# Deep-learning in Mobile Robotics - from Perception to Control Systems: A Survey on Why and Why not

Lei Tai[1], Ming Liu[1,2], *Member, IEEE,*

*Abstract*—Deep-learning has dramatically changed the world overnight. It greatly boosted the development of visual perception, object detection, and speech recognition, etc. That was attributed to the multiple convolutional processing layers for abstraction of learning representations from massive data. The advantages of deep convolutional structures in data processing motivated the applications of artificial intelligence methods in robotic problems, especially perception and control system, the two typical and challenging problems in robotics. This paper presents a survey of the deep-learning research landscape in mobile robotics. We start with introducing the definition and development of deep-learning in related fields, especially the essential distinctions between image processing and robotic tasks. We described and discussed several typical applications and related works in this domain, followed by the benefits from deep-learning, and related existing frameworks. Besides, operation in the complex dynamic environment is regarded as a critical bottleneck for mobile robots, such as that for autonomous driving. We thus further emphasize the recent achievement on how deep-learning contributes to navigation and control systems for mobile robots. At the end, we discuss the open challenges and research frontiers.

*Index Terms*—Deep Learning, Perception, Sensory-motor, Robotics.

## I. INTRODUCTION

**R**OBOTS have been more often in use for both industry and daily life along the rapid advancement in science and technology over the last decades. It is convinced that robotics will be indispensable in the future: warehousing robots, autopilot cars, unmanned aerial vehicles, industrial robots, *etc*. Everyone from this era has direct experience of this inevitable trend. However, there are still some unsolved challenges, among which perception and intelligent control are taking important positions. Deep-learning (*DL*) is the leading breakthrough technique to overcome these challenges. A key representative of the advancement in deep-learning is the application of the GO game. A deep-learning enhanced agent (Silver et al., 2016) beat the best human player in this game which was once to be considered as the most difficult problem for artificial intelligence (Bengio, 2009). In this paper, we try to survey the deep-learning related research, especially

for solving robotic related problems with emphasis to mobile robotic challenges.
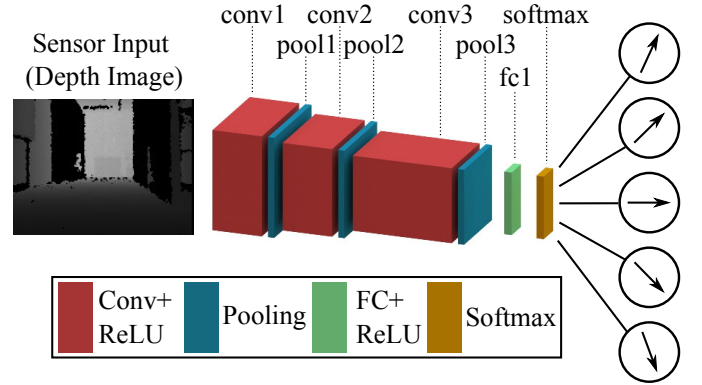


Fig. 1. A typical *CNN* framework for robotics. Raw sensor information (depth image from an RGB-D sensor) was taken as input to connect with the robot control command directly (Tai et al., 2016).

We divide *DL* methods for robotic problems into two categories: perception and control systems. Firstly, the perception of the environment relies on various sensor information. Classical methods extract information from the raw sensor readings based on artificially designed complex features (*e.g.*, *SIFT* (Lowe, 1999, 2004; Liu et al., 2014), *SURF* (Bay et al., 2006, 2008), *ORB* (Rublee et al., 2011; Moreno et al., 2015; Mur-Artal et al., 2015)). Most of the classical methods are constrained by the adaptivity to generic environments. Regarding unstructured dynamic environments, such as hybrid environments for autonomous cars or complicated production lines for industrial robots, those methods are prone to errors. Secondly, the process from sensory information to motion is generally regarded as a sensory-motor control system. A wider scope of such systems includes mobile navigation, planning, grasping and assistant system, etc. Based on environmental perception, appropriate decision-making policy or control strategy should be realized effectively and efficiently. For example, the navigation for mobile robots with rational motion and coarse precision is considered as solved (Cadena et al., 2016). However, in challenging environments or with high expectation on the precision over large-scale, further research for robust methods are still needed. We show some *DL*-based solutions for this problem.

Deep-learning has been widely used to solve artificial intelligent problems (Bengio, 2009). With benefits from the improvement of computational hardware, multiple processing layers, like deep convolution neural network, reached great success in solving complicated estimation problems.
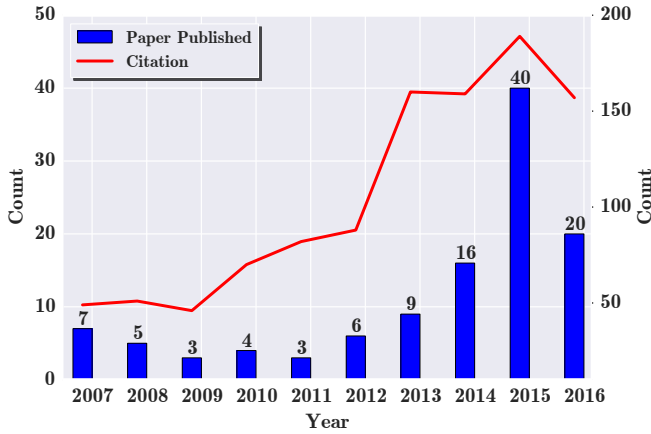
Fig. 2. *Deep-learning Robot* keywords related paper count and citation count in the last decade until 22 Nov 2016. It is obvious that both the count of *Deep-learning Robot* related publications and citations have been increasing dramatically.

The multiple levels of abstraction also indicates the great advantage in high-dimensional data representation and processing (LeCun et al., 2015) compared with other state-of-the-art methods. Convolutional Neural Networks (*CNN*) have won almost all of contests in machine learning and pattern recognition, for example image recognition (He et al., 2016), object detection (Ren et al., 2015), semantics segmentation (Chen et al., 2016a), audio recognition (Deng et al., 2013), and natural language processing (Cambria and White, 2014). Deep-learning is capable of extracting multi-level features from raw sensor information directly without any handcrafted designs. It implies the great potential of deep-learning in future applications. Novel deep-learning frameworks (*e.g., Caffe* (Jia et al., 2014), *Tesnforflow* (Abadi et al., 2016)) and algorithms have been created constantly, which has also accelerated this progress (LeCun et al., 2015).

Since the potential of *DL* has been released and revealed, more researchers have progressively explored other possibilities and applications using deep-learning related methods, especially in robotics. Figure 2 depicts the increasing trend of *Deep-learning Robot* related publications over the past decades. We believe the trend will keep on exploding for the near future.

### A. Scope and paradigms

We give a broad survey of the current development of *DL* in robotics and discuss its open challenges for mobile robotics in perception and control systems. Since both robotics and deep-learning as themselves contain a large number of aspects and wide paradigms. We would like to constrain our scope within the following paradigms.

- *Robot:* This paper mainly talks about mobile robots, unmanned aerial vehicle, manipulators, *etc*. Other forms such as nano-robots, emotional robots, humanoid or designation are not included.
- *Perception:* The raw sensor inputs are constrained within cameras, depth cameras, Lidar, *etc*. The perception here

refers to those as auxiliary for mobile robotic tasks. e.g. speech recognition is also essential for the robot and human interaction, but since it is irrelevant to robotic research thus not included in this survey.

