2018

Review of deep learning methods in robotic grasp detection

Shehan Caldera
Edith Cowan University

Alexander Rassau
Edith Cowan University

Douglas Chai
Edith Cowan University

10.3390/mti2030057

Caldera, S., Rassau, A., & Chai, D. (2018). Review of deep learning methods in robotic grasp detection. *Multimodal Technologies and Interaction, 2*(3), 57. Available here

Multimodal Technologies and Interaction

Review

Review of Deep Learning Methods in Robotic Grasp Detection

Shehan Caldera *, **Alexander Rassau and Douglas Chai**

School of Engineering, Edith Cowan University, 270 Joondalup Drive, Joondalup, WA 6027, Australia; a.rassau@ecu.edu.au (A.R.); d.chai@ecu.edu.au (D.C.)
* Correspondence: shelessa@our.ecu.edu.au

Received: 31 May 2018; Accepted: 3 September 2018; Published: 7 September 2018

Abstract: For robots to attain more general-purpose utility, grasping is a necessary skill to master. Such general-purpose robots may use their perception abilities to visually identify grasps for a given object. A grasp describes how a robotic end-effector can be arranged to securely grab an object and successfully lift it without slippage. Traditionally, grasp detection requires expert human knowledge to analytically form the task-specific algorithm, but this is an arduous and time-consuming approach. During the last five years, deep learning methods have enabled significant advancements in robotic vision, natural language processing, and automated driving applications. The successful results of these methods have driven robotics researchers to explore the use of deep learning methods in task-generalised robotic applications. This paper reviews the current state-of-the-art in regards to the application of deep learning methods to generalised robotic grasping and discusses how each element of the deep learning approach has improved the overall performance of robotic grasp detection. Several of the most promising approaches are evaluated and the most suitable for real-time grasp detection is identified as the one-shot detection method. The availability of suitable volumes of appropriate training data is identified as a major obstacle for effective utilisation of the deep learning approaches, and the use of transfer learning techniques is proposed as a potential mechanism to address this. Finally, current trends in the field and future potential research directions are discussed.

Keywords: deep learning; deep convolutional neural networks; dcnn; convolutional neural networks; cnn; robot learning; transfer learning; robotic grasping; robotic grasp detection; human-robot collaboration

1. Introduction

Recent advancements in robotics and automated systems have led to the expansion of autonomous capabilities and more intelligent machines being utilised in ever more varied applications [1,2]. The capability of adapting to changing environments is a necessary skill for task generalised robots [3,4]. Machine learning plays a key role in creating such general-purpose robotic solutions. However, most robots are still developed analytically, based on expert knowledge of the application background. Even though this is considered an effective method, it is an arduous and time-consuming approach, and has significant limitations for generalised applicability. Due to the recent successful results of deep learning methods in computer vision and robotics applications, many robotics researchers have started exploring the application of deep learning methods in their research.

The type of learning that is applied varies according to the feedback mechanism, the process used for training data generation, and the data formulation. The learning problem can vary from perception to state abstraction, through to decision making [5]. Deep learning, a branch of machine learning, describes a set of modified machine learning techniques that, when applied to robotic systems, aims to enable robots to autonomously perform tasks that come naturally to humans. Inspired by the biological nervous system, a network of parallel and simultaneous mathematical operations are performed directly on the available data to obtain a set of representational heuristics between the input and output data. These heuristics are then used in decision making. Deep learning models have proven effective in diverse classification and detection problems [6–8] and there is a great deal of interest in expanding their utilisation into other domains.

The grasp or grasping pose describes how a robotic end-effector can be arranged to successfully pick up an object. The grasping pose for any given object is determined through a grasp detection system. Any suitable perception sensors including cameras or depth sensors can be used to visually identify grasping poses in a given scene. Grasp planning relates to the path planning process that is required to securely grab the object and maintain the closed gripper contacts to hold and lift the object from its resting surface [9]. Planning usually involves the mapping of image plane coordinates to the robot world coordinates for the detected grasp candidate. The control system describes certain closed-loop control algorithms that are used to control the robotic joints or degrees of freedom (DOF) to reach the grasping pose while maintaining a smooth reach [10].

This paper reviews deep learning approaches for the detection of robotic grasping poses for a given object captured in an image. The paper is organised as follows: Section 2 provides the relevant background information into robot grasping; Section 3 describes how robotic grasps have been represented in the literature; Section 4 identifies popular datasets and training methodologies; Section 5 introduces the main convolutional neural network approaches for detecting robotic grasps and explores current trends for neural network architectures; and finally the conclusion and recommendations for future research directions can be found in Section 6.

2. Background

Traditional analytical approaches, also known as hard coding, involve manually programming a robot with the necessary instructions to carry out a given task. These control algorithms are modelled based on expert human knowledge of the robot and its environment in the specific task. The outcome of this approach explains a kinematic relationship between the parameters of the robot and its world coordinates. Ju et al. [11] suggested that the kinematic model helps in further optimising the control strategies. However, direct mapping of results from a kinematic model to the robot joint controller is inherently open-loop and is identified to cause task space drifts. Therefore, they have, in addition, recommended the use of closed loop control algorithms to address these drifts [2].

Even though such hard coded manual teaching is known to achieve efficient task performance, such an approach has limitations; in particular, the program is restricted to the situations predicted by the programmer, but in cases where frequent changes of robot programming is required, due to changes in the environment or other factors, this approach becomes impractical [4]. According to Ju et al. [2], unstructured environments remain a large challenge for intelligent robots that would require a complex analytical approach to form the solution. While deriving of models requires a great deal of data and knowledge of the physical parameters relating to the robotic task, use of more dynamic robotic actuators make it nearly impossible to model the physics, thus they conclude that manual teaching is an efficient but exhaustive approach [2]. In such cases, empirical methods will provide an increased cognitive and adaptive capability to the robots, while reducing or completely removing the need to manually model a robotic solution [3]. Early work in empirical methods takes a classical form that explores the adaptive and cognitive capability of robots to learn tasks from demonstration. Non-linear Regression techniques, Gaussian process, Gaussian mixture models, and Support Vector Machines are some of the popular techniques related to this context [12]. Although these techniques have provided some level of cognition for the robots, the task replication is limited to the demonstrated tasks.

Deep learning has recently made significant advancements in the application of computer vision, scene understanding, robotic arts, and natural language processing [10,13]. Due to the convincing results that have been achieved in the scope of computer vision, there is an increasing trend towards implementation of deep learning methods in robotics applications. Many recent studies show that the unstructured nature of a generalised robotics task makes it significantly more challenging. However, to advance the state-of-the-art of robotic applications, it is necessary to create a generalised robotic solution for various industries such as offshore oil rigs, remote mine sites, manufacturing assembly plants, and packaging systems where the work environments and scenarios can be highly dynamic. A desired primary ability for these general-purpose robots is the ability to grasp and manipulate objects to interact with their work environment. The visual identification and manipulation of objects is a relatively simple task for humans to perform, but for a robot this is a very challenging task that involves perception, planning, and control [10,14]. Grasping can enable the robots to manipulate obstacles in the environment or to change the state of the environment if necessary. Early work such as [15,16] show how far researchers have advanced the research methods in robotic grasping. These studies discuss the early attempts of grasping novel objects using empirical methods.

Object grasping is challenging due to the wide range of factors such as different object shapes and unlimited object poses. Successful robotic grasping systems should be able to overcome this challenge to produce useful results. Unlike robots, humans can almost immediately determine how to grasp a given object. Robotic grasping currently performs well below human object grasping benchmarks, but is being continually improved given the high demand. A robotic grasping implementation has the following sub-systems [10]:

- **Grasp detection sub-system**: To detect grasp poses from images of the objects in their image plane coordinates
- **Grasp planning sub-system**: To map the detected image plane coordinates to the world coordinates
- **Control sub-system**: To determine the inverse kinematics solution of the previous sub-system

We identify the grasp detection sub-system as the key entry point for any robotic grasping research and aim to review current deep learning methods in grasp detection through the subsequent sections of this paper. A popular deep learning method that has been applied in most related literature is the Convolutional Neural Network (CNN) or sometimes referred to as the Deep Convolutional Neural Network (DCNN) due to the heavy involvement of convolutional layers in their architectures. It is evident that there are two approaches to apply a CNN to a problem:

1. Create an application-specific CNN model
2. Utilise a complete or part of a pre-existing CNN model through transfer learning

Creating a proprietary application-specific CNN model requires a deep understanding of the concept and a reasonable level of experience with CNNs. Therefore, most researchers that implement CNNs in their grasp detection work have opted for transfer learning given the reduced number of parameters to be dealt with. Training of such a CNN requires a large volume of data [17]. The data can be labelled or unlabelled depending on whether a supervised or unsupervised training method is used. Training is the process of tuning the network parameters according to the training data. Some studies [10,18–20] focus on simplifying the problem of grasp detection and build on the transfer learning model to improve the results. While there are several platforms to implement deep learning algorithms, most studies have used Tensorflow [21], Theano [22], or Matlab [6]. With the recent advancements of software applications and programming languages, there are now more streamlined tools such as Keras [23], Caffe [24] or DarkNet [25] to implement the same functionality of former deep learning frameworks but in an easier and more efficient way. Even though most recent deep learning approaches for robotic grasping follow purely supervised learning, software platforms such as NVIDIA ISAAC [26] encourage unsupervised learning methods with the support of virtual simulation capabilities.

3. Robotic Grasp Detection

Grasp detection is identified as the ability to recognise the grasping points or the grasping poses for an object in any given image [27]. As shown in Figures 1 and 2, a successful grasp describes how a robotic end-effector can be orientated on top of an object to securely hold the object between its gripper and pick the object up. As humans, we use our eyesight to visually identify objects in our vicinity and find out how to approach them in order to pick them up. In a similar manner, visual perception sensors on a robotic system can be used to produce information on the environment that can be interpreted into a useful format [19]. A mapping technique is necessary to classify each pixel of the scene on the basis of belonging or not belonging to a successful grasp. Recent robotic grasping work has used several different definitions for successful grasp configurations [10,18–20]. In this regard, a representation or a definition of a good grasp is necessary. This section reviews some of the promising grasp representations and their method of detection. Section 3.1 discusses several grasp configurations and how they are represented in images. Section 3.2 discusses how these grasp representations are detected from images. While this section aims to provide a brief overview of how grasps can be represented and detected in deep learning applications, the reader is directed towards References [9,28] for a more comprehensive description.

