DOI: 10.48011/asba.v2i1.1163

Redes Neurais Convolucionais para Identificação e Preensão Robótica de Objetos

Caio Cristiano Barros Viturino *
Kleber de Lima Santana Filho ** **Daniel M. de Oliveira** *
Cézar Bieniek Lemos * **André Gustavo Scolari Conceição** *

* *LaR - Laboratório de Robótica, Departamento de Engenharia Elétrica e de Computação, Universidade Federal da Bahia, Salvador, Brasil. E-mails: engcaiobarros@gmail.com, danielmoura@ufba.br, cezarcbl@protonmail.com, andre.gustavo@ufba.br*
** *PPGM - Programa de Pós-Graduação em Mecatrônica, Universidade Federal da Bahia, Salvador, Brasil. E-mail: engkleberf@gmail.com*

Abstract: This paper proposes a two-step cascaded system with the Generative Grasping Convolutional Neural Network (GG-CNN) and a modified version of the Single Shot Multibox Detector architecture (SSD) to perform robotic grasping in a vision-based object recognition system. The proposed method is called Single Shot Generative Grasping Neural Network (SSGG-CNN). The GG-CNN is a powerful object-independent grasping synthesis method well-known for the outstanding performance in open-loop and closed-loop systems using a pixel-wise grasp quality prediction. However, this technique does not allow the robot to selectively grasp objects. In order to mitigate this problem, the modified SSD was adopted to perform the object detection and selection preceding the grasping. It was achieved an average success rate of 85% considering open-loop and uncluttered objects randomly organized on a planar surface. Code is provided at `github.com/lar-deeufba/ssggcnn_ur5_grasping`.

Resumo: Este artigo propõe um sistema em cascata de duas etapas utilizando a *Generative Grasping Convolutional Neural Network* (GG-CNN) e uma versão modificada da arquitetura *Single Shot Multibox Detector* (SSD) para a realização de preensões robóticas através de reconhecimento de objetos, utilizando visão computacional. O método proposto foi denominado *Single Shot Generative Grasping Convolutional Neural Network* (SSGG-CNN). A GG-CNN é uma técnica eficiente de preensão robótica que funciona em objetos de qualquer geometria, conhecida pelo excelente desempenho em malha aberta e malha fechada, através de inferências realizadas em cada pixel de uma imagem de profundidade. No entanto, essa técnica não permite ao manipulador efetuar preensões em objetos de forma seletiva. Para mitigar este problema, a versão modificada da SSD foi adotada para a detecção e seleção de objetos para a posterior preensão. Experimentos em malha aberta produziram uma taxa média de sucesso de 85% na preensão de objetos organizados aleatoriamente em superfície plana. O código do projeto está disponível em `github.com/lar-deeufba/ssggcnn_ur5_grasping`.

Keywords: Robotic, grasping, computer vision, manipulation, robot grasping detection

Palavras-chaves: Robótica, preensão, visão computacional, manipulação, detecção de preensão robótica.

1. INTRODUÇÃO

Nos últimos anos, o avanço registrado na área de robôs autônomos, sejam eles móveis ou manipuladores, tem sido notável. As características interdisciplinares da robótica podem ser a chave para o seu crescimento exponencial. Áreas como Inteligência Artificial e Visão Computacional, por exemplo, tem contribuído significativamente nos resultados de pesquisas recentes, superando métodos clássicos baseados em abordagens analíticas e empíricas (Morrison et al., 2018).

Quando inseridos em ambientes não-estruturados, os robôs devem analisar de forma independente e contínua os dados pertencentes ao ambiente de trabalho, de forma a permitir tomadas de decisões e subsequentes reações a eventos previsíveis e imprevisíveis (Katz et al., 2008). Ademais, problemas como a preensão robótica são inerentemente baseados na percepção do ambiente; dessa forma, altas eficiências computacionais se tornam requisitos para algoritmos de controle pertencentes ao referido contexto (Lenz et al., 2015). Pesquisas recentes relacionadas à preensão robótica têm produzido resultados promissores devido ao avanço de técnicas de visão computacional baseadas em aprendizagem profunda.

É possível programar robôs manipuladores de forma manual para a realização de tarefas específicas, fornecendo instruções detalhadas ao algoritmo de controle. A este tipo de abordagem se dá o nome de analítica, sendo também comum a designação *hand-designing features* (características planejadas à mão). Esse tipo de metodologia foi amplamente utilizada em pesquisas mais antigas (Maitin-Shepard et al., 2010; Kragic and Christensen, 2003).

Apesar de apresentarem resultados satisfatórios em ambientes totalmente conhecidos, os métodos analíticos aplicados à preensão robótica não são praticáveis em ambientes dinâmicos e não-estruturados (Kober and Peters, 2010). Métodos analíticos consideram que a posição do objeto e suas regiões de contato com o efetuador final são inteiramente conhecidos (Prattichizzo and Trinkle, 2008).

Diferindo dos métodos analíticos, os métodos empíricos têm a vantagem de remover parcial ou completamente a necessidade de um modelo analítico completo (Konidaris et al., 2012). Esses métodos se concentram no uso de modelos baseados em experiências passadas, empregando técnicas de aprendizado de máquina para desenvolver modelos que relacionem os dados de sensores RGB-D a dados previamente rotulados por humanos.

Por sua vez, técnicas baseadas em aprendizagem profunda, como as Redes Neurais Convolucionais (CNN), proporcionaram um avanço considerável na preensão robótica aplicada a objetos desconhecidos. Por meio dessas técnicas, é possível extrair atributos de objetos que correspondam a uma preensão específica ou a um conjunto de preensões, superando os resultados obtidos por meio de modelos analíticos e empíricos (Johns et al., 2016; Lenz et al., 2015; Mahler et al., 2017; Morrison et al., 2018).

Apesar do bom desempenho apontado em técnicas recentes baseadas em CNN, como as de Lenz et al. (2015), Pinto and Gupta (2016), Levine et al. (2016), Johns et al. (2016), Mahler et al. (2017), Ribeiro and Grassi (2019) e Viereck et al. (2017), estas não têm apresentado bons resultados em aplicações em sistemas de malha fechada e em tempo real, em grande parte devido ao seus altos custos computacionais e por nem sempre serem capazes de considerar as melhores preensões possíveis, dado um conjunto de objetos desordenados.

Morrison et al. (2018) demonstraram experimentalmente que o baixo desempenho em alguns métodos de preensão baseados em CNN está diretamente relacionado ao número de parâmetros da rede e à sua arquitetura. Em seu trabalho, desenvolveram um método de síntese de preensão denominado *Generative Grasping Convolutional Neural Network* (GG-CNN) e, para testar seu desempenho, foi utilizado um conjunto de objetos padronizados encontrados em Mahler et al. (2017) e Viereck et al. (2017). A técnica supera métodos anteriores no quesito de geração rápida de preensões, mesmo ao utilizar malha-fechada em ambientes com objetos desordenados e dinâmicos.

