris, France: Atlantis Press, 2017. ISBN 978-94-6252-315-9. Disponível em: <http://www.atlantis-press.com/php/paper-details.php?id=25873299>.

16 TAN, P.; MONASSE, P. Stereo disparity through cost aggregation with guided filter. v. 4, oct 2014.

17 SCHARSTEIN, D.; SZELISKI, R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, Springer, v. 47, n. 1-3, p. 7–42, 2002.

18 SEITZ, S. et al. A comparison and evaluation of multi-view stereo reconstruction algorithms. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06)*. IEEE, 2006. p. 519–528. ISBN 0-7695-2597-0. Disponível em: <http://ieeexplore.ieee.org/document/1640800/>.

19 KOHAVI, R.; PROVOST, F. Glossary of terms. v. 2, p. 271–274, jan 1998.

20 RUSSELL, S. J.; NORVIG, P. *Artificial intelligence: a modern approach*. [S.l.]: Malaysia; Pearson Education Limited„ 2016.

21 PRATI, R.; MONARD, M.-C. Novas abordagens em aprendizado de mÃąquina para a geraÃğÃčo de regras, classes desbalanceadas e ordenaÃğÃčo de casos. jul 2006.

22 McCulloch, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *Bull Math Biophys*, v. 5, n. 4, p. 115–133, dec 1943. ISSN 0007-4985. Disponível em: <http://link.springer.com/10.1007/BF02478259>.

23 ROSENBLATT, F. *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*. [S.l.], 1961.

24 NIELSEN, M. A. OTHER, *Neural Networks and Deep Learning*. Determination Press, 2018. Disponível em: <http://neuralnetworksanddeeplearning.com/>.

25 ELMAN, J. Finding structure in time. *Cogn Sci*, v. 14, n. 2, p. 179–211, jun 1990. ISSN 03640213. Disponível em: <http://doi.wiley.com/10.1016/0364-0213(90)90002-E>.

26 LUDERMIR, T. B.; SOUTO, M. C. P. d.; OLIVEIRA, W. R. d. Weightless neural networks: Knowledge-based inference system. In: *2008 10th Brazilian Symposium on Neural Networks*. IEEE, 2008. p. 207–212. ISBN 978-1-4244-3219-6. Disponível em: <http://ieeexplore.ieee.org/document/4665917/>.

27 GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016.

28 HAYKIN, S. S. et al. *Neural networks and learning machines*. [S.l.]: Pearson Upper Saddle River, NJ, USA:, 2009.

29 SZE, V. et al. Efficient processing of deep neural networks: A tutorial and survey. *Proc. IEEE*, v. 105, n. 12, p. 2295–2329, dec 2017. ISSN 0018-9219. Disponível em: <http://ieeexplore.ieee.org/document/8114708/>.

30 UTGOFF, P. E.; STRACUZZI, D. J. Many-layered learning. *Neural Comput*, v. 14, n. 10, p. 2497–2529, oct 2002. Disponível em: <http://dx.doi.org/10.1162/08997660260293319>.

31 LeCun, Y. et al. Backpropagation applied to handwritten zip code recognition. *Neural Comput*, v. 1, n. 4, p. 541–551, dec 1989. ISSN 0899-7667. Disponível em: <http://www.mitpressjournals.org/doi/10.1162/neco.1989.1.4.541>.

32 LECUN, Y. et al. OTHER, *Handwritten Digit Recognition with a Back-Propagation Network*. jan 1990. 396-404 p.

33 LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proc. IEEE*, v. 86, n. 11, p. 2278–2324, 1998. ISSN 00189219. Disponível em: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=726791>.

34 RAMSUNDAR, B.; ZADEH, R. *TensorFlow for Deep Learning: From Linear Regression to Reinforcement Learning*. O'Reilly Media, 2018. ISBN 9781491980453. Disponível em: <https://books.google.com.br/books?id=rtlEtAEACAAJ>.

35 KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet classification with deep convolutional neural networks. *Commun ACM*, v. 60, n. 6, p. 84–90, may 2017. ISSN 00010782. Disponível em: <http://dl.acm.org/citation.cfm?doid=3098997.3065386>.

36 ZEILER, M. D.; FERGUS, R. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013. Disponível em: <http://arxiv.org/abs/1311.2901>.

37 HE, K. et al. Deep residual learning for image recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016. p. 770–778. ISBN 978-1-4673-8851-1. Disponível em: <http://ieeexplore.ieee.org/document/7780459/>.

38 ZENG, X. et al. Crafting GBD-net for object detection. *IEEE Trans Pattern Anal Mach Intell*, v. 40, n. 9, p. 2109–2123, sep 2018. Disponível em: <http://dx.doi.org/10.1109/TPAMI.2017.2745563>.

39 HU, J.; SHEN, L.; SUN, G. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, v. 7, 2017.

40 GU, J. et al. Recent advances in convolutional neural networks. *Pattern Recognit*, v. 77, p. 354–377, may 2018. ISSN 00313203. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0031320317304120>.

41 LIU, F. et al. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Trans Pattern Anal Mach Intell*, v. 38, n. 10, p. 2024–2039, 2016. Disponível em: <http://dx.doi.org/10.1109/TPAMI.2015.2505283>.

42 YOSINSKI, J. et al. How transferable are features in deep neural networks? In: *Advances in neural information processing systems*. [S.l.: s.n.], 2014. p. 3320–3328.

43 BALDI, P.; CHAUVIN, Y. Neural networks for fingerprint recognition. *Neural Comput*, v. 5, n. 3, p. 402–418, may 1993. ISSN 0899-7667. Disponível em: <http://www.mitpressjournals.org/doi/10.1162/neco.1993.5.3.402>.

44 BROMLEY, J. et al. Signature verification using a"siamese"time delay neural network. In: *Advances in neural information processing systems*. [S.l.: s.n.], 1994. p. 737–744.

45 ZAGORUYKO, S.; KOMODAKIS, N. Learning to compare image patches via convolutional neural networks. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015. p. 4353–4361. ISBN 978-1-4673-6964-0. Disponível em: <http://ieeexplore.ieee.org/document/7299064/>.

46 PFLUGFELDER, R. P. Siamese learning visual tracking: A survey. *CoRR*, abs/1707.00569, 2017. Disponível em: <http://arxiv.org/abs/1707.00569>.

47 ABADI, M. et al. OTHER, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Disponível em: <https://www.tensorflow.org/>.

48 JIA, Y. et al. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

49 CHOLLET, F.; OTHERS. OTHER, *Keras*. 2015.

50 GEIGER, A. et al. Vision meets robotics: The KITTI dataset. *Int J Rob Res*, v. 32, n. 11, p. 1231–1237, sep 2013. ISSN 0278-3649. Disponível em: <http://journals.sagepub.com/doi/10.1177/0278364913491297>.

51 SILBERMAN, N. et al. Indoor segmentation and support inference from rgbd images. In: *European Conference on Computer Vision*. [S.l.: s.n.], 2012. p. 746–760.

52 SCHARSTEIN, D. et al. High-resolution stereo datasets with subpixel-accurate ground truth. In: *German Conference on Pattern Recognition*. [S.l.: s.n.], 2014. p. 31–42.

53 DONAHUE, J. et al. Decaf: A deep convolutional activation feature for generic visual recognition. In: *International conference on machine learning*. [S.l.: s.n.], 2014. p. 647–655.

54 SERMANET, P. et al. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.