3.1. Grasp Representation

In most of the earlier works, grasps were represented as points on images of actual scenes or from 3D mesh models based on simulations. Using a supervised learning approach, Saxena et al. [15] investigated a regression learning method to infer the 3D location of a grasping point in a Cartesian coordinate system. They used a probabilistic model over possible grasping points while considering the uncertainty of the camera position. Extending their investigation, they had discretised the 3D workspace in order to find the grasping point g, given by $g = (x, y, z)$. They reported that by using two or more images captured from different angles it would simplify the grasp point inference and also referred to the smaller graspable regions on the images as grasp points as shown in Figure 1 [16]. In their reinforcement learning approach for grasp point detection, Zhang et al. [29] simply defined a grasp as a point in a 2D image plane. A major drawback of such point defined grasps, however, was that it only determined where to grasp an object and it did not determine how wide the gripper had to be opened or the required orientation for the gripper to successfully grasp the object.

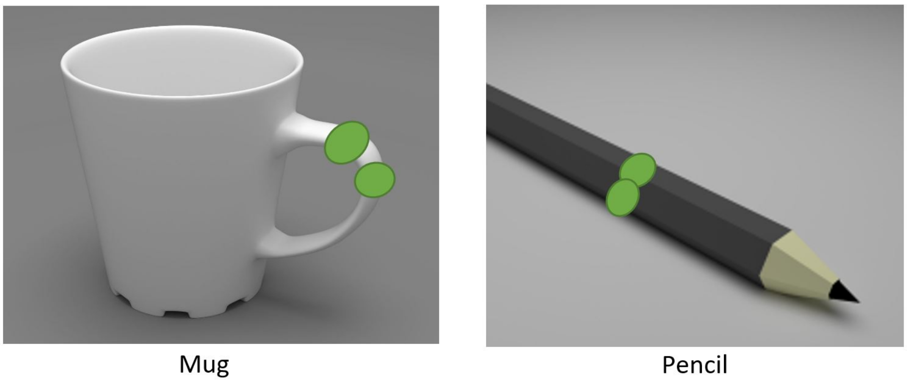


Figure 1. Example of the grasp point representation shown on 3D models of a mug and a pencil.

As a way to overcome this limitation, another popular grasp representation that has been proposed is the oriented rectangle representation that was used in [10,18–20,30,31]. According to Jiang et al. [30], their grasping configuration has a seven-dimensional representation containing the information of a **Grasping point**, **Grasping orientation**, and **Gripper opening width**. In world coordinates, their grasp representation, G, is stated as $G = (x, y, z, \alpha, \beta, \gamma, l)$. Their grasp representation is shown in Figure 2a. The red lines represent the opening or closing width of the gripper along with the direction of the motion. The blue lines represent the parallel plates of the end-effector.

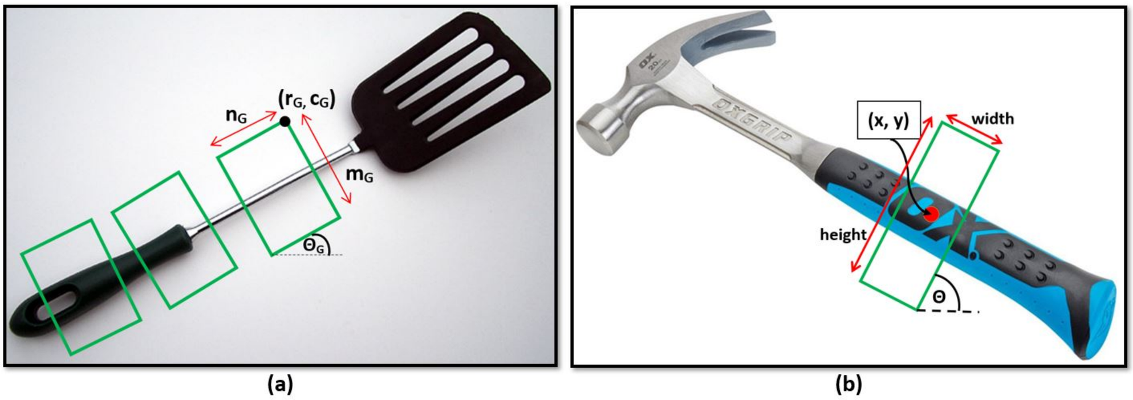


Figure 2. Grasping rectangle representations. (**a**) The representation by Jiang et al. [30]: Top vertex (r_G, c_G), length m_G, width n_G and its angle from the x-axis, θ_G for a kitchen utensil. There can be multiple ground truth grasps defined as shown. (**b**) The simplified representation by Redmon et al. [19] for a hammer, showing its grasp centre at (x, y) oriented by an angle of θ from its horizontal axis. The rectangle has a width and height of w and h respectively.

Simplifying the previously introduced seven dimensional grasp rectangle representation from [30], Lenz et al. [18] proposed a five dimensional representation. This was based on the assumption of a good 2D grasp being able to be projected back to 3D space. While they failed to evaluate their approach, Redmon et al. [19] confirmed the validity of the method with their own results. They further supported the statements by Jiang et al. [30] and Lenz et al. [18] that detection of grasping points in this manner was analogous to object detection methods in computer vision but with an added term for the gripper orientation. Adapting the method of [18,30], they also presented a slightly updated representation of a grasp rectangle, as shown in Figure 2b. This modified rectangle grasp representation has been used in a number of later publications demonstrating its usefulness [10,20,31]. In their work one using deep learning algorithms for robotic grasping detection, Kumra et al. [10] used the grasp rectangle originally proposed by Redmon et al. [19]. A very recent online project page [20] has cited the same Redmon grasp rectangle.

Despite not employing a CNN in their dictionary learning method for grasp detection, Trottier et al. [32] used the same grasp rectangle that was used in previous methods, most likely due to its similarity to the object detection representations that were widely used at that time. In another study conducted by Park et al. [33], the same grasp rectangle representation was again used. They argued that the grasp rectangle was analogous to the standard object detection bounding box with the added feature being the orientation. In their novel classification method for grasp detection, Zhou et al. [34] used a similar five-element grasp rectangle representation following the previous work in [10,18–20,31–33,35]. Wang et al. [36] proposed a minor variation to this approach that differed simply by excluding the parameter for gripper plate height (h). They argued that this parameter can be controlled in the robotic set-up configurations thus the authors used a four-element grasp representation of $G = (x, y, \theta, w)$.

Another grasp representation introduced in more recent research is the combined location and orientation representation. In [37], the authors used the simple $G = (x, y, \theta)$ representation that dropped the dimensional parameters (h, w). The dimensional parameters provided a sense of the physical limitations for certain end-effectors. Similar representations are used in [38,39]. This representation described a grasp in a 2D image plane. This representation was improved by Calandra et al. to include the 3D depth information by adding the z coordinates to the representation, resulting in a grasp representation, $G_z = (x, y, z, \theta)$ [40,41]. The G_z grasp representation was also used by Murali et al. [42] in their approach to detect robotic grasps through the use of tactile feedback and visual sensing.

From the three grasp representations described in this section, the Rectangle representation can be identified as the most commonly used for any grasp detection applications. While a detailed analysis of the relative suitability of the approaches has not been conducted, the literature survey suggests that any preference is application specific. Table 1 summarises the characteristics of each grasp representation type with regards to depth, pose, and physical limitations of the end-effector. Lenz et al. [18] argued that, in most cases, the depth information can be manually controlled specific to the application. Therefore the rectangle representation can be selected as the most suitable grasp representation in most cases.

Table 1. Comparison between different grasp representations.

Type	Grasp Parameters	Depth	Pose	Physical Limitations
Point representation	(x, y)	No	No	No
	(x, y, z)	Yes	No	No
Location + Orientation representation	(x, y, θ)	No	Yes	No
	(x, y, z, θ)	Yes	Yes	No
Rectangle representation	(x, y, θ, h, w)	No	Yes	Yes

Pixel-wise grasp representations were used when structured grasps were not useful. Ku et al. [43] used convolutional layer activations to find the anthropomorphic grasping points in images. They created a mask that represented the grasping points for the robotic index finger and the thumb. The mask contains all pixels that are part of the grasp. Their method only works for cuboid and cylindrical shaped objects. They reported that only one trial failed from the complete set of 50 trials, and they have managed to achieve an average success rate of 96%.

In applications where simple object localisation translates back to the simple pick-up points in the 2D image plane, dense captioning was used to localise objects in order to pick them up. In their work for Amazon Picking Challenge [44], Schwarz et al. [45] used the popular dense captioning [46] to localise objects in images. Dense captioning provides a textual description of each region of the image and it can be used to identify or localise objects in an image. During the testing, they successfully picked up 10 objects out of the 12 test object set, and their fine-tuned system responded within 340 milliseconds during the testing.

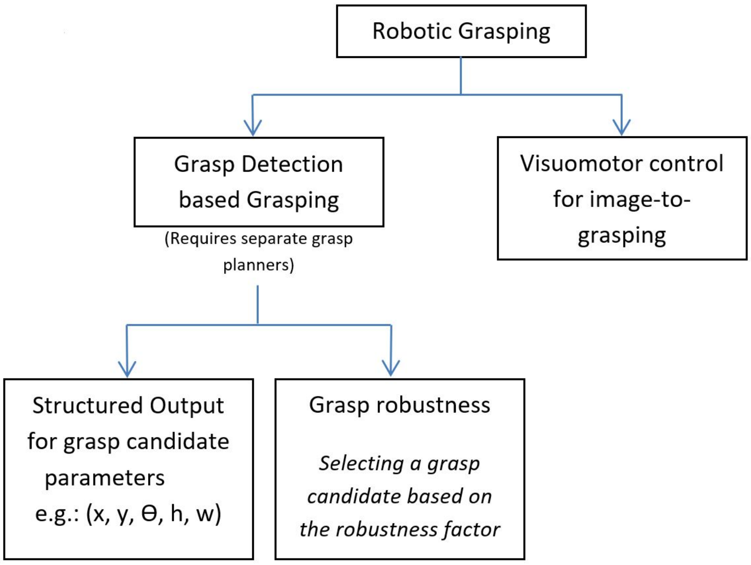
These graspable region representations are widely used in picking or sorting objects in clutter when there are no particular requirements with respect to the order in which objects are picked up. More structured grasp representations are generally employed in conjunction with object recognition in order to grasp the identified objects [19]. The works discussed in this section demonstrate that a consistent grasp representation method must be adopted in order to start working with learning algorithms for the detection of robotic grasps. The ground truth labels should have the optimal number of parameters to represent a grasp while ensuring that it is not over-defined. The five dimensional grasp representation originally presented by Lenz et al. [18] for a 2D image should, thus, be further explored.