Quando se trata de indústria 4.0, existe uma alta demanda por manipuladores robóticos flexíveis e versáteis, capazes de lidar com novos ambientes e peças de novos produtos (Caldas et al., 2019). Nesse cenário, a manufatura aditiva tem mostrado uma crescente demanda por sistemas do tipo, objetivando o aumento da eficácia da cadeia de suprimentos e disponibilidade de produto, a qual regularmente lida com um número crescente de novas peças (de Brito et al., 2019). Apesar do exposto, nenhum dos trabalhos citados anteriormente foi projetado para efetuar preensões em ambientes onde apenas alguns dos objetos no campo de visão da câmera devem ser manipulados. Por não serem capazes de efetuar a distinção entre diferentes classes de objetos acabam por computar possíveis preensões em todos aqueles presentes na cena.

Este trabalho propõe o desenvolvimento de um sistema de preensão robótica, utilizando a rede GG-CNN, e uma variação, previamente utilizada e treinada por Lemos et al. (2019), da arquitetura *Single Shot Multibox Detector* (SSD) (Liu et al., 2016) em cascata. A partir da integração com a variante da SSD modificada, espera-se que sistema permita reconhecer objetos de interesse e estimar os parâmetros necessários para uma preensão adequada. As informações provenientes da SSD modificada são processadas e o resultado é usado como entrada para a GG-CNN. O presente método foi denominado *Single Shot Generative Grasp Convolutional Neural Network* (SSGG-CNN), uma vez que incorpora propriedades dos dois métodos. O método proposto será validado por meio da simulação em uma célula de manufatura aditiva, onde vários objetos manufaturados serão selecionados e em seguida manipulados.

O restante do artigo está organizado da seguinte forma: serão discutidos os trabalhos relacionados na Seção 2. A definição do problema será tratada na Seção 3. As redes GG-CNN e SSD serão apresentadas, respectivamente, nas Seções 4 e 5. O método de preensão proposto (SSGG-CNN) será descrito na Seção 6. Os protocolos experimentais serão detalhados na Seção 7. Por fim, a Seção 8 trará a conclusão deste trabalho.

2. TRABALHOS RELACIONADOS

Johns et al. (2016) usou uma CNN para detectar preensões robóticas em objetos fixos, individualmente. Em vez de considerar apenas uma posição específica de preensão, esse método analisa um conjunto de preensões e seleciona a que considera melhor, de acordo com as incertezas relacionadas à dinâmica desta. Nessa pesquisa, a rede neural foi treinada usando objetos sintéticos. Este método fornece a posição (x, y) e orientação (θ) em relação ao plano dos objetos em questão. Dessa forma, o efetuador final deve aproximar-se do objeto ortogonalmente ao plano composto pelos objetos (preensão planar).

Mahler et al. (2017) usou um banco de dados de objetos sintéticos para treinar sua rede *Grasp Quality Convolutional Neural Network* (GQ-CNN). A CNN usada infere a posição de preensão diretamente da imagem de profundidade gerada por um sensor RGB-D. Em seu trabalho, as preensões estáveis são geradas por meio de um método analítico robusto denominado *quasi-static GWS analysis* (Goldberg et al., 1999). O método demonstrou baixa eficiência ao considerar objetos desordenados.

O trabalho de Levine et al. (2016) foi um dos primeiros a associar técnicas de aprendizagem profunda à preensão robótica de objetos desordenados em malha fechada. A rede neural utilizada possui um grande número de camadas, cerca de um milhão de parâmetros, resultando em

frequências experimentais que variavam entre 2 e 5 Hz. Além disso, nessa abordagem, a câmera é fixa, dificultando a adaptação a mudanças bruscas no ambiente de trabalho. Viereck et al. (2017) supera a limitação encontrada em Levine et al. (2016) diminuindo o número de camadas da rede neural e posicionando a câmera no efetuador final do robô. A frequência máxima de processamento alcançada foi de 5 Hz, não sendo ainda adequada para aplicações em malha fechada.

Morrison et al. (2018) demonstraram em seu estudo o impacto direto da quantidade de parâmetros de um modelo CNN no desempenho de um sistema de geração de preensões, algo crucial em ambientes dinâmicos e desordenados. Morrison et al. (2018) trataram o problema de desempenho computacional encontrado nos trabalhos de Viereck et al. (2017) e Levine et al. (2016), através da sua *Generative Grasping Convolutional Neural Network* (GG-CNN), um modelo CNN relativamente pequeno de apenas 6 camadas e 62 mil parâmetros. A GG-CNN foi capaz de gerar e avaliar possíveis preensões para cada pixel de uma imagem em frequências de aproximadamente 50 Hz, demonstrando eficiência na geração de preensão em malha fechada. A técnica alcançou a taxa de sucesso de 83% na tentativa de preensões executadas em um conjunto de objetos desconhecidos com geometria adversa (Mahler et al., 2017) em malha fechada, 88% em um conjunto de objetos domésticos em malha fechada, movidos durante a tentativa de captura, e precisão de 81% ao ser aplicada a objetos desordenados e dinâmicos também em malha fechada.

3. DEFINIÇÃO DO PROBLEMA

No presente trabalho, o sistema proposto deve gerar preensões planares robustas, utilizando um sensor RGB-D. Este deve ser capaz de executar preensões antipodais em objetos de interesse previamente selecionados.

3.1 Premissas

É assumido que os objetos são inseridos em uma superfície plana e a preensão é realizada por um efetuador final de dedos paralelos usando uma única câmera posicionada no punho do robô. Todos os parâmetros internos da câmera são conhecidos. A preensão é definida pela posição e orientação de um efetuador final de geometria conhecida, como mostrado na Fig. 1. Os parâmetros associados à preensão, utilizados na referida figura e restante do trabalho podem ser encontrados na Subseção 3.2.

3.2 Definições

Imagem de profundidade. $\boldsymbol{I} = \mathbb{R}^{H\times W}$ denota uma imagem de profundidade 2.5D, em que todos os objetos no ambiente de trabalho são considerados.

Imagem de profundidade filtrada. $\boldsymbol{I_f} = \mathbb{R}^{H\times W}$ descreve uma imagem de profundidade 2.5D filtrada, em que somente objetos de interesse são considerados. H e W representam a altura e largura da imagem, respectivamente.