- *Control:* We consider the transformation from sensory input to control output as a sensory-motor control system. We show a lot of robotic sensory-motor problems aided by *DL*, including mobile robots navigation, robot arms grasping, path planning, *etc*, through the direct or indirect connection between the sensory information and the control command. We also survey the deep-learning related Model-Predictive Control research, where the dynamics transition model is learned from deep neural networks. Other control theory-related paradigms such as system stability, controllability and observability are not included.
- *Deep-learning:* Deep hierarchical model and its extensions or enhancement by other learning methods are all considered as deep-learning models in this survey, such that we try to maximize the scope of related methods while focusing on the applications to mobile robots.

Deep-learning provides hands-on and practical solutions to various fields with aid from data. Note that a minority that is beyond but highly related to robotics is also discussed. We believe the cross-fertilization among these fields can be also inspiring to the readers.

The paper starts with presenting a brief introduction to deep-learning both from the machine learning point-of-view and robotics in Section I-B and Section I-C. A large variety of tasks where deep-learning has been successfully applied to mobile robotics is presented in this paper. Section II surveys the application of deep-learning in robotics perception. The perception of robotics is divided into two aspects: the object detection problem; the environment and place recognition problem. Section III deals with deep-learning in robotics control system. The recent improvement in Deep Reinforcement Learning (*DRL*) and Deep Inverse Reinforcement Learning (*DIRL*) are highlighted. Future directions are presented at the end of these sections, followed by conclusion in Section IV.

### B. Deep-learning as machine learning

*DL* is a typical machine learning method. Machine learning is typically classified into three categories: supervised learning, unsupervised learning, and reinforcement learning (Russell et al., 2003; Bishop, 2006).

**Supervised Learning** The most common form of machine learning is supervised learning. In supervised learning, an objective function is computed to measure the error between the output and the desired labeled with a cost function. The final output is the one with the minimum cost.

**Unsupervised Learning** Compared with supervised learning, there is no labeled data provided in unsupervised learning. It aims to find the hidden pattern, structure or features embedded with the data. The primary form of learning for humans and animals is unsupervised learning, i.e. to explore the world not from being told so but from observation.

**Reinforcement Learning** In reinforcement learning, an agent is defined to explore and exploit the space of possible

strategies. Feedback in form of reward or cost from the dynamic environment is referred as the outcome of the chosen action.

All above are inherently tackling features from the training data. Supervised and unsupervised learning intend to locate the nearest neighbors in feature spaces through supervision, and reinforcement learning is a stochastic search problem to identify the highest-reward policy by trial-and-error. Internal adjustable parameters of these models are called weights. For supervised learning and reinforcement learning, weights of the models are motivated to converge by gradient descent methods. Traditionally, machine learning methods are seriously limited to the feature quality and complexity. For example, LeCun et al. (2015) described a selectivity-invariance dilemma for supervised learning: representations abstracted by extractors should be both selective for discrimination and invariant to irrelevant aspects. Generic non-linear features like kernel methods (Smola and Schölkopf, 1998) can boost the performance of the classifiers, but such kind of features always brings over-fitting problems for the training samples. Hand-crafted feature extractor is another useful method conventionally, but complicated domain expertise is essential for such kind of extractors. Deep-learning aims to provide a general-purpose learning procedure so that high-quality features can be learned autonomously and efficiently from data.

*DL* structure always involves a multilayer stack of learning and non-linear modules (layers) (LeCun et al., 2015), which increase both the selectivity and the invariance. We also consider this is the primary contribution and advantage of deep-learning. It also applies to feature representations in unsupervised learning and reinforcement learning.
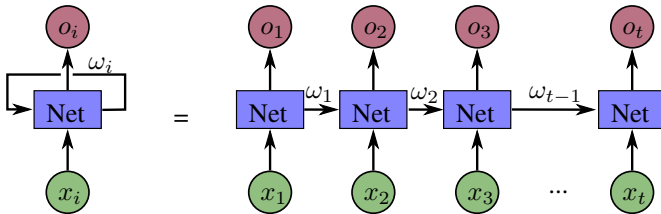


Fig. 3. Recurrent Neural Network Structure. The left is the typical *RNN* structure. The right part is the unfolding version where the previous information is transformed to the later time step.

The complexity of modules in *DL* model are dramatically increasing in the last several years to be sensitive to minute details so that the abstracted representations are more meaningful and precise for specific targets (Krizhevsky et al., 2012; Simonyan and Zisserman, 2015). A state-of-the-art image classification method can use a structure with more than 150 layers (He et al., 2016). The realization of such a huge multilayer structure benefits from the hardware development like Graphics Processing Unit (*GPU*). The enormous calculation ability accelerated by high-performance hardware architecture make it possible to construct the huge deep-

learning modules combination. Normally, purely accumulating the deep-learning modules may not lead the weights of the model to converge. Several reasonable training strategies like *batch-normalization* (Ioffe and Szegedy, 2015) and *drop-out* (Srivastava et al., 2014) also make a contribution to the deep-learning development.

Typical deep-learning modules include pure Deep Neural Networks (*DNN*), Convolutional Neural Networks (*CNN*) (Le-Cun et al., 1990), and their transformations like Recurrent Neural Networks (*RNN*) (Pineda, 1987), Long-Short Term Memory (*LSTM*) (Hochreiter and Schmidhuber, 1997), *etc*. As shown in Figure 1, *CNN* is a combination of convolutional processing with learned filters. Normally a *CNN* model also consisted of pooling layers for dimension reduction and activation layers (*Sigmoid*, *ReLU*, *Leaky-ReLU* (Maas et al., 2013)) for the non-linearity increasing.

Except for instant input, the output of *RNN* is also influenced by the prior information. Like *CNN* share features through space, *RNN* can be regarded as a feature sharing structure along time. As shown in the left of Figure 3, there is a loop in the inner net of *RNN* so that information of last time step is allowed to be kept in the neuron. It can also be regarded as connected networks between inputs at different time steps as shown in the right of Figure 3, so that information can be transported from the previous step to the next. Each neuron passes a message to a successor. The special framework of *RNN* is particularly efficient for sequence input as in speech recognition and natural language processing. However, when the output is influenced by an input where there is a long time gap between them, it is unpredicted that *RNN* neurons can still keep the related information. Further as an extension, *LSTM* builds the long-term dependencies between the sequence of outputs. The advantage of RNN and LSTM for sequence input provides the same convenience for the continuous states of robotics. Several related applications are introduced in Section. III-A.

As for deep-learning models, these mentioned structures are multi-layer nonlinear models and can be trained through gradient descent methods. They can be used directly for feature representations in supervised or unsupervised learning. For reinforcement learning, it is more like an optimization problem. The feature representation takes partial samples from the entire problem space. Mnih et al. (2013, 2015) defined Deep Reinforcement Learning (*DRL*) by using a *CNN* framework for feature extractions through screen displays of Atari games. Not like the ground-truth setting from a human in supervised learning, the agent in *DRL* is self-motivated through the interactions with the environment based on the feedback scores. It surpassed the performance of all previous algorithms and even accomplished a level comparable with the best human player across a test set of 49 games. Various problems in robotics can be considered as problems of reinforcement learning naturally (Kober et al., 2013), so the success of deep reinforcement learning motivates a lot of breakthroughs in robotics which are introduced in Section III-B.