3.2. Grasp Detection

The conventional analytical method of robotic grasp detection is performed on the premise that certain criteria such as object geometry, physics models, and force analytics are known [9]. The grasp detection applications are built based on a model developed with this information. The modelling of such information is often challenging due to the current fast-changing industry requirements. An alternative approach is to use empirical methods, also known as data-driven approaches, that rely on previously known successful results. These methods are developed using existing knowledge of object grasping or by using simulations on real robotic systems [28]. A major drawback of analytical methods is that they rely on the assumption that the object parameters are known, therefore they cannot be used for a generalised solution [28]. There are two types of empirical approaches in robotic grasp detection:

1. Methods that use learning to detect grasps and use a separate planning system for grasp planning
2. Methods that learn a visuomotor control policy in a direct image-to-action manner

In the literature, direct grasp detection has been carried out using two different techniques. The most popular one is to detect structured grasp representations from images. An alternative approach is to learn a grasp robustness function. Both techniques require a separate grasp planning system to execute the grasp. During the last few years, there has been a growing interest into learning a visuomotor control policy using deep learning. The illustration in Figure 3 further clarifies the terminology. The introduction of tools such as NVIDIA Isaac [26] has enabled the extensive use of reinforcement learning in simulated environments with domain adaptation. These visuomotor control policy learning methods do not require a separate grasp planning system.


Robotic Grasping

Grasp Detection based Grasping

(Requires separate grasp planners)

Visuomotor control for image-to-grasping

Structured Output for grasp candidate parameters e.g.: (x, y, ϴ, h, w)

Grasp robustness

Selecting a grasp candidate based on the robustness factor

Figure 3. Categorisation of robotic grasping and grasp detection.

The most popular method for structured grasp detection was the sliding window approach proposed by Lenz et al. [18]. In their approach, a classifier is used to predict if a small patch of the image contains a potential grasp. The image is divided into a number of small patches and each patch is run through the classifier in an iterative process. The patches that contain higher ranking grasps are considered as candidates and pushed as outputs. This method yielded a detection accuracy of 75% and a processing time of 13.5 s per image. Similar results were reported from the studies by Wang et al. [47] and Wei et al. [48] who followed a similar approach. Guo et al. [35] used the reference rectangle method to identify graspable regions of an image. This method was adapted from region proposal neural networks [49]. The locations of the reference rectangle were identified using the sliding window approach. Due to the repetitive scanning method for identifying graspable regions of images, this method was largely considered unsuitable where a real-time detection speed is necessary. As an alternative, Redmon et al. [19] proposed the one-shot detection method.

In most one-shot detection methods, a direct regression approach for predicting a structured grasp output is used. In these approaches, the structured output represents the oriented grasp rectangle parameters in the image plane coordinates. In the first one-shot detection approach, the authors argued that a faster and more accurate method was necessary and proposed to use transfer learning techniques to predict grasp representation from images [19]. They reported a detection accuracy of 84.4% in 76 milliseconds per image. This result produced a large performance boost compared to the then state-of-the-art method, the sliding window. The one-shot detection method assumed that each image contained one graspable object and predicted one grasp candidate as opposed to the iterative scanning process of the sliding window approach. Following a similar strategy, Kumra et al. [10] reported an improvement whereby a detection accuracy of 89.21% for their multi-modal grasp detector with a processing speed of 100 milliseconds per image was achieved. They explained that it was due to the deeper network architecture that they had used in their experimentations. Therefore, it was evident that most work in one-shot detection followed deep transfer learning techniques to use pre-trained neural network architectures [20,31].

Although the preferred method for one-shot detection is the direct regression of the grasp representation, there were numerous occasions where the combined classification and regression techniques were employed for one-shot detection. While arguing that the orientation predictions of a structured grasp representation lay in a non-Euclidean space, where standard regression loss (L_2) had not performed well, Chu et al. [50] proposed to classify the orientation among 19 different classes in the range of $[0°, 360°]$. They used a direct regression method to predict the bounding box of the grasp. The authors reported a detection accuracy of 94.4% with RGB images. Building on the concept by Guo et al. [35], Zhou et al. [34] proposed to use the anchor boxes for predefined regions of the images. Each image was divided into $N \times N$ regions. The orientation of the anchor box was classified between k classes. The authors argued that the k can be a variable integer. By default it was set to $k = 6$. The angles ranged between $[-75°, 75°]$. They achieved a detection accuracy of 97.74% for their work. The literature reasoned that these improvements were achieved as it was easier to converge to a classification during the training and the associated errors were minimum, but such a classification would limit the output to a predefined set of classes [37].

Learning a grasp robustness function also had been the central idea of many studies in deep grasp detection. The researchers used this function to identify the grasp pose candidate with the highest score as the output. Grasp robustness described the grasp probability of a certain location or an area of an image [27]. Binary classification was a well researched technique for this approach that classified the grasp points as valid or invalid (1 or 0). Park et al. [33] used a multi stage spatial transformer network to predict the success of a grasp candidate. They reported a grasp detection accuracy of 89.60% in 23 milliseconds per image [33]. Using end-to-end learning, Ten Pas et al. [51] performed a binary classification to identify graspable regions in a dense clutter of objects. They presented a 77% detection accuracy with passive point cloud data. In their work, Lu et al. [52] performed a CNN based grasp probability study to achieve a detection accuracy of 75.6% for previously unseen novel objects and 76.6% for previously seen objects during the training. In addition to the applicability of this method to cluttered objects, researchers concluded its usefulness when partial information was present [33].

A method to learn an optimal grasp robustness function was proposed by Mahler et al. [27]. They considered the robustness as a scalar probability in the range of $[0, 1]$. The authors compiled a dataset known as *Dex-Net 2.0* with 6.7 million point clouds and analytic grasp quality metrics with parallel-plate grippers planned using robust quasi-static grasp wrench space analysis on a dataset of 1500 3D object mesh models. They further trained a grasp quality convolutional network (GQ-CNN) that was used to learn a robustness metric for grasp candidates. They tested their CNN with their dataset which achieved an accuracy of 98.1% for grasp detection. Robust grasp detection is explored in [53]. Johns et al. reported that they achieved a grasp success rate of 75.2% with minor gripper pose uncertainties and 64.8% with major gripper pose uncertainties. They described the gripper pose uncertainties to be associated with varying shapes and contours associated with the objects.

The literature survey suggested that a direct mapping of images to robot actions could be predicted by learning a visuomotor control policy. This method would not require a separate grasp planning system and, thus was also considered a pixel-to-action method. Using their previous findings [27], Mahler et al. [39] proposed a method to find deep learnt policies to pick objects from clutter. The authors reported that by using a transfer learning technique with their previous findings in [27], they achieved a grasp detection accuracy of 92.4%. When they tested their learnt policies on robotic grasping, they achieved a success rate of 70% with five trials for each of 20 objects of the test dataset [39]. The winning team from the Amazon Picking Challenge 2017 [44], Zeng et al. [54] proposed a visuomotor control policy prediction method for images of objects in clutter. The authors proposed an action space with four individual actions: (a) suction down; (b) suction side; (c) grasp down; and (d) flush grasp. They reported a maximum accuracy of 96.7% for grasping and 92.4% for suction with the *Top-1* confidence percentile [54]. Zhang et al. [29] proposed a method to use reinforcement learning [55] to determine the action to extend the robot end-effector to a point in a 2D image plane. Closely following the proposed deep Q network by Mnih et al. [55], the authors managed to adapt it to a robotic system using synthetic images. In testing, their system achieved a 51% success rate in reaching the target point [29]. They further concluded that these results were largely affected by not having an optimal domain adaptation from synthetic to realistic scenes.

In summary, it is identified that most of the work in grasp pose detection has focused on the detection of a structured grasp representation. For structured grasp representation detection, the one-shot detection method has achieved state-of-the-art results according to recent research studies. Most of one-shot detectors use deep transfer learning techniques to use pre-trained deeper convolutional networks to predict the grasp candidates from images. This section has introduced the popular grasp detection methods and the reader is directed to Section 5 for an in-depth discussion of the convolutional network approach for grasp detection.

4. Types and Availability of Training Data

Research has consistently shown that deep learning requires a large volume of labelled data to effectively learn the features during the training process [17]. This requirement is also apparent in supervised learning methods in robotic grasp detection [10,18–20]. In recently published work, researchers either use training data from a third party or introduce their own application specific proprietary data sources or methods to automate the data generation [27,37]. Johns et al. [53] highlighted that the major challenge with deep learning is the need for a very large volume of training data, thus they opted to generate and use simulated data for the training process. Another challenge of training deep neural networks is the lack of domain specific data as mentioned by Tobin et al. [14]. They proposed a method to generate generalised object simulations to address this challenge, although it has not yet been proven how effective the results can be. For real-time applications, use of simulated data and the availability of 3D object models is not practically achievable [10,19]. As a way to overcome this, there are reports of network pre-training as a solution when there is limited domain specific data [10,19,20].

Brownlee [56] specified that annotations of the available data will be more important if the learning was purely supervised and less important for unsupervised learning. He further described the importance of the three subsets of data for training, validation, and testing. In [10,19,20,31], the authors had followed the same argument. A comprehensive explanation can also be found in [57]. The literature survey suggests that, in addition to larger training datasets, domain specific data are necessary for effective results.

4.1. Multi-Modal Data

Use of multi-modal data has become popular in many research studies into robotic grasp pose detection. Early work from Saxena et al. [15] stated that most grasping work assumed prior knowledge of the 2D or 3D model of the object to be grasped, but such approaches encounter difficulties when attempting to grasp novel objects. The authors experimented with depth images for five different objects in their training. They reported the grasp success rates for basic objects such as mugs, pens, wine glasses, books, erasers, and cellphones. An overall success rate of 90% with a mean absolute error of 1.8 cm was reported.