Sistema de coordenadas da base do robô. $\boldsymbol{C}$ expressa o sistema de coordenadas da base do robô.

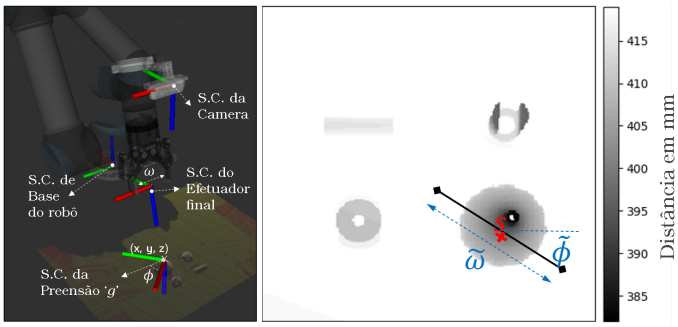

Figura 1. (Esquerda) Representação do sistema de coordenadas, representado de forma abreviada na imagem por S.C., de preensão $\boldsymbol{g} = (\boldsymbol{p}, \phi, \omega, q)$ e dos outros sistemas de coordenadas aplicados à GG-CNN. (Direita) Representação de $\tilde{\boldsymbol{g}}^*_{\boldsymbol{\theta}}$ na Imagem de profundidade $\boldsymbol{I}$, com os objetos encontrados em Lemos et al. (2019)

.

Preensão única em $\boldsymbol{C}$. $\boldsymbol{g} = (\boldsymbol{p}, \phi, \omega, q)$ representa uma única preensão em $\boldsymbol{C}$, onde $p = (x, y, z)$ expressa o ponto central do efetuador final no objeto, ϕ simboliza a orientação do efetuador final em torno do eixo z, ω denota a largura da preensão e q descreve a chance de sucesso da preensão identificada, medida a partir do valor de qualidade de uma das saídas da GG-CNN.

Preensão única em $\boldsymbol{I}$. $\tilde{\boldsymbol{g}} = (\boldsymbol{s}, \tilde{\phi}, \tilde{\omega}, q)$ expressa uma preensão única em $\boldsymbol{I}$, onde s indica o ponto central da preensão em *pixels*, $\tilde{\phi}$ evidencia a rotação no eixo de coordenadas da câmera e $\tilde{\omega}$ se refere à largura da preensão.

Conjunto de preensões. $\tilde{\boldsymbol{G}} = (\tilde{\boldsymbol{\Phi}}, \tilde{\boldsymbol{W}}, \tilde{\boldsymbol{Q}}) \in \mathbb{R}^{3\times H\times W}$ descreve (em $\boldsymbol{I}$) um conjunto de preensões $\tilde{\boldsymbol{g}}$, onde $\tilde{\boldsymbol{\Phi}}, \tilde{\boldsymbol{W}}$ e $\tilde{\boldsymbol{Q}}$ consistem em valores de $\tilde{\phi}, \tilde{\omega}$ e q para cada *pixel*.

Função de preensão. $M(\boldsymbol{I}) = \tilde{\boldsymbol{G}}$ descreve uma função de preensão que converte uma imagem de profundidade $\boldsymbol{I}$ em um mapa de preensões $\tilde{\boldsymbol{G}}$.

Função aproximada de preensão. $M_\theta(\boldsymbol{I})$ representa uma aproximação da função de preensão através de uma CNN, em que θ indica os pesos da rede.

Preensões estimadas. $\tilde{\boldsymbol{G}}_{\boldsymbol{\theta}} = (\tilde{\boldsymbol{\Phi}}_{\boldsymbol{\theta}}, \tilde{\boldsymbol{W}}_{\boldsymbol{\theta}}, \tilde{\boldsymbol{Q}}_{\boldsymbol{\theta}}) \in \mathbb{R}^{3\times H\times W}$ descreve um conjunto de preensões estimadas $\tilde{\boldsymbol{G}}$ em $\boldsymbol{I}$, onde $\tilde{\boldsymbol{\Phi}}_{\boldsymbol{\theta}}, \tilde{\boldsymbol{W}}_{\boldsymbol{\theta}}$ e $\tilde{\boldsymbol{Q}}_{\boldsymbol{\theta}}$ consistem nos valores aproximados de $\tilde{\phi}, \tilde{\omega}$ e q para cada *pixel*.

Preensões estimadas e filtradas. $\tilde{\boldsymbol{G}}_{\boldsymbol{f}} = (\tilde{\boldsymbol{\Phi}}_{\boldsymbol{f}}, \tilde{\boldsymbol{W}}_{\boldsymbol{f}}, \tilde{\boldsymbol{Q}}_{\boldsymbol{f}}) \in \mathbb{R}^{3\times H\times W}$ evidencia a função de aproximação de preensão em $\boldsymbol{I_f}$, onde $\tilde{\boldsymbol{\Phi}}_{\boldsymbol{f}}, \tilde{\boldsymbol{W}}_{\boldsymbol{f}}$ e $\tilde{\boldsymbol{Q}}_{\boldsymbol{f}}$ consistem nos valores aproximados de $\tilde{\phi}, \tilde{\omega}$ e q para cada *pixel*.

Melhor preensão visível em C. $\boldsymbol{g}^*_{\boldsymbol{\theta}}$ simboliza a melhor preensão visível em $\boldsymbol{C}$.

Melhor preensão visível em $\boldsymbol{I}$. $\tilde{\boldsymbol{g}}^*_{\boldsymbol{\theta}}$ representa a melhor preensão visível em $\boldsymbol{I}$, resultante de $\tilde{\boldsymbol{G}}_{\boldsymbol{\theta}}$.

Melhor preensão visível em $\boldsymbol{I_f}$. $\tilde{\boldsymbol{g}}^*_{\boldsymbol{f}}$ indica a melhor preensão visível em $\boldsymbol{I_f}$, proveniente de $\tilde{\boldsymbol{G}}_{\boldsymbol{f}}$.