Deep-learning modules are often regarded as black boxes, which bring lots of uncertainty. Actually, the convolutional filter is a typical image processing method and *CNN* can be re-

garded as a deformation of the traditional mathematical model. For example, Deformable Part Models (*DPMs*) (Felzenszwalb et al., 2010) as graphical models (Markov Random fields), were once the state of art in object detection task for RGB images. Girshick et al. (2015) unrolled the *DPM* algorithm and represented each step in the *DPM* to an identical *CNN* layer. Based on this synthesis, a learned feature extractor can replace the hand-crafted feature like histograms of oriented gradients (*HOG*) and be connected with the *DPM* directly. Zheng et al. (2015) implemented Conditional Random Fields (*CRFs*) with Gaussian pairwise potentials as *RNN*, which also combined the strengths of the probabilistic graphical model and the deep-learning model. Because of the integration of these models, both of the two projects made it possible to train the whole deep network end-to-end. Finally, they dramatically improved the state of art in object detection and semantics segmentation task in conventional computer vision. In the regard of multi-level convolution, the so-far most popular hand-crafted feature *SIFT* (Lowe, 1999, 2004), as a Gaussian Pyramid with multi-scale convolutional filters can also be taken as a *CNN* model. We consider it is the essential reason why *CNN* can replace the effects of a similar generic hand-crafted feature, in applications such as place recognition.

For conventional robotics researchers, the review about deep-learning (LeCun et al., 2015) (in aspects of data-mining or image processing) written by Yann LeCun, Yoshua Bengio, and Geoffrey Hinton is highly recommended for further analysis and comparison.

### C. Deep-learning in robotics: challenges

Deep-learning has shown great potential in artificial intelligence. As a relative of artificial intelligence, robotics provides a fabulous playground for deep-learning. However, robotics hold several unique challenges for learning algorithms compared with other fields.

**Uncertainty:** Robots do not always operate in controlled and structured environment. Autonomous cars drive in outdoor environments that may be crowded with pedestrians and vehicles; service robots run in indoor environments that may contain variously appearances; industrial robots may need to support a wide range of products; aerial robots can go across low-altitude hybrid space, etc. Unlike the conventional distribution estimation in data analysis (*aka.* distributional uncertainty), significant uncertainty may propagate unboundedly in robotics applications in practical environments. Although the assumptions of the noise model may solve partially a number of practical problems, such as regular SLAM (Cadena et al., 2016), it is still difficult to solve a variety of practical problem with conventional methods.

Deep-learning is appropriate to solve, at least in part, some of the problems with stronger uncertainty with the help from massive available data. However, the success of deep-learning is often attributed to the extremely high number of parameters which is particularly useful for robotics. That is why deep-learning is also criticized by serious over-fitting to training samples. The dynamics in environments ask for a much higher capability to model robustness. Lenz (2016) showed that over-fitting of deep-learning in robotics can be reduced by carefully

designed networks of the deep-learning model, but there are still demanding research in this direction.

**Evaluation Metrics** For robots tasks, the output of the learning algorithms, no matter perception or control, are aimed to make decisions, like whether the autonomous car should stop before a crossing street. However, for conventional computer vision tasks, like classification and object detection, the output is primarily a label prediction where the precision is highly emphasized regardless the predictive confidence. But for robotics, the decision has to be checked carefully based on the confidence especially with series of prior and posterior when the human is in potentially hazardous conditions. A single false output makes less sense for image classification, but a false command can lead to disaster results in robotic practices.

Grimmett et al. (2013, 2015) proposed the definition of introspection for robotics learning algorithms which is also necessary for deep-learning methods. The classifier with high introspection only makes mistakes with high uncertainty and output correct result with high certainty, which is the definition of the introspective quality of learning algorithms. Generally speaking, because of the extreme uncertainty in robotics, traditional metrics may be insufficient to characterize system performance of deep-learning algorithms for robotics.

**Continuous States and Actions** Note that besides the high-dimensional data and various uncertainty, robotics are often connected with continuous actions and states and the states of robotics are often non-deterministic. The general assumption like noise free and totally visible is impractical for robotics. Because the cluster of small errors may lead to a totally different consequence, the maintenance of uncertainty is undoubtedly necessary. As a partially observed system, sometimes true states should be estimated by filters (Kober et al., 2013), e.g. such problems are always critical in reinforcement learning procedures with real robot platforms.

Another problem for learning in robotics is that it is often unrealistic to implement the training procedure in the real world, such as that for reinforcement processes. The trial-and-error process may lead to serious damage to real systems. Most of the reinforcement learning applications in robotics are trained in simulation environments (Tai and Liu, 2016a; Zhang et al., 2016a). However, model learned in simulation alone unfortunately cannot be directly transferred to the real-world task. With the development of deep-learning, this problem can be partially solved by suitable initialization and post-tuning with real world samples with transferred learning results(Tai and Liu, 2016c).

## II. DEEP-LEARNING FOR PERCEPTION IN ROBOTICS

Perception in robotics is quite similar to the conventional computer vision tasks like image classification, object detection, and semantics segmentation, where most of the models are trained with supervised datasets as ground truth. In this section, perception techniques for robotics fall into two categories: methods to extract parts of the scene such as object detection, and those describe the whole environment for place recognition. Generalized mobile robotics research

TABLE I
THE SURVEY OF DEEP-LEARNING IN ROBOTIC PERCEPTIONS, BASED ON THE SENSOR USED, MODEL CATEGORIZATION, DATASET SOURCE, REAL-TIME
TEST, EQUAL EFFECT AS A COMPUTER VISION TASK, APPLICATION, TRAINING EFFICIENCY AND PREDICTION PRECISION.

| Approach | Sensor | Deep Model | Dataset Source | Real-time Test | Effect | Application | Training Efficiency | Precision |
|---|---|---|---|---|---|---|---|---|
| Lenz et al. (2015b) | RGBD | DNN | Benchmark | ✓ | Object Detection | Grasping | ★★ | ★ |
| Redmon and Angelova (2015) | RGBD | CNN | Benchmark | ✓ | Object Detection | Grasping | ★ | ★★ |
| Li et al. (2016) | 3D Lidar | CNN | Benchmark | | Object Detection | Auto Drive | ★ | ★ |
| Maturana and Scherer (2015a) | Lidar | 3D Conv | Self-Labeled | ✓ | Classification | Aerial | ★★ | ★★ |
| Engelcke et al. (2016) | 3D Lidar | 3D Conv | Benchmark | ✓ | Object Detection | Auto Drive | ★★ | ★★ |
| Yang et al. (2016a) | RGB | CNN | Self-Labeled | ✓ | Segmentation | SLAM | ★★★ | ★★★ |
| Oliveira et al. (2016) | RGB | CNN | Benchmark | ✓ | Segmentation | Auto Drive | ★ | ★★★ |
| Sünderhauf et al. (2015a) | RGB | CNN | Benchmark | ✓ | Classification | Auto Drive | ★ | ★★ |
| Sünderhauf et al. (2015b) | RGB | CNN | Benchmark | ✓ | Object Detection | Auto Drive | ★★ | ★★★ |

also includes robot arms, so the related applications of deep learning in grasping and manipulation of robot arms were also surveyed. Several main related publications in grasping, place recognition of autonomous driving and SLAM based on various metrics are listed in Table I.