Following this work, Jiang et al. [30] scaled the problem space to 194 images of nine classes. They stated that the availability of multi-modal data could be useful in identifying edges and contours in the images to clearly differentiate graspable regions. Lenz et al. [18] supported the same claim in their work that used multi-modal RGB-D data. In a few recent transfer learning applications, the authors used the multi-modality in a way that overcame the three-channel data limitation with existing pre-trained CNNs. The authors in [19,20,31] replaced the Blue channel in RGB images with depth disparity images and created 3-channel RG-D images.

Kumra et al. [10] proposed a novel method to use pre-trained DCNN architectures with 3-channel input limitations. Instead of replacing the Blue channel, the authors trained two convolutional networks for RGB and depth features individually. They used a similar encoding to [58] to create three-channel depth disparity images. They further reported that proper pre-training for the depth CNN was not available since all of the pre-trained networks were pre-trained on RGB images. More streamlined methods (e.g., [59]) would help in this endeavour. While most of these works used visual information, there were some reported studies that had used tactile sensing with deep learning approaches in grasp detection.

In [42], Murali et al. explored using tactile sensing to complement the use of visual sensors. This method involved a re-grasping step to accurately grasp the object. They reported a success rate of 85.92% with a deep network and 84.5% with an SVM. A similar approach was followed by Calandra et al. [40] in their work on using tactile sensing in robotic grasp detection.

Some researchers had also experimented with uni-modal data as well. Kumra et al. [10] trained their neural network with uni-modal RGB images and achieved an accuracy of 88.84%.

Our literature survey indicates there are several types of multi-modalities involved in grasp pose detection research with the most popular one being the RGB-D data. Evidence suggests that the added benefit of edge and contour information in RGB-D images has an advantage over uni-modal RGB images. There is not yet enough evidence to suggest whether tactile sensing has an added advantage over depth imaging for grasp detection work using RGB images. A major challenge with respect to using RGB-D data, however, is the access to suitable training datasets.

4.2. Datasets

Goodfellow et al. [17] stated that the performance of a simple machine learning algorithm relied on the amount of training data as well as the availability of domain specific data. The recent publications suggested that the availability of training data is a prevailing challenge for this learning method. Some researchers combined datasets to create a larger dataset while others collected and annotated their own data.

4.2.1. Pre-compiled datasets

The Cornell Grasp Dataset (CGD) from [60] is a popular grasp dataset that was compiled for most transfer learning approaches in robotic grasping [10,19,20,31]. The CGD was created with grasp rectangle information for 240 different object types and it contained about 885 images, 885 point clouds and about 8019 labelled grasps including valid and invalid grasp rectangles. A sample set of images is shown in Figure 4. The grasps were specifically defined for the parallel plate gripper found on many robotic end-effectors. The CGD appeared in a number of research studies during the recent past, which might suggest that it has a reasonable diversity of examples for generalised grasps [19]. The recent trend of using RGB-D for learning to predict grasps was covered with the CGD dataset through the inclusion of point cloud data by its creators. Lenz et al. argued that having the depth information would result in a better depth perception for an inference system that was trained on depth data [18]. A sense of good and bad grasps was also necessary to differentiate a better grasp from the alternatives [18,19]. Therefore, the CGD could be selected as a suitable dataset for its quality and adaptability. The CGD was extensively used in [10,18–20,31,33,35,50].

In the grasp detection work by Wang et al. [36], the authors used the Washington RGB-D dataset [58] for its rich variety of RGB-D images. The authors self-annotated as they preferred to combine the resulting dataset with the CGD. The authors further stated that the combined Washington data instances of 25,000 with the 885 instances from the CGD would help in pre-training a deep network [36].

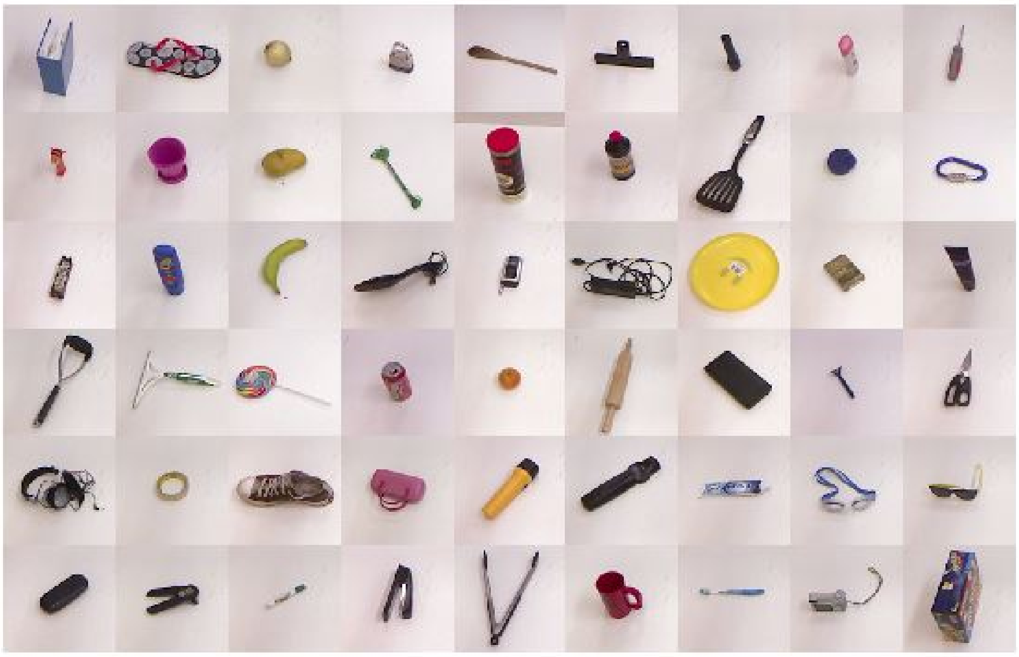


Figure 4. Sample of Cornell Grasp dataset [60].

4.2.2. Collected Datasets

When application-specific data were necessary, researchers provided intuitive methods for data collection. Murali et al. [42] used a previously learned grasping policy to collect valid grasp data and performed random grasps to collect invalid grasp data. They have collected data for 52 different objects. Calandra et al. [40] collected data from 9269 grasp trials for 106 unique objects. Pinto et al. [37] stated how time consuming it was to collect data for robotic grasping and proposed a novel approach inspired from reinforcement learning. Their approach would predict centre points for grasps from a policy learned using reinforcement learning and the orientation was classified for 18 different classes using grasp probability. The authors scaled the data collection to 50,000 grasp trials using 700 robotic hours. They used a Mixture of Gaussians (MOG) background subtraction that identified graspable regions in images to avoid random object-less spaces in images. Levine et al. [61] further improved this approach through the collection of grasping data from nearly 900,000 grasp trials using 8 robots.

4.2.3. Domain Adaptation and Simulated Data

As pointed out by Tobin et al. [14], most applications lacked domain specific data. While arguing the importance of a large volume of domain specific data, the authors proposed a method to use physics simulations to generate domain specific data using 3D mesh models for a set of primitive shapes. Most of the work that has used simulation and 3D model data relies on domain adaptation to its real world equivalent set of objects. Bousmalis et al. [62] conducted several experiments to verify the domain adaptation capability of a deep grasp detection application that was trained on 3D mesh models randomly created by the authors. The authors randomly mixed simulated data with realistic data to compile a dataset of 9.4 million data instances. Using this a grasp success rate of 78% was achieved. In a similar approach, Viereck et al. [38] proposed a method for learning a closed-loop visuomotor controller from simulated depth images. The authors generated about 12,500 image-action pairs for the training. They reported a grasp pose detection success rate of 97.5% for objects in isolation, and 94.8% for objects in clutter. Mahler et al. [27] suggested populating a dataset containing physics based analyses such as caging, grasp wrench space (GWS) analyses and grasp simulation data for different types of object shapes and poses. They further suggested that cloud computing could be leveraged to train a convolutional neural network with this dataset that would in turn, predict a robustness metric for a given grasp instead of directly predicting a grasp. The proposed dataset was called Dex-Net 2.0 [63] and contained about 6.7 million point clouds and analytic grasp quality metrics with parallel-jaw grasps planned using robust quasi-static GWS analysis on a dataset of 1500 3D object models [27].

4.2.4. Summary

Mahler et al. [27] concluded that human annotation is a tedious process that requires months of work and the simulations would have to be run for a large number of iterations on a robotic system. With the limited availability of domain specific data, Redmon et al. [19] proposed to use pre-training. Pre-training assumes that by using the weights of the convolutional overhead of a CNN model that was trained on a large dataset such as ImageNet [64] would transfer the universal filtration capabilities to a smaller dataset, providing better results compared to the usual training approach. Even though most prior work used 3D simulations to find suitable grasp poses for objects, Kumra et al. [10] stated that, despite those previous works having performed well, they required a known 3D model of the object to calculate a grasp. This 3D model would not be known a priori and the complex modelling techniques of forming the 3D model was beyond the capacity for most of the general purpose robots as their desired primary function was faster adaptation to dynamic scenarios [19]. In such cases, a learning algorithm would produce the necessary results provided that there were enough, domain-specific data instances for the training.

In conclusion, the number of training data plays a key role in the outcome of the trained algorithm. Some approaches (e.g., [37,61,65]) try to reduce or completely avoid the challenges of compiling such huge datasets. In cases where an extended information set is necessary to produce a grasp prediction, a dataset such as the Dex-Net 2.0 [27] could be used. For most generalised grasp prediction networks, however, the CGD [60] would be an optimal starting point considering its adaptability for more generalised object shapes and poses with the added benefit of the inclusion of depth information.

5. Convolutional Neural Networks for Grasp Detection

Most recent work in robotic grasp detection apply different variations of convolutional neural networks to learn the optimal end-effector configuration for different object shapes and poses [10,18–20,27,37]. They do so by ranking multiple grasp configurations predicted for each object image instance. Ranking is done based on the learned parameters from the representation learning capability of deep learning. As opposed to the manual feature design and extraction steps of classical learning approaches, deep learning can automatically learn how to identify and extract different application specific feature sets [17]. The authors of [17,55] explained the importance of the CNN architecture towards learning. It was further reported that networks with greater depth would be able to learn more complex hierarchical representations from images [66].