4. GENERATIVE GRASP CONVOLUTIONAL NEURAL NETWORK

Esta seção descreve o funcionamento da GG-CNN de forma sucinta. Embora seja suficiente para a compreensão do método proposto, para maiores detalhes, consulte Morrison et al. (2018). A GG-CNN é uma rede neural de topologia puramente convolutiva que aproxima a função complexa $M(\boldsymbol{I})$. Foi demonstrado que $M_\theta(\boldsymbol{I}) = \tilde{\boldsymbol{G}}_{\boldsymbol{\theta}} = (\tilde{\boldsymbol{\Phi}}_{\boldsymbol{\theta}}, \tilde{\boldsymbol{W}}_{\boldsymbol{\theta}}, \tilde{\boldsymbol{Q}}_{\boldsymbol{\theta}})$ pode ser aprendida através de entradas $\tilde{\boldsymbol{I}}_T$ e suas saídas correspondentes $\tilde{\boldsymbol{G}}_{\boldsymbol{T}}$, provenientes de um conjunto de dados de treinamento, aplicando a função de custo L2, representada por $\mathcal{L}$, de forma que:

$$\theta = \underset{\theta}{argmin}\ \mathcal{L}(\tilde{\boldsymbol{G}}_{\boldsymbol{T}}, M_\theta(\tilde{\boldsymbol{I}}_T)) \tag{1}$$

Ao invés de realizar uma análise de I para gerar uma única preensão, $\tilde{\boldsymbol{g}}_{\boldsymbol{\theta}}$ é obtida em cada *pixel* da imagem de profundidade I para gerar um mapa de preensões $\tilde{\boldsymbol{G}}_{\boldsymbol{\theta}}$.

4.1 Representação da Preensão

O mapa de preensões $\tilde{\boldsymbol{G}}_{\boldsymbol{\theta}}$ é representado como um conjunto de três imagens de $300 \times 300\,px$, as quais correspondem a $\tilde{\boldsymbol{\Phi}}_{\boldsymbol{\theta}}$, $\tilde{\boldsymbol{Q}}_{\boldsymbol{\theta}}$ e $\tilde{\boldsymbol{W}}_{\boldsymbol{\theta}}$. Os valores em $\tilde{\boldsymbol{Q}}_{\boldsymbol{\theta}}$, contidos no intervalo $[0, 1]$, indicam a chance de sucesso da preensão, sendo que os valores mais altos indicam uma maior chance de sucesso. Os valores em $\tilde{\boldsymbol{\Phi}}_{\boldsymbol{\theta}}$, contidos no intervalo $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$, indicam a orientação da preensão antipodal, simétrica em torno de $\pm\frac{\pi}{2}\,rad$. Os valores em $\tilde{\boldsymbol{W}}_{\boldsymbol{\theta}}$, contidos no intervalo $[0, 150]$ (para permitir invariâncias na profundidade), correspondem à largura da preensão em *pixels*. Esses *pixels* são convertidos em metros utilizando medições de profundidade e parâmetros intrínsecos da câmera.

Através de $\tilde{\boldsymbol{G}}_{\boldsymbol{\theta}}$ é possível calcular $\tilde{\boldsymbol{g}}^*_{\boldsymbol{\theta}}$, encontrando o índice do maior valor presente em $\tilde{\boldsymbol{Q}}_{\boldsymbol{\theta}}$. Esse índice é usado na obtenção da orientação do efetuador final e da largura da preensão em $\tilde{\boldsymbol{\Phi}}_{\boldsymbol{\theta}}$ e $\tilde{\boldsymbol{W}}_{\boldsymbol{\theta}}$, respectivamente. Por fim, $\boldsymbol{g}^*_{\boldsymbol{\theta}}$ é obtido através de uma sequência de transformações homogêneas conhecidas, como em

$$\boldsymbol{g}^*_\theta = t_{RC}(t_{CI}(\tilde{\boldsymbol{g}^*_{\boldsymbol{\theta}}})) \tag{2}$$

onde t_{CI} indica a transformação do sistema de coordenadas de $\boldsymbol{I}$ para o sistema de coordenadas da câmera e t_{RC} representa a transformação do sistema de coordenadas da câmera em relação a $\boldsymbol{C}$, baseando-se em parâmetros intrínsecos e conhecidos da câmera.

4.2 Banco de Dados de Treinamento

A performance da GG-CNN foi mantida, usando os mesmos dados de treinamentos encontrados em Morrison et al. (2018) para comparação. O banco de dados de preensões da Universidade de Cornell, nomeado de *Cornell Grasping Dataset* (CGD) (Lenz et al., 2015), foi aumentado para gerar um conjunto de 8.840 imagens de profundidade $\tilde{\boldsymbol{I}}_{\boldsymbol{T}}$ e mapas de preensões associadas $\tilde{\boldsymbol{G}}_{\boldsymbol{T}}$, integrando 51.100 amostras de preensão. O CGD disponibiliza múltiplos exemplos válidos de preensão por objeto, resultando em melhores inferências de $\tilde{\boldsymbol{G}}$, em comparação a trabalhos como o de Mahler et al. (2017), no qual apenas um exemplo de preensão válida era considerado. O CGD representa uma preensão antipodal como um retângulo, usando coordenadas em *pixels*, alinhadas à posição e orientação do efetuador final, como em Jiang et al. (2011).

Morrison et al. (2018) dividiu cada retângulo de preensão em três partes iguais, usando a divisão central para representar uma preensão válida. Essa posição, representada por $\boldsymbol{p}$, corresponde ao valor máximo de qualidade na imagem $\tilde{\boldsymbol{Q}}_{\boldsymbol{T}}$. Devido à simetria da preensão antipodal, cada orientação disposta em $\tilde{\boldsymbol{\Phi}}_{\boldsymbol{T}}$ é obtida através da atribuição de dois componentes vetoriais, $\text{sen}(\boldsymbol{2}\tilde{\boldsymbol{\Phi}}_{\boldsymbol{T}})$ e $\cos(\boldsymbol{2}\tilde{\boldsymbol{\Phi}}_{\boldsymbol{T}})$, no intervalo $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$. Esse processo elimina a descontinuidade em torno de $\pm\frac{\pi}{2}$ e pode ser encontrado em Hara et al. (2017). O valor correspondente à largura da preensão é calculada em *pixels* no intervalo $[0, 150]$ para cada retângulo de preensão e, em seguida, é associado à respectiva área de $\tilde{\boldsymbol{W}}_{\boldsymbol{T}}$. Os valores de $\tilde{\boldsymbol{W}}_{\boldsymbol{T}}$ são normalizados no intervalo $[0, 1]$ durante o treinamento.