### A. Object detection

The state-of-the-art object detection technique in computer vision is based on region proposal algorithms by sharing convolutional features with the detection network (Ren et al., 2015; Redmon et al., 2016) and combined with the features from intermediate layers in CNN (Liu et al., 2016). The model, trained end-to-end, can output both the bounding box and the class of the objects from the single image directly. The same strategy was used for pedestrian detection for autonomous cars (Angelova et al., 2015a; Chen et al., 2016c; Angelova et al., 2015b).

Besides RGB images, RGB-D data is also quite common nowadays and useful for robotics. Eitel et al. (2015) built a multi-modal learning framework by taking both the RGB and Depth image as input and processing them separately for object recognition. Considering the fact that there is always unpredicted noise in the real world, they proposed a depth augmentation strategies, especially for robotics. Gupta et al. (2014) used *CNN* to learn the features from the geocentric encoding of the depth image, where object detection, semantics segmentation and instance segmentation were output from the model directly.

Robot grasping is inherently a detection problem. Traditionally it was solved through matching with full 3-D model (Miller and Allen, 2004; León et al., 2010; Pelossof et al., 2004), where a strong prior information is necessary. Jiang et al. (2011) and Zhang et al. (2011) converted that to a detection problem. It can then be solved through deep-learning methods with object detection paradigm. As for robotic grasping in their work, the goal is to derive the possible location where the robot can execute grasping action through gripper. Compared with conventional object detection problem, robust performance and fast computation in grasp detection as part of

the control loop are extremely required so that the given object can maximize the possibility of successful grasping efficiently. An effective method in this field is required. Deep-learning provides an effective solution.

Lenz *et al.* successfully used a two-stage *DNN* to locate the grasp places through RGB-D information Lenz et al. (2015b). A small-scale network was used to detect the potential and coarse grasp positions exhaustively and the other larger-scaled network was used to find the optimal graspable position. Note that the convolutional networks here were taken as a classifier in a sliding window detection pipeline, which was extremely time-consuming. Redmon *et al.* trained the grasp detection end to end through another convolutional networks Redmon and Angelova (2015). Both of the detection accuracy and the computation speed were extremely improved. By integrating the bounding regression and object classification procedures, this model ran 150 times faster than previous methods. Redmon and Angelova (2015) also introduced a grid strategy as YOLO (Redmon et al., 2016) to propose multiple grasp positions in a single input. Further, Johns et al. (2016) used CNN to map the depth image to a grasp function by considering both the uncertainty of the grasp prediction and the location of the gripper.

Sparse point cloud captured from 3D Lidar like *Velodyne* [1] are widely applied in autonomous vehicles. Such kind of information is quite similar to the depth image from an RGB-D sensor, but with only sparse representation and mostly in an unorganized format. Compared with RGB or RGB-D images, they provide a larger range of depth and are not influenced by lighting conditions. Deep-learning is also widely used for pedestrian and vehicle detections in autonomous cars through sparse 3D points. Li et al. (2016) projected the 3D point cloud from *Velodyne* to a 2D point map and used a 2D convolutional network to achieve vehicle detection. Conventional 2D convolutional layer was also extended to 3D convolutional layer so that 3D point cloud can be processed directly (Chen et al., 2016b; Li, 2016; Wang and Posner, 2015; Engelcke et al., 2016; Maturana and Scherer, 2015b).

[1] http://velodynelidar.com

Amongst, Chen et al. (2016a) used a combined network of taking the bird view and the front view of the Lidar and the RGB image all together as the input. Engelcke et al. (2016) combined voting in sparse point cloud with 3D convolutional sliding window and saved a lot of useless calculation in sparse area. Maturana and Scherer (2015a) applied 3D convolutional neural networks to detect the landing zone for aerial robots.

Based on the success of Fully Convolutional Networks (*FCN*) (Long et al., 2015) on semantics segmentation, Yang et al. (2016a) built a *CNN* model to extract the ground and wall planes from a single image. Based on these plane features extracted by the deep-learning method, a SLAM framework (Yang et al., 2016b) was also implemented. Oliveira et al. (2016) used the same strategy for the road segmentation and lane prediction applied on the autonomous vehicles.

The success of *RNN* in object tracking (Ondruska and Posner, 2016) was also leveraged for autonomous driving. Dequaire et al. (2016) predicted future states of the objects based on the current inputs through a *RNN* model.

Some rarely used sensor information is also considered as the input for deep-learning model. Through tactile sensor on the end effector, object and material detection were also implemented through deep-learning (Schmitz et al., 2014; Baishya and Bauml, 2016). Valada et al. (2015) achieved terrain classification through the acoustic sensor mounted at the bottom of vehicles.

### B. Environment and place recognition

Visual place recognition is an extremely challenging problem considering the variety weather conditions, time-of-day, and environmental dynamics. Given an image of a place, mobile robots are requested to recognize a previously seen place (Lowry et al., 2016). With the development of autonomous cars, such kind of recognition is strongly demanding. The difficulties to realize robust place recognition include the perceptual aliasing (similar places), various viewpoints and other appearance variation during the revisits.

Trained in the dataset with millions of labeled images like *Imagent* (Deng et al., 2009), the state-of-art *CNN* model showed great generalities in different vision tasks including scene understanding (Sharif Razavian et al., 2014; Donahue et al., 2014). Even with simple linear classifiers, *CNN* features outperformed sophisticated kernel-based methods with hand-crafted features (Donahue et al., 2014). And it can quickly be transformed to other tasks with rarely labeled samples as references (Sharif Razavian et al., 2014). Even for the task like camera re-localization with extremely serious demand for place recognition precision, convolutional features also showed huge improvement compared with *SIFT*-based localizes (Kendall et al., 2015). Application in place recognition for mobile robots were naturally motivated (Chen et al., 2014; Sünderhauf et al., 2015a).

Sünderhauf et al. (2015a) tested the performance of several popular convolutional networks models for the global feature abstraction of place recognition. It showed that networks trained for semantic place categorization (Zhou et al., 2014; Liao et al., 2016b) were more effective than those trained

for object detection (Girshick et al., 2014). The utility of convolutional features for visual place recognition in robotics was proved as well. It also proposed that mid and high-level features from the *CNN* models were more robust for the changes of appearance and viewpoint. Naseer et al. (2015) and Arroyo et al. (2016) showed that features extracted by convolutional models for place recognition were evidently stable for extreme changes across the fours seasons.

Sünderhauf et al. (2015b) proposed a pipeline similar to object detection but for visual place recognition. *CNN* models were adopted to recognize the landmarks which are located by Edge Boxes, and an object proposal method (Zitnick and Dollár, 2014) was used as traditional object detection. Based on the similar place objects (buildings, squares, *etc.*), conventional object proposals and recognition were stable enough over significant viewpoints and condition changes, which make it suitable for place recognition.

Arandjelovic et al. (2016) combined the features extracted from *CNN* models with Vector of Locally Aggregated Descriptors (*VLAD*) (Jégou et al., 2010). They built an end-to-end model for place recognition task compared with the former methods where *CNN* was only used for feature recognition (Chen et al., 2014). The end-to-end training motivated the model to be trained life-long time with the continuously collected images from Google Street View. The recognition precision for places on benchmark datasets was improved a lot.