In analytical approaches, various grasping application specific parameters such as closure properties and force analysis are combined to successfully model the grasps [9]. Closure properties describe the force and momentum exerted at the point of contact, also known as a *Grasp Wrench*. Depending on the level of friction at each of these points, the point of contact could be further elaborated. According to Bicchi et al. [9], force analysis describes the required grasping force that should be applied by the robotic gripper on the object to grasp it securely without slipping or causing damage. Kinematic modelling between the contact points is a function that describes the relative motion between two different contact points. Reviews (e.g., [9]) suggest how practically impossible it would be to prepare a generalised grasping model using just analytical data. Given how well certain learning algorithms [10,18–20] performed in the past, however, it could be concluded that using visual representation of successful grasps as training data with these learning algorithms would result in usable generalised solutions.

5.1. Architecture

A deep CNN is built with multiple layers to extract information representations [67]. Goodfellow et al. [17] has stated that the representation of the learning process of a deep neural network has similar attributes to the method through which information is processed by the human brain. During the last five years, there have been many active improvements for DCNN architectures. Most of these approaches use ImageNet [64] tests for benchmarking. From inspecting many CNN methods that are originally evaluated on ImageNet data, it is evident that all of them have followed the general structure shown in Figure 5. Literature suggests that lower level features are identified using the convolutional layer while application specific features are extracted by the fully connected portion of the network where pooling and activations are widely employed [68]. This suggests that the results on the ImageNet data provide a reasonably useful evaluation of the architecture even though it is not specific to robotic grasping.

Conv. Layers
FullyConnected

Figure 5. General structure of a CNN.

The literature survey suggests two types of convolutional layer placements in DCNNs. Early approaches used a stacked architecture where each layer was placed one after the other. More recent DCNNs have used convolutional layers in parallel. Szegedy et al. [68] reported that this trend was accelerated due to the availability of increased computational capacities.

Both AlexNet and VGG-Net are stacked deep neural network architectures. With AlexNet, Krizhevsky et al. [66] produced a reduced error rate of 16.6%. Simonyan et al. further reduced the error rates to 7.0% with their introduction of VGG-Net [69]. Redmon et al. [19] were the first authors to implement AlexNet with their work in robotic grasp detection. They fine-tuned the DCNN architecture to accommodate their hardware. Their model is shown in Figure 6. Their direct regression model that was trained on RG-D images achieved an accuracy of 84.4% and the MultiGrasp model that divided an original image to $N \times N$ sub-images achieved an accuracy of 88%. Their work was later followed by Watson et al. [31] who achieved an accuracy of 78% with one fold cross validation. Ebert et al. [20] achieved an accuracy of 71% closely following the same work in [19].

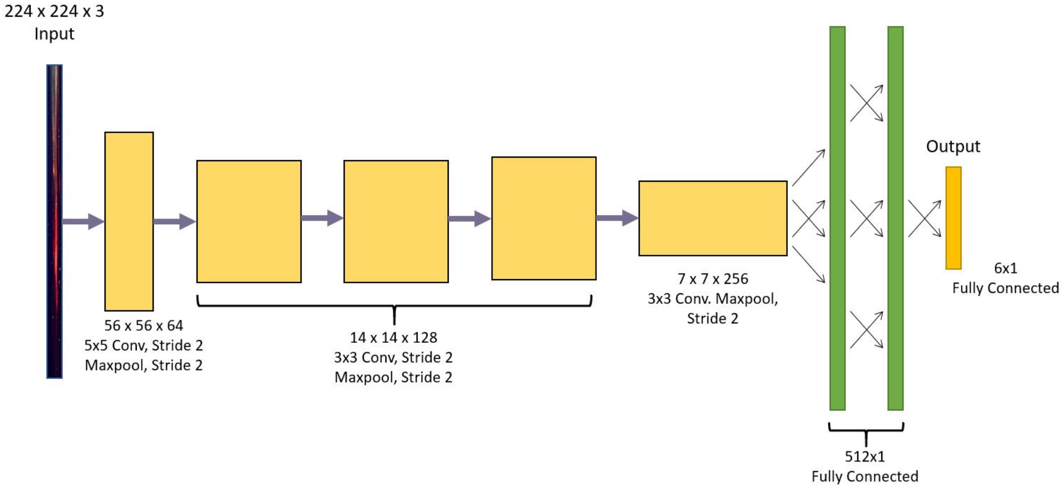


Figure 6. Neural network model proposed by Redmon et al. [19].

Modern dense DCNN architectures are developed under the premise that deeper networks are capable of extracting more advanced features from data. Szegedy et al. [68] reported that the drawbacks of increasing the depth of a DCNN are two-fold. In order to train such deep network models there should be a distinguishable variation between the training data and this was challenging even with human labelling. When the depth of a DCNN is increased, the number of trainable parameters automatically increases, which requires higher computational power for training [68]. Therefore they suggested sparsely connected deep network architectures. They proposed their DCNN architecture known as GoogleNet, with a reduced error rate of 6.8% in ImageNet testing [64]. Following that, He et al. [70] proposed a DCNN architecture with skip connections that further reduced the error rates to 3.57% using their ResNet architecture. A sample residual block is shown in Figure 7. By combining [70] with their original approach in [71], Szegedy et al. proposed the Inception-ResNet architecture [72].

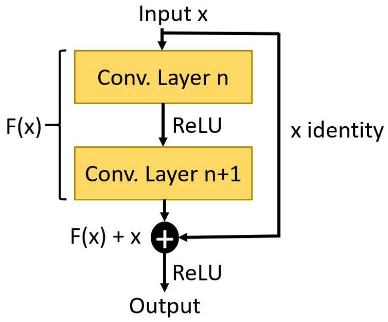


Figure 7. Example of a residual block. These skip connections perform identity mapping optimally and they add neither extra parameter nor computational complexity.

Denser deep networks have appeared in recent robotic grasp detection work. Kumra et al. [10] used the 50-layer version of the popular ResNet [70] architecture to extract features from RGB-D images in order to detect grasp configurations for the objects in the images from the Cornell grasp dataset [60]. The authors presented two different network models that were aimed at grasp detection using uni-modal and multi-modal images named *Uni-modal* and *Multi-modal* architectures, respectively. They achieved the highest grasp detection accuracy of 89.21% using their multi-modal DCNN architecture with image-wise splitting. This architecture is shown in Figure 8. Zhou et al. [34] used the ResNet-50 and ResNet-101 networks as feature extractors in their grasp detection work and achieved accuracies over 98% for both versions. Chu et al. [50] used the same ResNet-50 architecture [70] with their grasp detection work. In contrast to the previous approach [10], they used grasp labels from the grasp data [60] to propose regions of interest in the images to ultimately propose multiple grasps at once. They achieved a detection accuracy of 96% using image-wise splitting.

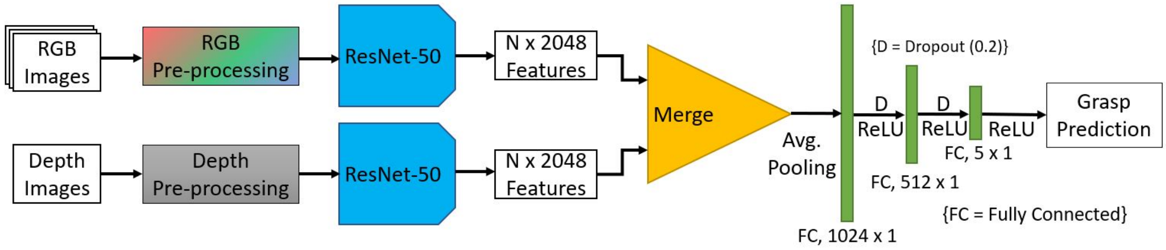


Figure 8. Multi-modal grasp predictor [10].

Another research direction in robotic grasp detection is the use of custom neural network models inspired from the popular region proposal convolutional networks known as R-CNN [49].

Lenz et al. [18] developed their two cascaded CNN models for grasp detection using a sliding window approach. The first neural network model extracted higher level features such grasp locations whereas the larger second network verified the valid grasps from those detected. In the first stage, the authors used a variant of the Sparse Auto Encoder [73] to initialise the weights of the hidden layers. Pre-training in this way was a necessary step to avoid overfitting. The authors reported a grasp detection accuracy of 75% that was tested with a Baxter robot by carrying out object grasping. The creation of application specific DCNNs has received greater attention recently. When there has been enough higher quality training data available, researchers used their own custom neural network models. Most of these works were motivated from recent DCNN success stories in ImageNet classification tests [64]. Lu et al. [52] used a custom architecture for their work in multi-fingered grasp prediction. The max-pooling and rectified linear units were used as activation functions in their work. They further concluded the adaptability of their work into the realm of two-fingered grasp detection. Detection accuracies of 75.6% for novel objects and 76.6% for previously seen objects were claimed.

The literature survey has further suggested that researchers have used DCNN or CNN methods to find the inverse kinematic or dynamic solutions. The kinematic and dynamic modelling of robotic manipulation are the focus of Xia et al. [74] and Polydoros et al. [75], respectively. The proposed methods were further experimented with by various authors in motion planning in unstructured environments [48,76]. Even though these works do not directly align with robotic grasp detection, they can be extended to grasp planning.

Szegedy et al. in [68] stated that advances in the quality of image recognition had relied on newer ideas, algorithms, and improved network architectures as well as more powerful hardware, larger training datasets, and bigger learning models. They further commented that neither the deep networks nor the bigger models alone would result in such improvements but combining them n to create a deeper architecture would suggest these improvements over the classical theories. They experimentally presented that by increasing the depth (the number of deep levels) and the width (the number of units at each level) of a deep network would improve the overall network performance. However, it would also require a commensurately larger set of training data that would result in the following drawbacks:

1. Larger datasets would result in increased features to be extracted while limited datasets would result in overfitting.
2. Deeper networks would require increased computational resources during the training.

The authors evaluated their original DCNN performance in [68] that had an error rate of 6.67% with their own improved version in [71] with an error rate of 3.5%. They concluded in [71] that this performance boost was a result of going deeper with convolutions.

5.2. Transfer Learning Techniques

The most successful robotic grasp detection work has used transfer learning methods to achieve accuracies close to 90%. Any transfer learning approach includes the following steps:

1. Data pre-processing
2. Pre-trained CNN model

Compared to image classification, robotic grasp detection requires the capability of a DCNN to identify grasp configurations for novel objects. This requires training on generalised object scene images. Therefore, most researchers limit their pre-processing techniques to just accommodate CNN input dimensions such as image width and the number of channels. Unlike in image classification, the ground truth data for grasping were less augmented. Redmon et al. [19] reported the minimum amount of necessary pre-processing for RGB-D datasets as centre cropping of images and replacing the blue channel of RGB images with depth data while normalising the depth data to [0, 255] range, which is the default RGB colour space range. While following the exact same procedure in [19], Watson et al. [31] normalised all RGB values and grasp labels to [0, 1] arguing that training targets should be in the same range as the training data. In their method, Pinto et al. [37] resized images to 227 × 227 which was the input size for their model. Their network had a similar architecture to AlexNet [66] as shown in Figure 9.