4.3 Treinamento e Arquitetura

A GG-CNN apresenta como saída quatro imagens de $300 \times 300\,px$: $\text{sen}(\boldsymbol{2}\tilde{\boldsymbol{\Phi}}_{\boldsymbol{\theta}})$, $\cos(\boldsymbol{2}\tilde{\boldsymbol{\Phi}}_{\boldsymbol{\theta}})$, $\tilde{\boldsymbol{Q}}_{\boldsymbol{\theta}}$ e $\tilde{\boldsymbol{W}}_{\boldsymbol{\theta}}$. A imagem $\tilde{\boldsymbol{\Phi}}_{\boldsymbol{\theta}}$ é obtida através de vetor unitário $2\tilde{\Phi}_\theta$ por $\tilde{\Phi}_\theta = \frac{1}{2}\,\text{arctg}\frac{\text{sen}(\boldsymbol{2}\tilde{\boldsymbol{\Phi}}_{\boldsymbol{\theta}})}{\cos(\boldsymbol{2}\tilde{\boldsymbol{\Phi}}_{\boldsymbol{\theta}})}$. A GG-CNN possui uma arquitetura pequena, contendo 62.420 parâmetros, resultando em uma performance ideal para aplicações em malha fechada. 80% das 8.840 imagens $\boldsymbol{I}_{\boldsymbol{T}}$ foram utilizadas para treinamento e 20% foram utilizadas para avaliação. Nenhum ruído adicional é adicionado, uma vez que o CGD já incorpora ruídos da câmera RGB-D. A partir da avaliação em um grupo de 1701 imagens, Morrison et al. (2018) selecionou dentre 95 arquiteturas aquela que obteve maior performance.

5. SINGLE SHOT MULTIBOX DETECTOR

A presente seção apresenta de forma sucinta o funcionamento da SSD, para maiores detalhes o trabalho de Liu et al. (2016) pode ser consultado.

A SSD é um modelo de aprendizagem profunda proposto por Liu et al. (2016) para a identificação de múltiplos objetos. Em seu trabalho apresentou duas variantes do modelo: a SSD300 e a SSD512, as quais foram treinadas a partir de imagens de respectivamente $300 \times 300\,px$ e $512 \times 512\,px$. O modelo SSD produz, para cada objeto de interesse na imagem de entrada, as coordenadas de uma caixa delimitadora que envolve o objeto, um identificador de classe para o objeto, e um valor de confiança na predição. Segundo o referido autor, a SSD supera a performance de trabalhos anteriores em termos de imagens processadas por segundo.

No modelo original da SSD, Liu et al. (2016) utiliza, como camadas iniciais, parte de uma rede previamente projetada e treinada para a identificação de objetos, dispensando as camadas utilizadas para a classificação da mesma. A este grupo de camadas é dado o nome *base network* (rede base). Em seu trabalho escolheu para esse papel a arquitetura VGG16 (Simonyan and Zisserman, 2014), sugerindo a possibilidade de adoção de outras diferentes.

A parte restante da rede, denominada parte auxiliar, é composta por um número de camadas convolucionais que decrescem em tamanho, de forma progressiva. O autor do referido trabalho sugere que tal característica possibilita ao sistema realizar previsões de detecção em múltiplas escalas.

Lemos et al. (2019) avaliou a performance dos modelos: *Faster R-CNN*, SSD300 e a SSD512, com base ResNet50 (He et al., 2016), na detecção de objetos produzidos por impressora 3D (manufatura aditiva). Devido à alteração da VGG16 para ResNet50 como rede base, a SSD512 modificada será tratada como SSD-ResNet50 dessa parte do texto em diante. A SSD-ResNet50, apesar de apresentar uma taxa de acertos levemente inferior à *Faster R-CNN*, apresentou uma frequência de processamento aproximadamente 15 vezes maior, sendo superior nos dois quesitos à SSD300 com a base original (VGG16).

Para a validação do sistema será utilizada a SSD-ResNet50 previamente treinada e avaliada em Lemos et al. (2019). Para o treinamento, os referidos autores utilizaram uma base de dados composta por imagens de seis classes de objetos fabricados por meio de manufatura aditiva (Fig. 2).

.

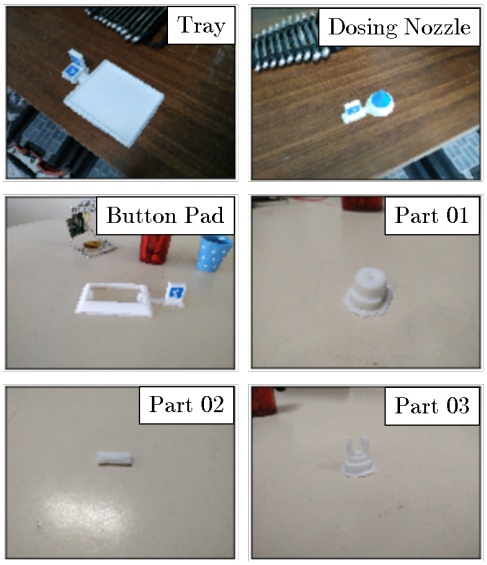

Figura 2. Classes dos objetos utilizados no treinamento e no teste da rede (Lemos et al., 2019)

6. SINGLE SHOT GENERATIVE GRASPING CONVOLUTIONAL NEURAL NETWORK

Como apontado anteriormente, a GG-CNN permite a geração de preensões em objetos desordenados de forma eficiente. No entanto, a técnica não foi desenvolvida para a realização de preensões em ambientes em que, existam objetos que não devam ser manipulados, considerando-se o campo de visão da câmera. A Fig. 3 mostra que $\tilde{\boldsymbol{g}}^*_{\boldsymbol{\theta}}$ pode ser encontrado, mesmo para objetos que não foram considerados durante o treinamento da SSD-ResNet50. É importante salientar que a implementação da SSGG-CNN requer um sensor RGB-D, uma vez que os objetos são detectados utilizando imagens RGB e as preensões robóticas são geradas a partir de imagens de profundidade. O processo de detecção da SSGG-CNN será descrito na próxima subseção.

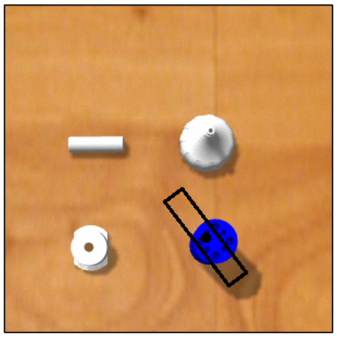

Figura 3. Imagem RGB demonstrando objetos de interesse (em branco) utilizados no trabalho de Lemos et al. (2019) e objetos que não devem ser manipulados (em azul). O retângulo de preensão apresentado na imagem corresponde à detecção realizada pela GG-CNN em um objeto que deveria ser ignorado.

6.1 Processo de detecção de preensões

O processo de detecção SSGG-CNN é composto por sete estágios: (1) armazenamento das imagens de profundidade e das posições preliminares de preensão (2) detecção de objetos usando a SSD-ResNet50, (3) indexação das caixas delimitadoras nas imagens RGB, (4) criação da imagem de profundidade com objetos envoltos em caixas delimitadoras, (5) Avaliação do GG-CNN, (6) processamento da preensão sugerida e (7) Obtenção de $\boldsymbol{g}^*_{\boldsymbol{\theta}}$.