The success of *CNN* features on place recognition of robots reciprocally provide much more possibility for interesting ideas, like matching the ground-level image with a database collected from aerial robots (Lin et al., 2015) and using 2D occupancy grids constructed by Lidar to classify the place categorization (into a corridor, doorway, room, etc.) (Goeddel and Olson, 2016).

### C. Future directions in DL-aided perception for robotics

*1) Multimodal:* There are various sensors adapted for robotics. Right now, the *DL*-related applications are mainly about extracting features from images, following the trend of deep-learning in computer vision. Multi-modal combination by taking different raw sensor information (RGB, RGB-D, Lidar, *etc.*) as input (Eitel et al., 2015) is still deserved to explore. Liao et al. (2016a) predicted the depth of the image by taking the raw image and a single-line 2D-lidar reflection as input, which showed that *CNN* models can help to find the essential relations between different sensor information. Further study is necessary to enhance the robustness of output.

*2) 3D Convolution:* Autonomous driving is one of the most popular topics in both robotics and artificial intelligence. As we mentioned in Section II-A, there have been some pedestrians and vehicles detection tasks through the widely used 3D Lidar on autonomous vehicles. However, up-to-date 3D convolution processes are simply extending the 2D convolutional filters to a 3D version. Wang and Posner (2015) and Engelcke et al. (2016) explored that voting could be regarded as a convolution process and built a combined model by taking voting as a layer of 3D convolution. We believe

such kind of effective combinations between the deep-learning and traditional approaches will bring further breakthroughs in perception.

*3) Pix-level place recognition:* In Section II-B, most introduced methods are based on global *CNN* features, and object detection pipelines were proposed to solve the place recognition problem. Besides that, semantic segmentation methods (Zheng et al., 2015; Long et al., 2015) can provide more precise features. The related high-level representations of the places can overcome the challenges for various appearance and viewpoints in the same scene.

*4) End-to-end:* An essential reason to perform *DL* with hierarchical convolutional operations (Girshick et al., 2015; Zheng et al., 2015; Engelcke et al., 2016) is to build the end-to-end model. The end-to-end model trained through back-propagation dramatically speed up the time for calculation (Arandjelovic et al., 2016). Compared with pure feature extractions such as that in (Sharif Razavian et al., 2014), end-to-end model brings a lot of efficiency and convenience. End-to-end solutions, aka model-less methods, to robotic problems are also promising for complex tasks.

## III. DEEP-LEARNING FOR CONTROL IN ROBOTICS

Compared with the perception related tasks, control problem is generally much more sophisticated: *i)* Control problem demands real-time performance, which means the time-consuming data collection and training procedure should be simplified for online systems. In real world application, the dataset is always limited in quantity and usually to be collected only for specified tasks. *ii)* The control problem is highly dynamic and usually non-linear. High order mathematical control model is totally different compared with convolution-based image model, where the latter is abstractly linear operations. *iii)* Control systems are conducted with the real world with uncertain circumstances decorated by various noise. *iv)* Despite the assumptions and conditions, stability is necessary for classical control analysis. Redundancy of the system is necessary, as the failure often leads to considerable sacrifice.

Note that the robotic control problems are often model-based and non-data-driven, so that it is irrelevant to the previous categorization for learning methods. However, as for a control system, we can consider it as a *black box*, which takes input and generates output. Such a definition provides the ground that we may consider control problem as a learning problem to this black box with sufficient data collected by either demonstration or generated target data. With this regard, we categorize the robotic control problems by *DL* to the following three types:

- **Deep data-driven sensory-motor system.** It directly maps the raw sensor inputs to control commands or build the belief on the pair-wise sensor-control information, including most of the model-free data-driven deep-learning control methods.
- **Deep Reinforcement Learning**. Referring to the conventional reinforcement learning methods, these methods train a representation of the policy learning models via *DNN*, *CNN*, *RNN*, *etc*. Based on the different learning

targets, Deep Reinforcement Learning (*DRL*) related algorithms can be further divided into *value-based* and *policy-optimization* methods.
- **Deep Model-Predictive Control.** Based on the conventional Model-Predictive control (*MPC*) and feed-forward control, a dynamics transition model of the control agent is learned by *DNN*. The data can be not only collected and labeled by human supervision, but also generated through simulation based on the learned models.

### A. Deep data-driven sensory-motor system

The data-driven sensory-motor control attempts to find the relation between the raw sensor inputs and the control commands directly. When taking the visual sensor information as a reference, it is named as visual servoing or visual-feedback control (Kragic et al., 2002). Traditional visual-based control methods depend on human designed features for feedback control (Corke, 1993; Crétual and Chaumette, 2001b; De Luca et al., 2008). Prior to visual servoing, visual sensing and manipulation are two separated processes such as 'looking' and 'moving'. The precision of this combined operation relies on the accuracy of the sensor and the controller of the manipulator. Visual-feedback control loop increases the overall accuracy of the system. Now it is widely used in mobile robot control (Fang et al., 2012; Liu et al., 2013), UAV control (Lee et al., 2012), target tracking (Crétual and Chaumette, 2001a; Altuğ et al., 2005), and robot manipulation (Ruf and Horaud, 1999).

The advance of deep-learning in image processing indicates its potential to solve visual-based robot control problems. However, even though *CNN* related methods accomplished lots of breakthroughs and challenging benchmarks about computer vision, the application in visual servoing is still less prevalent compared with other perception tasks such as object detection and image recognition. Especially it performs badly when generalized with unknown scenarios, unfamiliar objects or new appearances, due to lack of training data. Notice that human also make decisions of actions involving a visual feedback. We believe such generalized visual control problems are solvable using *DL* with further research effort in the near future.

Vision-based navigation is a fundamental technology for mobile robots as an extension to visual-feedback control. Based on supervised deep-learning, Muller et al. (2005) mapped raw image pairs to steering commands (left, straight, right) directly through convolutional networks for obstacle avoidance. Giusti et al. (2016) used the similar method to train a UAV to avoid trees in a forest through a monocular camera mounted on it. Tai et al. (2016) trained a deep-learning model for a mobile robot to explore in an indoor corridor environment based on single depth image from a RGB-D sensor as shown in Figure 1. Unlike the direct classification (Muller et al., 2005) of the moving commands, Giusti et al. (2016) and Tai et al. (2016) took the output of the *Softmax* layer multiplied with the moving commands coefficients to smooth the output steering angles. DeepDriving (Chen et al., 2015) estimated several indicators for affordance of driving actions using *CNN* models, but not directly mapping sensor

input to driving commands (Muller et al., 2005; Giusti et al., 2016; Tai et al., 2016; Bojarski et al., 2016). *CNN* models provided more comprehensive model of the environment for decisions making in driving behaviors.

Most mentioned methods intend to generate a control command directly through classification (for discrete control command) or regression (for continuous control command). However, the model between the sensor input and control command may not be direct especially considering temporal processes. Levine et al. (2016b) built a success prediction network for robotic grasping, which can also continuously control the robotic manipulator to accomplish the grasping task. The output of the deep-learning model is the grasp success probability.