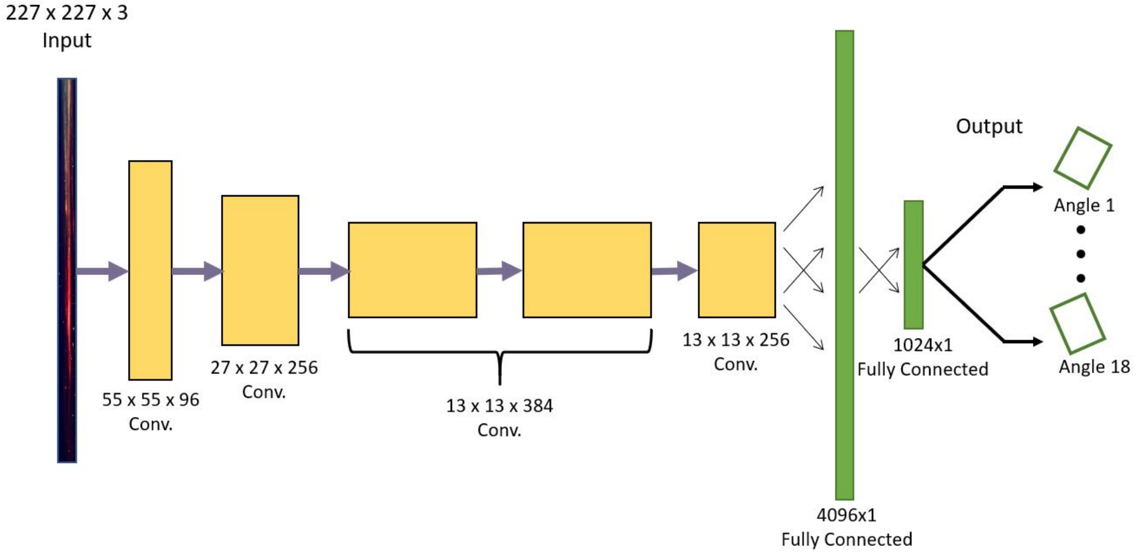


Figure 9. Eighteen-way binary classifier by Pinto et al. [37].

Due to the problem of overfitting with the limited available datasets, the deep learning robotic grasp detection literature indicates that many authors have used pre-training. Pre-training was identified as a transfer learning technique where the deep network was pre-trained on larger datasets prior to the training on domain specific data. The weights of the convolutional layers that were originally learned during the pre-training were kept frozen during the training on domain specific data.

In their one-shot detection method, Redmon et al. [19] used the AlexNet [66] convolutional network architecture in compiling their network model [19] which achieved a grasp detection accuracy of 84.4%. The same approach was used in [20,31] with similar results reported. They modified the orientation parameter from the grasp representation while arguing that the angle predictions are two-fold (positive or negative) [19]. Therefore, the authors replaced θ with $(\sin 2\theta, \cos 2\theta)$ following the trigonometric definitions. The argument was further supported in [20,31].

A similar pre-training approach was employed in [10] using the deeper ResNet-50 architecture as opposed to the variant of the AlexNet from [19]. They reported that the deeper network model afforded them an increased accuracy of 89.21%. They further reported a second model as shown in Figure 10. It was aimed at uni-modal data such as RGB or RG-D images. This model achieved an accuracy of 88.84%.

Even though the transfer learning made it less challenging to use a pre-trained model for the convolutional part of a DCNN, researchers still had to design the fully connected part of the DCNN. While there is not enough evidence to determine the optimal number of units required, Redmon et al. [19] used two fully connected layer that had 512 units in each individual layer. In [10], Kumra et al. used one fully connected layer with 512 units for their Uni-modal architecture and two layers with 512 units for their Multi-modal architecture. A model similar to the deep network in [19] was later followed in [20,31]. In DCNN training applications, the popular learning optimiser is the Stochastic Gradient Decent (SGD). In deep grasp detection, most authors used the SGD but they argued that it was not an optimal optimiser and reported more advanced optimisers were necessary. Ruder provided a comprehensive overview of different learning optimisers in [77].

While most transfer learning approaches employed pre-trained network models in an end-to-end learning process some researchers used them as feature extractors for shallow network models.

Chollet [78] stated that, due to the two-step method of feature extraction, it was impossible to employ data augmentation techniques if required as the learned features would not be the same as the training images. In addition, running end-to-end learning was costlier as it required the convolutional base of the network to be run on data repetitively.

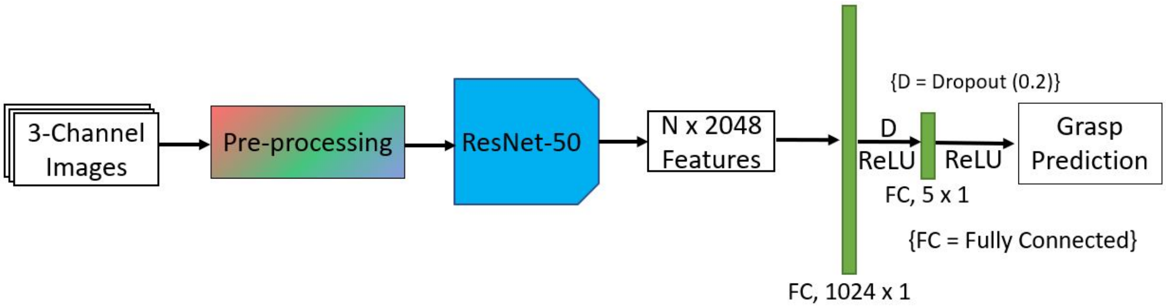


Figure 10. Uni-modal grasp predictor [10].

Detecting object grasping configurations from images is still accurately solved using analytical methods but the use of empirical methods is exponentially increasing due to successful results in recent publications. One commonly used method is to train a visuomotor controller using deep learning that iteratively corrects the grasping point until the object is successfully grasped between the gripper jaws. The next best method is to learn a function that scores the possible grasps on an image and use it to select the highest scored grasp as the candidate. There are other methods that learn a certain heuristic and exhaustively search for possible grasps on the images. Training CNNs to detect grasps requires a high volume of manually labelled data. As a solution, most researchers opt to use simulated training data (e.g., [14,38,53]). Alternatively, data collection can be automated (e.g., [37]). Recent approaches have suggested network pre-training can help to avoid overfitting due to the limited availability of training data [10,19,20]. As shown in Table 2, AlexNet [66] and ResNet-50 [70] have been widely used in these studies. The reasoning for the reduced results in [20] and [31] compared to [19] was not clarified by any author. However, the literature survey suggested that these methods utilised a lower number of training data instances and a varied set of data augmentation techniques as compared to [19]. This, further supported the arguments made by many different researchers regarding the limited availability of high quality annotated training data for such robotic grasp detection work.

Table 2. Comparison between different transfer learning techniques in one-shot grasp detection. Results were reported from tests performed on the Cornell Grasp Dataset.

Method	Architecture	Accuracy (%) (Image-Wise)	Accuracy (%) (Object-Wise)
Direct regression by Redmon et al. [19]	AlexNet [66]	84.4%	84.9%
Regression + Classification by Redmon et al. [19]	AlexNet [66]	85.5%	84.9%
MultiGrasp Detection by Redmon et al. [19]	AlexNet [66]	88.0%	87.1%
Uni-modal, SVM, RGB by Kumra et al. [10]	ResNet-50 [70]	84.76%	84.47%
Uni-modal RGB by Kumra et al. [10]	ResNet-50 [70]	88.84%	87.72%
Uni-modal RG-D by Kumra et al. [10]	ResNet-50 [70]	88.53%	88.40%
Multi-modal SVM, RGB-D by Kumra et al. [10]	ResNet-50 [70]	86.44%	84.47%
Multi-modal RGB-D by Kumra et al. [10]	ResNet-50 [70]	89.21%	88.96%
Direct Regression (RG-D) by Basalla et al. [20]	AlexNet [66]	71%	NA
Direct Regression (RG-D) By Watson et al. [31]	AlexNet [66]	78%	NA

5.3. Evaluation of Results

Each grasp configuration that was predicted by the learning algorithms should go through an evaluation process to identify if it is in fact a valid grasp. There are two evaluation metrics for grasps:

rectangle metric and **point metric** [10,19,20]. The point metric evaluates the distance between the predicted grasp centre and the actual grasp centre relative to a threshold value, but the literature does not provide further insight as to how best to determine this threshold value. Furthermore, this metric does not consider the grasp angle, which neglects the object orientation in the 2D image plane.

The rectangle metric defined a successful grasp under the following two conditions:

1. Difference between the grasp angles to be less than 30°
2. Jacquard index between the two grasps to be less than 25%

The Jacquard index between $Grasp_{pred}$ and $Grasp_{true}$ is given by:

$$J(Grasp_{pred}, Grasp_{true}) = \frac{|Grasp_{pred} \cap Grasp_{true}|}{|Grasp_{pred} \cup Grasp_{true}|} \quad (1)$$

This method evaluates grasp poses in their image plane. Watson et al. [31] argued that further evaluation metrics would be necessary in order to evaluate grasps on the fly. They proposed an on-line evaluation method known as "marker based evaluation method". The process evaluates grasps detected on images which had objects with the labels marked by the authors. In Figure 11, a marked banana is shown with red markers. The authors evaluated the distance to the predicted grasp centres from these markers and reported the grasp success based on this [31].

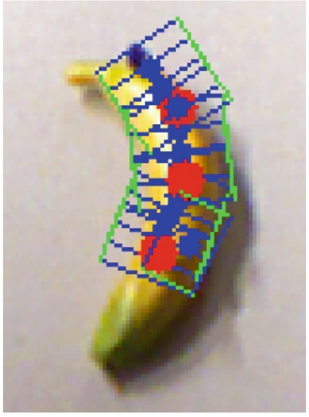


Figure 11. Marker based evaluation method [31].