No estágio (1), o robô é movido para uma posição fixa prévia à preensão e todos os objetos são mantidos fora do espaço de trabalho do robô. Nessas condições, uma imagem de profundidade é adquirida e armazenada para ser usada no estágio 4.

No estágio (2), todos os objetos são colocados na área de trabalho do robô. A SSD-ResNet50 analisa uma imagem RGB contendo todos os objetos. Dessa forma, os objetos de interesse, mostrados na Fig. 2, são detectados em meio aos outros objetos que não devem ser manipulados pelo robô. Ao final desse estágio são geradas caixas delimitadoras para cada objeto.

No estágio (3), obtemos o índice de cada caixa delimitadora advinda do estágio (2) para a gerar índices em posições análogas na imagem de profundidade $\boldsymbol{I}$. É indispensável que a imagem RGB e a imagem de profundidade estejam alinhadas. Desse modo, a região correspondente à caixa delimitadora na imagem RGB corresponderá a mesma região na imagem de profundidade.

No estágio (4), utilizando os índices gerados, as caixas delimitadoras são recriadas para a imagem de profundidade $\boldsymbol{I}$, dessa forma gerando a imagem de profundidade filtrada $\boldsymbol{I_f}$, na qual apenas os objetos de interesse, marcados por caixas delimitadoras, são considerados.

No estágio (5), a imagem de profundidade filtrada $\boldsymbol{I_f}$ é cortada em um quadrado de $300 \times 300\,px$, satisfazendo a entrada da GG-CNN. Possíveis valores inválidos são substituídos utilizando-se a função *inpaint* do OpenCV (Bradski, 2000).

No estágio (6), $\tilde{\boldsymbol{g}}_{\boldsymbol{f}}^{*}$ é fornecida pela GG-CNN. Morrison et al. (2018) descobriu que o processo de filtragem de $\tilde{\boldsymbol{Q}}_{\boldsymbol{f}}$ utilizando um *kernel* gaussiano, semelhante ao aplicado em Johns et al. (2016), melhorou a robustez no processo de preensão e melhorou a convergência de $\tilde{\boldsymbol{g}}_{\boldsymbol{f}}^{*}$.

No estágio (7), $\boldsymbol{g}_{\boldsymbol{\theta}}^{*}$ é calculada usando (2) da seguinte forma:

$$\boldsymbol{g}_{\theta}^{*} = t_{RC}(t_{CI}(\tilde{\boldsymbol{g}_{\boldsymbol{f}}^{*}})) \tag{3}$$

O processo especificado é demonstrado pela Fig. 4.

7. EXPERIMENTOS

7.1 Simulação e Componentes Físicos

Neste trabalho, os experimentos foram realizados no simulador robótico Gazebo (Koenig and Howard, 2004) integrado ao *Robot Operating System* ROS (Quigley et al., 2009) versão *Kinetic*. Foi utilizado um modelo virtual do robô UR5 da *Universal Robots* (Robots, 2019), equipado com o efetuador final 2F-85 da *ROBOTIQ* e com a câmera Intel Realsense D435. A distância entre a câmera e o ponto central entre os dedos do efetuador final é de, aproximadamente, 23 cm. O suporte da câmera, manufaturado em uma impressora 3D, foi rotacionado em 14°, na direção do efetuador final, como mostra a Fig. 1. As simulações foram realizadas em um *notebook* com o processador Intel Core i7 7700HQ e a placa de vídeo NVIDIA GeForce GTX1050 Ti, utilizando o sistema operacional Ubuntu 16.04. A GG-CNN obtém uma nova preensão em aproximadamente 28 ms e a SSD-ResNet50 identifica um objeto em aproximadamente 99 ms, com o código escrito predominantemente em Python. Quando as duas redes neurais são executadas em cascata, o custo computacional aumenta e o tempo necessário para a obtenção de uma nova preensão é de 205 ms aproximadamente. Como o *hardware* especificado é inferior ao encontrado em Lemos et al. (2019) e Morrison et al. (2018), o tempo para obtenção da preensão e detecção de objetos é maior se comparado aos mesmos.

7.2 Experimentos

Nas simulações, as preensões foram realizadas em malha aberta, com múltiplos objetos organizados aleatoriamente em uma superfície plana. Dentre os objetos dispostos, apenas 3 são considerados como sendo de interesse, os remanescentes não fazem parte do conjunto utilizado durante o treinamento do sistema para serem reconhecidos. Após o término da preensão, cada objeto movido não é reposto no espaço de trabalho do robô. As classes dos objetos empregados estão indicadas na Fig. 2. É estabelecido que a tentativa de preensão é bem-sucedida quando um objeto de interesse é identificado e transportado até uma bandeja. A taxa de acertos é representada como uma fração do número total de tentativas de preensão. Dois experimentos foram realizados para verificar a eficiência da SSGG-CNN em relação a GG-CNN. As posições dos objetos foram escolhidas aleatoriamente e essa mesma configuração foi mantida nos dois experimentos. Para obter os ângulos das juntas do robô, dadas a posição e a orientação da preensão, utilizou-se o TRAC-IK (Beeson and Ames, 2015), uma biblioteca *open-source* para solucionar problemas de cinemática inversa.

7.3 Execução da preensão com a GG-CNN

A Fig. 6a mostra um experimento com modelos 3D de objetos que devem ser manipulados (em branco) encontrados em Lemos et al. (2019), e objetos que devem ser ignorados (em azul) organizados de forma aleatória. As preensões são realizadas em $\boldsymbol{I} = \mathbb{R}^{H \times W}$, como mencionado na Seção 3. Na Fig. 6, o robô realiza uma preensão bem-sucedida no *Dosing Nozzle* (Fig. 6b), e depois a preensão é realizada na *part 02* com sucesso (Fig. 6c). Como a GG-CNN não é capaz de selecionar objetos, o robô pode inferir preensões em objetos indesejáveis, como mostrado na Fig. 6d. Essa preensão é então considerada como falha.

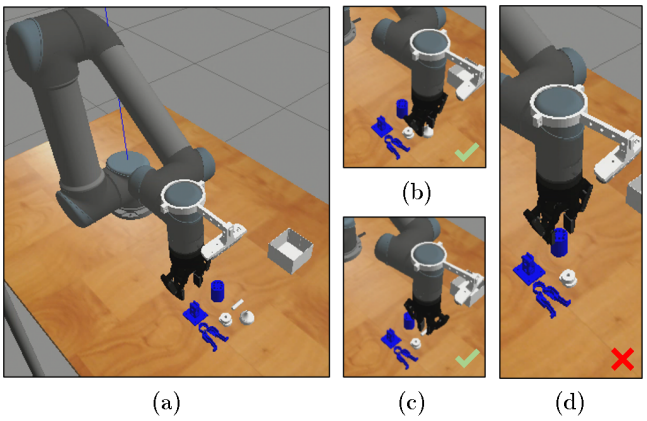

Figura 6. (a) Espaço de trabalho com objetos que devem ser ignorados (em azul) e objetos que devem ser manipulados (em branco) (b) Preensão bem-sucedida realizada no *Dosing Nozzle* (c) Preensão bem-sucedida realizada na *part 02* (d) GG-CNN falha ao associar uma preensão a um objeto que não pertence ao conjunto de objetos de interesse.