Besides visual servoing, several works used *CNN* models to realize outdoor scene segmentation (Sermanet et al., 2009; Hadsell et al., 2009) based on stereo vision. Multi-range traversability and moving cost-map were estimated as the output of the *CNN* models through end-to-end training. Then, topological mapping and planning tasks for robot navigation were implemented based on the segmentation results.

For a long time path planning or global motion planning, not only the instant sensor input but also the target position are necessary as input/references. Zhang et al. (2016a) trained a *CNN* model to recognize the target point coordinates from simulated images. The model was afterward fine-tuned through real-world images. The output of the target recognition was taken as the input for motion planning of robot arms. Pfeiffer et al. (2016) directly added target coordinates to the fully connected layer paralleling to the sensor feature extraction through *CNN*. The model can also be trained end-to-end for mobile robot motion planning. Additional to single-robot tasks, Long et al. (2016) proposed a multi-agent collision avoidance policy by *DNN* through the observations about the conditions of other moving agents.

### B. Deep Reinforcement Learning

Robotics problems are often task-based. For a task-based problem with temporal structure, reinforcement learning is an appropriate approach (Kober et al., 2013). A variety of robotics control problems are solvable by reinforcement learning, such as inverted helicopter flight learning (Kim et al., 2003; Ng et al., 2006), gliding aircraft control (Chung et al., 2015), mobile robot navigation (Kretzschmar et al., 2016), modular robot control (Varshavskaya et al., 2008) and biped walking (Benbrahim and Franklin, 1997).

For a standard reinforcement learning problem as shown in Figure 4, the final target of the agent is to learn an optimal policy to control the system with states $s \in \mathcal{S}$ and actions $a \in \mathcal{A}$, where $\mathcal{S}$ is the state space and $\mathcal{A}$ is the action space. The inner dynamics transition model of the agent is represented as $p(s_{t+1} \mid s_t, a_t)$. In each time step $t \in [1, T]$, based on the policy ($\pi(a \mid s)$ for stochastic policy, or $a = \pi(s)$ for deterministic policy), an action $a_t$ is chosen to execute and a reward $r(s_t, a_t)$ is observed from the environment. The system can be infinite horizon when $T = \infty$. With the optimal policy, the sum of the final reward is $R_t = \sum_{i=t}^{T} \gamma^{i-t} r(s_i, a_i)$. Here $\gamma$
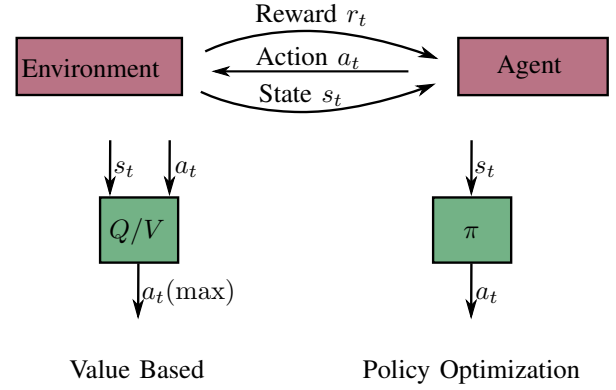


Fig. 4. Different Reinforcement Learning methods. The upper part shows the standard reinforcement learning framework and the interactions between the environment and the control agent. The action is chosen based on the reward and the current state. The lower part compares the two different reinforcement learning methods. The value or policy are taken as the learning target for value-based and policy-optimization methods.

is a discount factor to allocate different weights for earlier and later rewards (Sutton and Barto, 1998). For a specific policy $\pi$, the value function for specific state $s_t$ as initialized state is define as $V(s_t)$. The value function (Q value) for action-state pair $(s_t, a_t)$ is defined as $Q^\pi(s_t, a_t)$.

$$V^\pi(s_t) = \mathbb{E}[R_t \mid s_t, a_t = \pi(s_t)]$$

$$Q^\pi(s_t, a_t) = r(s_t, a_t) + \gamma V^\pi(s_{t+1})$$

The optimal policy should be the one with maximum expectation reward in the future as

$$\pi^* = \arg\max_\pi V^\pi(s)$$

As a general framework for representation learning, reinforcement learning is reinforced through the deep-learning to automate the design of feature representations from raw inputs, which is called deep reinforcement learning (DRL). Before the application of deep reinforcement learning method like Deep Q-Network *DQN* (Mnih et al., 2015), several reinforcement-learning-based methods used multi-layer neural networks to train the autoencoder for dimension reduction of the state's representation.

*State representation* aims to extract principle components of the original data which are sufficient to describe the full states, but the representation should also be able to induce fast convergence of the learning algorithm (Böhmer et al., 2015). As robot applications prefer to process necessarily few states to keep the efficiency, an auto-encoder can be trained to find a latent state representing the original image. With encoder map $\phi : z \to x$ and decoder map $\psi : x \to z$, minimize the least-squares reconstruction error of all the training samples $\{x^t\}_{t=1}^n$:

$$\min_{\phi,\psi} \sum_{t=1}^{n} \|\psi(\phi(x^t)) - x^t\|$$

To capture enough necessary information for describing the original data and improve the efficiency, multilayer neural networks were used to train the auto-encoder model (Lange and

Riedmiller, 2010; Lange et al., 2012; Finn et al., 2015). Lange and Riedmiller (2010) defined the *deep-fitted Q* algorithm to reduce the dimension of the raw input image as the state of the reinforcement learning agent. With this abstracted image representation through neural networks, control policy is collected by a reinforcement learning framework separately. It was applied to a mobile vehicle control successfully. However, the drawback of this method is that it may be hard to train an autoencoder with input including high variability.

Based on the different optimization targets, we categorize *DRL* into value-based and policy-optimization methods, as the categorizations for standard reinforcement learning as shown in Figure 4.

*1) Value-based:* The value function is an expectation for the future reward based of the state as $V(s_t)$ or the Q-value as $Q(s_t, a_t)$. For Q learning, which is based on the optimization of the Q value, the learned deterministic policy is

$$a_t = \pi(s_t) = \arg\max_{a_t} Q^*(s_t, a_t)$$

Q value can be iteratively updated or calculated by approximation. Mnih et al. (2013, 2015) estimated the Q-value through a *CNN* model with parameter $\theta$ as $Q(s, a; \theta) \approx Q^*(s, a)$. The transition $(s_t, a_t, r_t, s_{t+1})$ of every time step were saved as memory for iteratively updating. Raw RGB images were taken as the state $s$. Based on Bellman Equation, the real time optimal Q-value for $(s_t, a_t)$ can be estimated as

$$Q^*(s_t, a_t) = \mathbb{E}[r(s_t, a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta_i)|s_t, a_t]$$

The loss function to update the parameters $\theta$ is

$$L(\theta_i) = \mathbb{E}[(y_i - Q(s_t, a_t; \theta_i))^2]$$

where $y_i$ is the target and it is calculated through the equation for $Q^*(s_t, a_t)$ mentioned above based on the instant reward and the estimation for next state $s_{t+1}$ under the previous parameters.