Furthermore, the authors validated their results with one-fold cross validation and implemented an inverse kinematic grasp planner with a Baxter robot. An implementation of grasp planning supported their conclusions in [31] as opposed to previous attempts in [10,19] that failed to further validate grasp detection results on a real robot.

In most robotic grasp detection studies, researchers proposed the grasp detection as a computer vision problem and presented their results as detection accuracy, mean error of orientation, and the mean jacquard index for predictions. The only conclusion that can be drawn from these results is an evaluation metric for the grasp detection. There are very few approaches that have carried out grasping with real robotic hardware. Although the evaluation metrics draw an acceptable conclusion for the grasp detection, the additional results from actual robotic grasping will further confirm the results from grasp detection.

6. Conclusions

Deep learning has been showing remarkable success in the applications of computer vision such as classification, detection and localisation. Thus far, however, it has not been adopted very extensively in robotic applications, although this is now a rapidly growing area of research. Due to the requirements of higher computational power and large volumes of training data, it is still challenging to implement an end-to-end learning approach for the complete robotic grasping activity. Despite this, the detection of successful grasping poses for robotic systems using deep learning methods has been investigated in a number of recent publications. In this paper, these emerging methods have been reviewed and several key elements of deep learning based grasp detection have been identified as: grasp representation, grasp detection, training, and CNN architectures.

Several methods to represent successful grasps for a robotic system have been discussed. A successful grasp is identified as the location within the work area that a robotic end-effector can be placed to securely grasp the target object between its parallel plate grippers and lift it without losing its grip. The information such as the coordinates of this location, the width the gripper needs to be opened, and the gripper orientation with reference to the horizontal axis are commonly included in this grasp representation.

We have identified that the most popular deep learning approach to robotic grasp detection was the sliding window approach in which regions of interest on images are scanned for successful grasps in an iterative process. A major limitation of this approach, however, is that it is a comparatively slow process and unsuitable for real-time operations. This method has been superseded by the current state-of-the-art one-shot detection method in which a structured grasp configuration is directly predicted for an image. Recently, Redmon et al. [19] introduced the one-shot detection method that resulted in a detection accuracy of 88% and a detection speed of 76 milliseconds for an image. Kumra et al. [10] surpassed these results by employing a deeper network model and achieved a detection accuracy of 89.21%.

Training a suitable neural network for grasp detection purposes usually requires a large amount of manually labelled data, such as the Cornell Grasps Dataset, but this requires even higher volumes of domain specific data which is not readily available at the time of writing. Therefore researchers have opted to collect data with self-supervised methods. While most of these applications rely on realistic data such as images from objects in the real world, more recent literature discusses the use of simulation data and their domain adaptation capability. The one-shot detection algorithm follows the transfer learning technique to employ a pre-trained DCNN as its convolutional base. The current highest achieving DCNN architecture for grasp detection is based on the ResNet-50 [70] where an accuracy of 96% is reported for successful grasp detection work.

Future Work

A significant factor in the use of supervised learning algorithms is the general requirement for large datasets for better results. The prior work reviewed in this paper has highlighted that this requirement is a major constraint for many grasp detection approaches that utilise supervised learning methods. Most of the grasp data that are labelled by human annotators are biased on the semantics and there can be almost unlimited ways to define a good grasp, which can make useful annotation practically impossible. Due to the inherent difficulties in acquiring suitable datasets, novel methods to overcome these limitations are needed. Larger datasets can potentially be generated from simulated data through application of domain adaptation techniques to generate application-specific datasets from 3D simulations. In addition, very few researchers have explored methods of collecting data through the incorporation of reinforcement learning techniques to autonomously create larger datasets. Reinforcement learning is a reward based approach that involves a lengthier training process, similar to the trial-and-error process, but does not require a pre-labelled dataset. A major challenge in using reinforcement learning for data collection is the time required to obtain the necessary number of valid data instances compared to the numerous invalid data instances that are continuously generated due to its randomised nature. Further work is needed to identify more time efficient methods.

In current grasp detection systems, a separate grasp planning system is required to implement the robotic grasping function on robotic hardware. More recent literature discusses the possibility of detecting visuomotor control policies directly from images to avoid the need to have a separate grasp planning system. Recently, researchers have begun to explore the application of reinforcement learning techniques to predict visuomotor control policies for robotic grasping. Limited work has been carried out so far as researchers have been reluctant to use such methods with real-world systems considering the number of precautionary steps required to mitigate potential damage to the robotic equipment, as well as the longer training times required. However, with the recent introduction of domain adaptation techniques, there are now more suitable methods to train reinforcement learning algorithms in simulated environments. Given this, it is very worthwhile to further explore such methods in robotic grasp detection as well as their potential extension to learning direct visuomotor control policies.

Author Contributions: S.C. is the first author and contributed more than 75% of content and material for the article. A.R. is the principal supervisor of S.C. and is the second author. D.C. is the collaborative supervisor of S.C. and the third author of the article. A.R. and D.C. both contributed substantively for content refining and editing.

Funding: S.C. is a recipient of the ECU School of Engineering Postgraduate Research Scholarship.

Conflicts of Interest: The authors declare no conflict of interest as the work was not sponsored by any government, organisation, or individual.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
APC	Amazon Picking Challenge
CGD	Cornell Grasp Dataset
CNN	Convolutional Neural Networks
DCNN	Deep Convolutional Neural Networks
DDD	Depth in 3 Channels (Depth Depth Depth) Image
Dex-Net	Dexterity Network
DOF	Degrees of Freedom
GQ-CNN	Grasp Quality Convolutional Neural Networks
GWS	Grasp Wrench Space
MRI	Magnetic Resonance Imaging
RGB	Red Green Blue Image
RGB-D	Red Green Blue Depth Image
RG-D	Red Green Depth Image
SGD	Stochastic gradient decent

References

1. Lopes, M.; Santos-Victor, J. Visual Learning by Imitation with Motor Representations. *IEEE Trans. Syst. Man Cybern. B* **2005**, *35*, 438–449. [CrossRef]
2. Ju, Z.; Yang, C.; Li, Z.; Cheng, L.; Ma, H. Teleoperation of Humanoid Baxter Robot Using Haptic Feedback. In Proceedings of the 2014 Multisensor Fusion and Information Integration for Intelligent Systems (MFI), Beijing, China, 28–29 September 2014.
3. Konidaris, G.; Kuindersma, S.; Grupen, R.; Barto, A. Robot learning from demonstration by constructing skill trees. *Int. J. Robot. Res.* **2011**, *31*, 360–375. [CrossRef]
4. Kober, J.; Peters, J. Imitation and reinforcement learning. *IEEE Robot. Autom. Mag.* **2010**, *17*, 55–62. [CrossRef]
5. Peters, J.; Lee, D.D.; Kober, J.; Nguyen-Tuong, D.; Bagnell, A.; Schaal, S. Robot Learning. In *Springer Handbook of Robotics*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2017; pp. 357–394.
6. Mathworks. Deep Learning in Matlab. Available online: https://au.mathworks.com/help/nnet/ug/deep-learning-in-matlab.html (accessed on 28 April 2017).
7. Chen, Z.; Zhang, T.; Ouyang, C. End-to-End Airplane Detection Using Transfer Learning in Remote Sensing Images. *Remote Sens.* **2018**, *10*, 139. [CrossRef]
8. Rosenberg, I.; Sicard, G.; David, E.O. End-to-End Deep Neural Networks and Transfer Learning for Automatic Analysis of Nation-State Malware. *Entropy* **2018**, *20*, 390. [CrossRef]

9. Bicchi, A.; Kumar, V. Robotic grasping and contact: A review. In Proceedings of the 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), San Francisco, CA, USA, 24–28 April 2000; Volume 1, pp. 348–353. [CrossRef]
10. Kumra, S.; Kanan, C. Robotic grasp detection using deep convolutional neural networks. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 769–776. [CrossRef]
11. Ju, Z.; Yang, C.; Ma, H. Kinematics Modeling and Experimental Verification of Baxter Robot. In Proceedings of the Chinese Control Conference (CCC), Nanjing, China, 28–30 July 2014.
12. Billard, A.G.; Calinon, S.; Dillmann, R. Learning from Humans. In *Springer Handbook of Robotics*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2017; pp. 1995–2014.
13. Jeon, M. Robotic Arts: Current Practices, Potentials, and Implications. *Multimodal Technol. Interact.* **2017**, *1*, 5. [CrossRef]
14. Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; Abbeel, P. Domain randomization for transferring deep neural networks from simulation to the real world. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 23–30. [CrossRef]
15. Saxena, A.; Driemeyer, J.; Kearns, J.; Ng, A.Y. Robotic Grasping of Novel Objects. In *Advances in Neural Information Processing Systems 19*; Schölkopf, B., Platt, J.C., Hoffman, T., Eds.; MIT Press: Cambridge, MA, USA, 2007; pp. 1209–1216.
16. Saxena, A.; Driemeyer, J.; Ng, A.Y. Robotic Grasping of Novel Objects using Vision. *Int. J. Robot. Res.* **2008**, *27*, 157–173. [CrossRef]
17. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
18. Lenz, I.; Lee, H.; Saxena, A. Deep Learning for Detecting Robotic Grasps. *Int. J. Robot. Res.* **2015**, *34*, 705–724. [CrossRef]
19. Redmon, J.; Angelova, A. Real-time grasp detection using convolutional neural networks. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 1316–1322. [CrossRef]
20. Basalla, M.; Ebert, F.; Tebner, R.; Ke, W. Grasping for the Real World (Greifen mit Deep Learning). Available online: https://www.frederikebert.de/abgeschlossene-projekte/greifen-mit-deep-learning/ (accessed on 13 February 2017).
21. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. Available online: tensorflow.org (accessed on 23 February 2017).
22. Bergstra, J.; Breuleux, O.; Bastien, F.; Lamblin, P.; Pascanu, R.; Desjardins, G.; Turian, J.; Warde-Farley, D.; Bengio, Y. Theano: A CPU and GPU Math Expression Compiler. In Proceedings of the Python for Scientific Computing Conference (SciPy), Austin, TX, USA, 28 June–3 July 2010.
23. Chollet, F. Keras. 2015. Available online: https://keras.io (accessed on 8 November 2017).
24. Jia, Y.; Shelhamer, E.; Donahue, J.; Karayev, S.; Long, J.; Girshick, R.; Guadarrama, S.; Darrell, T. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv* **2014**, arxiv:1408.5093.
25. Redmon, J. Darknet: Open Source Neural Networks in C. 2013–2016. Available online: http://pjreddie.com/darknet/ (accessed on 12 April 2017).
26. NVIDIA. NVIDIA ISAAC Platform for Robotics. Available online: https://www.nvidia.com/en-us/deep-learning-ai/industries/robotics/ (accessed on 23 December 2017).
27. Mahler, J.; Liang, J.; Niyaz, S.; Laskey, M.; Doan, R.; Liu, X.; Aparicio, J.; Goldberg, K. Dex-Net 2.0: Deep Learning to Plan Robust Grasps with Synthetic Point Clouds and Analytic Grasp Metrics. In Proceedings of the Robotics: Science and Systems, Cambridge, MA, USA, 12–16 July 2017. [CrossRef]
28. Bohg, J.; Morales, A.; Asfour, T.; Kragic, D. Data-Driven Grasp Synthesis-A Survey. *IEEE Trans. Robot.* **2014**, *30*, 289–309. [CrossRef]
29. Zhang, F.; Leitner, J.; Milford, M.; Upcroft, B.; Corke, P. Towards Vision-Based Deep Reinforcement Learning for Robotic Motion Control. In Proceedings of the Australasian Conference on Robotics and Automation, Canberra, Australia, 2–4 December 2015.