7.4 Execução da preensão com a SSGG-CNN

Foi realizado um experimento com a SSGG-CNN no Gazebo (Fig. 7a) com os mesmos objetos utilizados no experimento anterior (Fig. 6).

Inicialmente, os objetos de interesse são identificados (Fig. 7b) e as preensões são executadas em $\boldsymbol{I_f} = \mathbb{R}^{H \times W}$, como mencionado na Seção 6. As duas primeiras tentativas de preensão (Fig. 7c e Fig. 7d) são equivalentes à GG-CNN, porém, na terceira tentativa de preensão, o robô executa a preensão em um objeto de interesse com sucesso (Fig. 7e). Desta forma, dos vinte experimentos realizados, 85% deles foram bem sucedidos.

O vídeo dos experimentos pode ser encontrado no link `https://www.youtube.com/watch?v=aJ39MruDdLo`.

8. CONCLUSÃO

Neste artigo, um sistema em cascata de duas etapas foi proposto através da utilização da GG-CNN e uma versão modificada da arquitetura SSD, denominada de SSD-ResNet50, para a realização de preensões robóticas através de reconhecimento de objetos. Este sistema, denominado *Single Shot Generative Grasping Convolutional Neural Network* (SSGG-CNN), demonstrou avanços em relação à

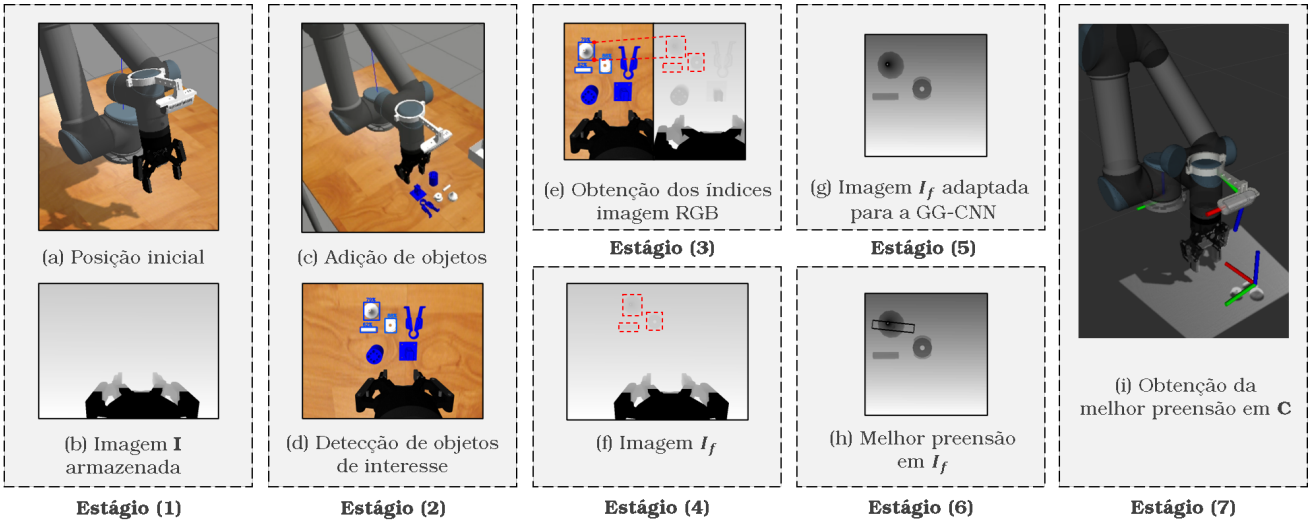

Figura 4. Processo de preensão da SSGG-CNN.

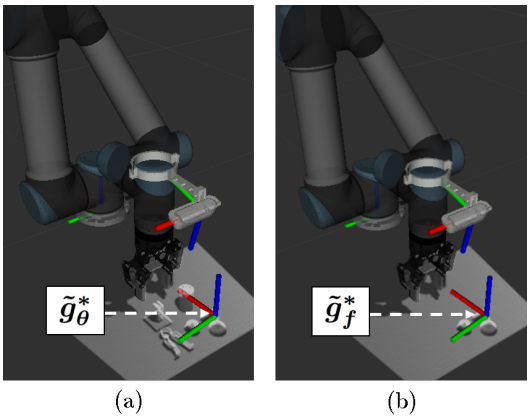

Figura 5. Imagens da nuvem de profundidade adquirida através da câmera *Intel Realsense D435* simulada. (a) Nuvem de profundidade bruta e a representação de $\tilde{g}^*_\theta$ (b) Nuvem de profundidade obtida no Estágio (4) e a representação de $\tilde{g}^*_f$.

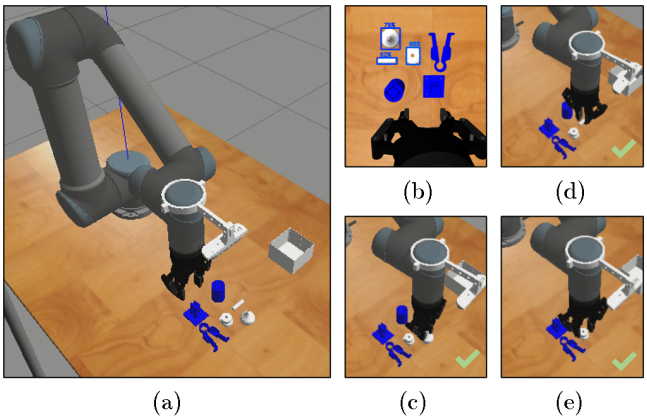

Figura 7. (a) Espaço de trabalho com objetos que devem ser ignorados (na cor azul) e objetos que devem ser manipulados (na cor branca) (b) Objetos de interesse identificados pela SSD-ResNet50, representados pela caixa delimitadora na cor azul (c) Preensão bem-sucedida realizada no *Dosing Nozzle* (d) Preensão bem-sucedida realizada na *part 02* (e) Preensão bem-sucedida na *part 01*.