In the training process, $y_i$ was the training target, which was calculated by the previous iteration result $\theta_{i-1}$. The gradient of the loss function is shown below:

$$\nabla_{\theta_i} L(\theta_i) = \mathbb{E}[(y_i - Q(s_t, a_t; \theta_i))\nabla_{\theta_i} Q(s_t, a_t; \theta_i)]$$

So far we introduced the Deep-Q-Network (*DQN*) as value-based reinforcement learning algorithm (Mnih et al., 2013, 2015). The observations of *DQN* are the raw images from the screen display of Atari 2600 games. Both the typical model-free Q-learning method and the convolutional feature extraction structures are combined as a human level controller. The learned policies beat human players and the previous algorithms in most of the Atari games.

Generally, *DQN* used a *CNN* framework to be the function approximator for a controller taking images as observation states. Notice that this was not the first time neural network was used to be the function approximator. There were several reasons inducing the success of *DQN* (Böhmer et al., 2015): *i)* Compared with a traditional shallow neural network, *DQN* used an overly large *CNN* to estimate the Q-function. That was pretty powerful to extract representational information of raw image input. *ii)* The deep *CNN* framework also benefited

from the GPU technology improvement. High efficient computation hardware decreased the training difficulty of this time-consuming task. *iii)* In the training process, transition groups of the related $(s, a, r, s')$ was stored, then it was supplied to the optimization of Q-function multiple times. Robustness of the system was therefore improved significantly.

Due to the potential of automating the design of data representations, deep reinforcement learning abstracted considerable attentions recently (Duan et al., 2016). Following the success of *DQN* (Mnih et al., 2015), revised deep reinforcement learning methods appeared to improve the performance on various of applications in robotics (Hausknecht and Stone, 2015; Wang et al., 2016).

Tai and Liu (2016a,c) used *DQN* to help the robot learn obstacle avoidance in a totally unfamiliar environment in a simulated environment. Here the observation input was raw depth image from an RGBD sensor mounted on a mobile robot. The result indicated that the cognitive ability of the robot was extended by deep reinforcement learning model for more complicated indoor environments in an efficient online-learning process continuously. Zhang et al. (2015) used *DQN* to train a target reaching task with a three-joint robot manipulator. The only input is the visual observation of the robot arm condition. The result in simulated environment got a consistent success rate.

A drawback of the origin *DQN* is that it is not adapted for the continuous task to find the maximum $Q(s, a)$ value. Gu et al. (2016b) used deep neural networks to estimate the decomposed $Q$ value which was the advantage term $A$ and the state value term $V$. It outputs a value function $V(s)$ and an advantage term $A(s, a)$ (*NAF*) instead, which can help to solve continues control problem. It was applied to various robot manipulation skills like the door opening (Gu et al., 2016a) by a robot arm.

*2) Policy optimization:* Several disadvantages of value-based method limited its application: The calculation of $Q$ value require for discreet and low dimensional action space to find $\arg\max_a Q^\pi(s, a)$; $V$ value cannot directly induce the policy to further identify the optimal action. Policy-based reinforcement learning directly search the optimal policy $\pi^*$ to achieve the maximum future reward which provides an accessible way for continuous control. They use the weights $\theta$ to parametrize the policy of the system, such as a the deterministic policy can be formatted as $a = \pi(s, \theta)$. For a specific policy, the sum of future reward to maximize is

$$J(\theta) = \mathbb{E}[R_t | \pi(s, \theta)]$$

If $a$ is continues and $Q$ is differentiable, the gradient of a deterministic policy is given by

$$\frac{\partial J(\theta)}{\partial \theta} = \mathbb{E}\left[\frac{\partial Q^\pi(s, a)}{\partial a} \frac{\partial a}{\partial \theta}\right]$$

Lillicrap et al. (2015) proposed an policy-based deep reinforcement learning framework (*DDPG*) on actor-critic algorithm (Sutton and Barto, 1998) to extend *DQN* to the continuous action domain. Two deep networks were built to update the value and the policy separately. It was successfully applied on more than 20 simulated classic control tasks like

TABLE II

THE SURVEY OF DEEP REINFORCEMENT LEARNING IN ROBOTICS BASED ON DIFFERENT LEARNING TARGET, THE REQUIREMENT FOR DYNAMICS MODEL, WHETHER THEY CAN BE APPLIED TO CONTINOUS CONTROL AND THEIR TRAINING EFFICIENCY

| Approach | Value Estimation | Policy Optimization | Model Requirement | Continuous Control | Training Efficiency | Application |
|---|---|---|---|---|---|---|
| DQN (Mnih et al., 2015) | ✓ | | | | ★ | Exploration (Tai and Liu, 2016c) Motion Control (Zhang et al., 2015) |
| DDPG (Lillicrap et al., 2015) | ✓ | ✓ | | ✓ | ★ ★ | UAV control (Zhu et al., 2016) Loop Closing (Mirowski et al., 2016) |
| NAF (Gu et al., 2016b) | ✓ | | | ✓ | ★ ★ | Manipulation (Gu et al., 2016a) |
| GPS (Levine et al., 2016a) | | ✓ | ✓ | ✓ | ★ ★ ★ | Tensegrity (Geng et al., 2016) UAV control (Zhang et al., 2016c) |

legged locomotion and cartpole swing-up. Further, Mnih et al. (2016) built an asynchronous gradient descent strategy for deep reinforcement learning which can be extended onto Q-learning, SARSA (Sutton and Barto, 1998) and actor-critic as Advantage Actor-Critic (*A3C*). The parallel learners greatly stabilized the learned policy.

Deep vision-based policy optimization methods were also applied in target-driven UAV control (Zhu et al., 2016) and UAV obstacle avoidance (Sadeghi and Levine, 2016). Mirowski et al. (2016) used the similar policy-based deep reinforcement learning method to learn target-driven navigation.

Model-free methods like *DQN* and *DDPG* require a huge number of samples to train the networks. Without the dynamics transition model, training samples have to be collected from the real rollouts. However, when the dynamic model $p(s_{t+1}|s_t, a_t)$ is known prior, the training samples can be directly derived from the policy $\pi$ and such a model is also known as model-based reinforcement learning. It is believed (Atkeson and Santamaria, 1997) that model-based reinforcement learning is more data efficient than direct reinforcement learning while it can identify better policies, plans, and trajectories. Even the dynamics model is unknown, a constructed linear-Gaussian controller can be used for policy optimization (Levine and Abbeel, 2014). Then KL-divergence was leveraged to control the updating bound of the policy in the region with the higher reward which was called guided policy search (*GPS*) (Levine and Koltun, 2013). GPS converted policy search into supervised learning to some extends without the real sample collection procedure. Through the alternative optimization between the policy and the trajectory distribution, and the minimization of the expected cost of the whole procedure, the policy was learned for general-purpose polices. *NAF* (Gu et al., 2016b) also took the advantage of model-based rollouts as auxiliary samplings for value-based model training.

*DNN*-based *GPS* (Levine et al., 2016a) learned to map from the pairwise raw visual information and joint states directly to joint torques. Compared with the previous work, it realized a high-dimensional control even from imperfect perception. It is widely used on robotics control, like robot manipulation (Zhang et al., 2016b), UAV control (Zhang et al., 2016c) and tensegrity robot locomotion (Geng et al., 2016). Through

end-to-end training, the learned control policies were more effective and generalize compared hand-engineered controllers especially for complex tasks. As a summary, the four representative *DRL* methods (*DQN*, *DDPG*, *NAF*, *GPS*) are compared in Table II based on their reinforcement learning characters, training efficiency and applications.