30. Jiang, Y.; Moseson, S.; Saxena, A. Efficient grasping from RGBD images: Learning using a new rectangle representation. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 3304–3311. [CrossRef]
31. Watson, J.; Hughes, J.; Iida, F. Real-World, Real-Time Robotic Grasping with Convolutional Neural Networks. In *Towards Autonomous Robotic Systems*; Gao, Y., Fallah, S., Jin, Y., Lekakou, C., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 617–626.
32. Trottier, L.; Giguère, P.; Chaib-draa, B. Sparse Dictionary Learning for Identifying Grasp Locations. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; pp. 871–879. [CrossRef]
33. Park, D.; Chun, S.Y. Classification based Grasp Detection using Spatial Transformer Network. *arXiv* **2018**, arxiv:1803.01356.
34. Zhou, X.; Lan, X.; Zhang, H.; Tian, Z.; Zhang, Y.; Zheng, N. Fully Convolutional Grasp Detection Network with Oriented Anchor Box. *arXiv* **2018**, arxiv:1803.02209. [CrossRef]
35. Guo, D.; Sun, F.; Liu, H.; Kong, T.; Fang, B.; Xi, N. A hybrid deep architecture for robotic grasp detection. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 1609–1614. [CrossRef]
36. Wang, Z.; Li, Z.; Wang, B.; Liu, H. Robot grasp detection using multimodal deep convolutional neural networks. *Adv. Mech. Eng.* **2016**, *8*, 1687814016668077. [CrossRef]
37. Pinto, L.; Gupta, A. Supersizing self-supervision: Learning to grasp from 50K tries and 700 robot hours. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 3406–3413. [CrossRef]
38. Viereck, U.; Pas, A.; Saenko, K.; Platt, R. Learning a visuomotor controller for real world robotic grasping using simulated depth images. In Proceedings of the 1st Annual Conference on Robot Learning, Mountain View, California, USA, 13–15 November 2017; Volume 78, pp. 291–300.
39. Mahler, J.; Goldberg, K.Y. Learning Deep Policies for Robot Bin Picking by Simulating Robust Grasping Sequences. In Proceedings of the 1st Annual Conference on Robot Learning, Mountain View, CA, USA, 13–15 November 2017.
40. Calandra, R.; Owens, A.; Upadhyaya, M.; Yuan, W.; Lin, J.; Adelson, E.H.; Levine, S. The Feeling of Success: Does Touch Sensing Help Predict Grasp Outcomes? *arXiv* **2017**, arxiv:1710.05512.
41. Calandra, R.; Owens, A.; Jayaraman, D.; Lin, J.; Yuan, W.; Malik, J.; Adelson, E.H.; Levine, S. More Than a Feeling: Learning to Grasp and Regrasp using Vision and Touch. *arXiv* **2018**, arxiv:1805.11085.
42. Murali, A.; Li, Y.; Gandhi, D.; Gupta, A. Learning to Grasp Without Seeing. *arXiv* **2018**, arxiv:1805.04201.
43. Ku, L.Y.; Learned-Miller, E.; Grupen, R. Associating grasp configurations with hierarchical features in convolutional neural networks. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 2434–2441. [CrossRef]
44. Amazon Robotics. Amazon Picking Challenge 2016. Available online: https://www.amazonrobotics.com/#/pickingchallenge (accessed on 6 July 2017).
45. Schwarz, M.; Milan, A.; Lenz, C.; Muńoz, A.; Periyasamy, A.S.; Schreiber, M.; Schüller, S.; Behnke, S. Nimbro Picking: Versatile Part Handling for Warehouse Automation. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017.
46. Johnson, J.; Karpathy, A.; Fei-Fei, L. DenseCap: Fully Convolutional Localization Networks for Dense Captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016.
47. Wei, J.; Liu, H.; Yan, G.; Sun, F. Robotic grasping recognition using multi-modal deep extreme learning machine. *Multidimens. Syst. Signal Process.* **2017**, *28*, 817–833. [CrossRef]
48. Wang, J.; Hu, Q.; Jiang, D. A Lagrangian network for kinematic control of redundant robot manipulators. *IEEE Trans. Neural Netw.* **1999**, *10*, 1123–1132. [CrossRef] [PubMed]
49. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; IEEE Computer Society: Washington, DC, USA, 2014; pp. 580–587. [CrossRef]
50. Chu, F.J.; Xu, R.; Vela, P. Real-world Multi-object, Multi-grasp Detection. *IEEE Robot. Autom. Lett.* **2018**, *3*, 3355–3362. [CrossRef]

51. ten Pas, A.; Gualtieri, M.; Saenko, K.; Platt, R. Grasp Pose Detection in Point Clouds. *Int. J. Robot. Res.* **2017**, *36*, 1455–1473. [CrossRef]
52. Lu, Q.; Chenna, K.; Sundaralingam, B.; Hermans, T. Planning Multi-Fingered Grasps as Probabilistic Inference in a Learned Deep Network. *arXiv* **2017**, arxiv:1804.03289.
53. Johns, E.; Leutenegger, S.; Davison, A.J. Deep learning a grasp function for grasping under gripper pose uncertainty. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 4461–4468. [CrossRef]
54. Zeng, A.; Song, S.; Yu, K.; Donlon, E.; Hogan, F.R.; Bauzá, M.; Ma, D.; Taylor, O.; Liu, M.; Romo, E.; et al. Robotic Pick-and-Place of Novel Objects in Clutter with Multi-Affordance Grasping and Cross-Domain Image Matching. In Proceedings of the IEEE International Conference on Robots and Automation (ICRA), Brisbane, Australia, 21–26 May 2018.
55. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
56. Brownlee, J. *Deep Learning with Python: Develop Deep Learning Models on Theano and TensorFlow Using Keras*; Machine Learning Mastery: Melbourne, Australia, 2017.
57. Ruiz-del-Solar, J.; Loncomilla, P.; Soto, N. A Survey on Deep Learning Methods for Robot Vision. *arXiv* **2018**, arxiv:1803.10862.
58. Lai, K.; Bo, L.; Ren, X.; Fox, D. A large-scale hierarchical multi-view RGB-D object dataset. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 817–1824. [CrossRef]
59. Song, X.; Jiang, S.; Herranz, L. Combining Models from Multiple Sources for RGB-D Scene Recognition. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, Melbourne, Australia, 19–25 August 2017; pp. 4523–4529. [CrossRef]
60. Cornell University. Robot Learning Lab: Learning to Grasp. Available online: http://pr.cs.cornell.edu/grasping/rect_data/data.php (accessed on 12 April 2017).
61. Levine, S.; Pastor, P.; Krizhevsky, A.; Ibarz, J.; Quillen, D. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *Int. J. Robot. Res.* **2018**, *37*, 421–436. [CrossRef]
62. Bousmalis, K.; Irpan, A.; Wohlhart, P.; Bai, Y.; Kelcey, M.; Kalakrishnan, M.; Downs, L.; Ibarz, J.; Sampedro, P.P.; Konolige, K.; et al. Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping. *arXiv* **2017**, arxiv:1709.07857.
63. Mahler, J. Releasing the Dexterity Network (Dex-Net) 2.0 Dataset for Deep Grasping. Available online: http://bair.berkeley.edu/blog/2017/06/27/dexnet-2.0/. (accessed on 6 July 2017).
64. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
65. Gandhi, D.; Pinto, L.; Gupta, A. Learning to Fly by Crashing. *arXiv* **2017**, arxiv:1704.05588.
66. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*; Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q., Eds.; Curran Associates, Inc.: New York, NY, USA, 2012; pp. 1097–1105.
67. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
68. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015, pp. 1–9. [CrossRef]
69. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arxiv:1409.1556.
70. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
71. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826. [CrossRef]

72. Szegedy, C.; Ioffe, S.; Vanhoucke, V. *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*; Google Inc.: Mountain View, CA, USA, 2016.
73. Goodfellow, I.; Lee, H.; Le, Q.V.; Saxe, A.; Ng, A.Y. Measuring Invariances in Deep Networks. In *Advances in Neural Information Processing Systems 22*; Bengio, Y., Schuurmans, D., Lafferty, J.D., Williams, C.K.I., Culotta, A., Eds.; Curran Associates, Inc.: New York, NY, USA, 2009; pp. 646–654.
74. Xia, Y.; Wang, J. A dual neural network for kinematic control of redundant robot manipulators. *IEEE Trans. Syst. Man Cybern. B* **2001**, *31*, 147–154. [CrossRef]
75. Polydoros, A.S.; Nalpantidis, L.; Krüger, V. Real-time deep learning of robotic manipulator inverse dynamics. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015.
76. Ding, H.; Wang, J. Recurrent neural networks for minimum infinity-norm kinematic control of redundant manipulators. *IEEE Trans. Syst. Man Cybern. A* **1999**, *29*, 269–276. [CrossRef]
77. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arxiv:1609.04747.
78. Chollet, F. *Deep Learning with Python*; Manning Publications Company: Shelter Island, NY, USA, 2017.