GG-CNN de Morrison et al. (2018), uma vez que permite a identificação de objetos de interesse, a partir da análise de uma imagem RGB-D. Considerando o cenário ao qual o modelo proposto foi submetido, observou-se que 85% das tentativas de preensão foram bem-sucedidas.

O processamento da SSGG-CNN é consideravelmente mais lento, quando comparado à GG-CNN original, devido ao número total de parâmetros da rede, visto que o desempenho é diretamente afetado, conforme apontado em Morrison et al. (2018). O SSGG-CNN leva cerca de 205 ms para calcular uma nova preensão em comparação com os 28 ms da GG-CNN. Vale ressaltar que o número de parâmetros da técnica proposta (SSGG-CNN) é aproximadamente 550% maior que a própria GG-CNN. Apesar da desvantagem apontada, o método teve desempenho suficientemente confiável, uma vez que é utilizado para a detecção e preensão de objetos em ambientes estáticos.

Embora no presente trabalho não tenham sido utilizados conjuntos de objetos padronizados, aqueles utilizados no presente trabalho estão disponíveis no repositório do projeto no Github, de forma a tornar mais fácil a reprodução dos experimentos.

Uma vez que este artigo corresponde à continuação do trabalho apresentado por César Bieniek Lemos, um dos co-autores deste trabalho, os resultados das métricas aplicadas à SSD512-ResNet50 devem ser consultados através de Lemos et al. (2019).

Trabalhos futuros serão conduzidos objetivando-se a redução do número de parâmetros, mantendo, no entanto, taxas aceitáveis de acerto para a aplicação da detecção e subsequente preensão em malha fechada, além disso, a técnica será testada e validada em ambientes reais.

AGRADECIMENTOS

Este estudo recebeu apoio financeiro do SEPINMCTI, no âmbito da 4ª Chamada Coordenada BR-EU no CIT e do programa de pesquisa e inovação European Unions Horizon 2020 no âmbito do Contrato de Doação n° 777096.

Os autores agradecem à FAPESB (Fundação de Amparo à Pesquisa do Estado da Bahia) pelo apoio financeiro. Este estudo foi financiado também pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

REFERÊNCIAS

Beeson, P. and Ames, B. (2015). Trac-ik: An open-source library for improved solving of generic inverse kinematics. In *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 928–935. IEEE.

Bradski, G. (2000). The opencv library. *Dr Dobb's J. Software Tools*, 25, 120–125.

Caldas, N., de Sousa, J.P., Alcalá, S.G., Frazzon, E., and Moniz, S. (2019). A simulation approach for spare parts supply chain management. In *Proceedings of the International Conference on Industrial Engineering and Operations Management, July*, 23–26.

de Brito, F.M., da Cruz Júnior, G., Frazzon, E.M., Basto, J.P., and Alcalá, S.G. (2019). An optimization model for the design of additive manufacturing supply chains. In *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, volume 1, 881–885. IEEE.

Goldberg, K., Mirtich, B.V., Zhuang, Y., Craig, J., Carlisle, B.R., and Canny, J. (1999). Part pose statistics: Estimators and experiments. *IEEE Transactions on Robotics and Automation*, 15(5), 849–857.

Hara, K., Vemulapalli, R., and Chellappa, R. (2017). Designing deep convolutional neural networks for continuous object orientation estimation. *arXiv preprint arXiv:1702.01499*.

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

Jiang, Y., Moseson, S., and Saxena, A. (2011). Efficient grasping from rgbd images: Learning using a new rectangle representation. In *2011 IEEE International Conference on Robotics and Automation*, 3304–3311. IEEE.

Johns, E., Leutenegger, S., and Davison, A.J. (2016). Deep learning a grasp function for grasping under gripper pose uncertainty. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4461–4468. IEEE.

Katz, D., Kenney, J., and Brock, O. (2008). How can robots succeed in unstructured environments. In *In Workshop on Robot Manipulation: Intelligence in Human Environments at Robotics: Science and Systems*. Citeseer.

Kober, J. and Peters, J. (2010). Imitation and reinforcement learning. *IEEE Robotics & Automation Magazine*, 17(2), 55–62.

Koenig, N. and Howard, A. (2004). Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*, volume 3, 2149–2154. IEEE.

Konidaris, G., Kuindersma, S., Grupen, R., and Barto, A. (2012). Robot learning from demonstration by constructing skill trees. *The International Journal of Robotics Research*, 31(3), 360–375.

Kragic, D. and Christensen, H.I. (2003). Robust visual servoing. *The international journal of robotics research*, 22(10-11), 923–939.

Lemos, C.B., Farias, P.C., Simas Filho, E.F., and Conceição, A.G. (2019). Convolutional neural network based object detection for additive manufacturing. In *2019 19th International Conference on Advanced Robotics (ICAR)*, 420–425. IEEE.

Lenz, I., Lee, H., and Saxena, A. (2015). Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5), 705–724.

Levine, S., Pastor, P., Krizhevsky, A., and Quillen, D. (2016). Learning hand-eye coordination for robotic grasping with large-scale data collection. In *International Symposium on Experimental Robotics*, 173–184. Springer.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., and Berg, A.C. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision*, 21–37. Springer.

Mahler, J., Liang, J., Niyaz, S., Laskey, M., Doan, R., Liu, X., Ojea, J.A., and Goldberg, K. (2017). Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. *arXiv preprint arXiv:1703.09312*.

Maitin-Shepard, J., Cusumano-Towner, M., Lei, J., and Abbeel, P. (2010). Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding. In *2010 IEEE International Conference on Robotics and Automation*, 2308–2315. IEEE.

Morrison, D., Corke, P., and Leitner, J. (2018). Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach. *arXiv preprint arXiv:1804.05172*.

Pinto, L. and Gupta, A. (2016). Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *2016 IEEE international conference on robotics and automation (ICRA)*, 3406–3413. IEEE.

Prattichizzo, D. and Trinkle, J.C. (2008). Grasping. *Springer handbook of robotics*, 671–700.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A.Y. (2009). Ros: an open-source robot operating system. In *ICRA workshop on open source software*, 5. Kobe, Japan.

Ribeiro, E.G. and Grassi, V. (2019). Fast convolutional neural network for real-time robotic grasp detection. In *International Conference on Advanced Robotics (ICAR)*.

Robots, U. (2019). Ur5 collaborative robot arm | flexible and lightweight robot arm. `https://www.universal-robots.com/products/ur5-robot/`. Accessed: 2019-10-15.

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

Viereck, U., Pas, A.t., Saenko, K., and Platt, R. (2017). Learning a visuomotor controller for real world robotic grasping using simulated depth images. *arXiv preprint arXiv:1706.04652*.