### C. Deep Model-Predictive Control

Robotics tasks are often involved with complex non-linear dynamics, especially for robot arms manipulation and locomotion. Hand-craft designation of the robotic continuous controllers is very difficult. In the last several years, Model-Predictive Control (*MPC*) (Garcia et al., 1989; Ou et al., 2013) is proven to be effective for many complicated robotic tasks through iterative, finite-horizon optimization. In Section III-B, we introduced the Markov Decision Process-based methods, where the dynamics transition model is $p(s_{t+1}|s_t, a_t)$. Through a meaningful prediction of this dynamics model, optimal control input can be calculated based on the predicted future outputs.

As mentioned, the complex non-linear model is significantly difficult to be designed accurately. Lenz et al. (2015a) leveraged a deep *RNN* to design a DeepMPC controller for food-cutting operations. Through the data from the human demonstration, the controller can be easily trained to converge based on back-propagation algorithms. Long-term recurrent features helped to keep the state information from previous observations effectively.

Agrawal et al. (2016) used a deep neural network to predict the dynamics model of robotic arm operation. Features extracted from raw sensor images are taken as the states of the system directly. Both the forward and inverse model for the prediction of the outcome were simultaneously learned. The forward mode can compute the predicted state which was the predicted image features. Further, Finn and Levine (2016) directly predict a visual imagination to push the objects to the desired place. As the state of the dynamics model in different time, the raw future image but not feature vector can be estimated directly. The video used for prediction model did not need human supervision in that method.

## D. Future directions and frontiers

*1) Multi task:* As a complex control agent, a robot is often occupied by several parallel tasks at the same time. Deep reinforcement learning also provides the possibility to learn multi-tasks together (Mujika, 2016). Mirowski et al. (2016) built a deep reinforcement learning model for target-driven navigation and the depth prediction. The model can even learn loop closure through combining the *LSTM* structure as well. The multi-tasks control based on deep-learning deserves many fundamental experiments for exploration.

*2) Deep inverse reinforcement learning:* Inverse reinforcement learning (*IRL*) (Ng et al., 2000; Abbeel and Ng, 2004) aims to learn a reward function or the control policy through the demonstrated behaviors based on the optimal policy. Wulfmeier et al. (2015) used a deep architecture to train the reward function based on Maximum Entropy paradigm for *IRL*. And it was successfully applied to a cost-map learning work for the mobile vehicle path planning (Wulfmeier et al., 2016). Finn et al. (2016) taught a high-dimensional robot system to learn behaviors like dish placing through deep inverse optimal control. Through the extraction ability of deep models, deep *IRL* may bring a lot of advantages for learning from demonstrations, like extracting efficient cost-map through the raw human demonstrations.

*3) Generalization of deep-learning control:* Deep-learning is also criticized for the generalization problem where the trained model can only be adapted to constraint conditions as in training. For example in our previous work (Tai and Liu, 2016b), a deep-learning model was used for the homing vector prediction in the mobile robot visual homing problem. As a basic and low-cost navigation ability, traditionally the homing vector was motivated by visual servoing algorithms (Liu et al., 2010, 2012, 2013). The result (Tai and Liu, 2016b) showed that when the same target as in training was used, the predicted homing vector was very precise. But randomly chosen image pairs can not be taken as the input to predict their relative vector. How to generalize the deep-learning model for mobile robotics control is still a challenge.

## IV. CONCLUSION

The problem of deep-learning experienced a great progress in the past five years. The application of deep-learning in robotics is just a beginning. There is a bright future for the widely spreading of deep-learning in robotics. Many classical robotics problems, like SLAM (Cadena et al., 2016) and visual place recognition (Lowry et al., 2016) begin to explore the possible breakthrough motivated by deep-learning. In this survey paper, special constraints for robotics tasks like continuous states and high uncertainty were discussed regarding deep-learning. The recent development of deep-learning and the highlighted applications on robotics are reviewed especially for robotics perception and control systems.

The perception task is very similar to pure computer vision task and it benefits a lot from the related deep-learning methods in computer vision tasks. To meet the constraint requirement for special sensor and calculation time, We highlighted several future directions in robot perception: the multimodal deep-learning structure by taking different sensor information, 3D convolution for detection in autonomous driving, pix-level semantics information to improve the understanding of the environment and the end-to-end training to speed up the calculation.

The robot control task was divided into data-driven sensory-motor control, reinforcement learning control and model-predictive control. The combination of deep-learning and traditional control policies can bring a lot of convenience and efficiency for the robot tasks. The potential of deep-learning to be applied in other special tasks like multi-tasks learning and Inverse Reinforcement Learning also deserves attention for related research.

In this survey, we present various research to answer the two questions: why we should use deep-learning to solve several typical challenging problems and why not use deep-learning to substitute the conventional methods for mobile robotics. Deep-learning may be an answer for the future of robotics and even artificial intelligence. Hundreds of amazing results appeared in computer vision area over the past several years, but there is still a long way and space for existing deep-learning architecture to serve robust and generic solutions for robotic tasks.

## REFERENCES

Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M et al. (2016) Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* .

Abbeel P and Ng AY (2004) Apprenticeship learning via inverse reinforcement learning. In: *Proceedings of the twenty-first international conference on Machine learning.* ACM, p. 1.

Agrawal P, Nair A, Abbeel P, Malik J and Levine S (2016) Learning to poke by poking: Experiential learning of intuitive physics. In: *Advances in Neural Information Processing Systems.*

Altuğ E, Ostrowski JP and Taylor CJ (2005) Control of a quadrotor helicopter using dual camera visual feedback. *The International Journal of Robotics Research* 24(5): 329–341.

Angelova A, Krizhevsky A and Vanhoucke V (2015a) Pedestrian detection with a large-field-of-view deep network. In: *2015 IEEE International Conference on Robotics and Automation (ICRA).* IEEE, pp. 704–711.

Angelova A, Krizhevsky A, Vanhoucke V, Ogale A and Ferguson D (2015b) Real-time pedestrian detection with deep network cascades. *Proceedings of BMVC 2015* .

Arandjelovic R, Gronat P, Torii A, Pajdla T and Sivic J (2016) Netvlad: Cnn architecture for weakly supervised place recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* pp. 5297–5307.

Arroyo R, Alcantarilla PF, Bergasa LM and Romera E (2016) Fusion and binarization of cnn features for robust topological localization across seasons. In: *IEEE/RSJ IROS.*

Atkeson CG and Santamaria JC (1997) A comparison of direct and model-based reinforcement learning. In: *In International Conference on Robotics and Automation.* Citeseer.

Baishya SS and Bauml B (2016) Robust material classification with a tactile skin using deep learning. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2016*.

Bay H, Ess A, Tuytelaars T and Van Gool L (2008) Speeded-up robust features (surf). *Computer vision and image understanding* 110(3): 346–359.

Bay H, Tuytelaars T and Van Gool L (2006) Surf: Speeded up robust features. In: *European conference on computer vision*. Springer, pp. 404–417.

Benbrahim H and Franklin JA (1997) Biped dynamic walking using reinforcement learning. *Robotics and Autonomous Systems* 22(3): 283–302.

Bengio Y (2009) Learning deep architectures for ai. *Foundations and trends in Machine Learning* 2(1): 1–127.

Bishop CM (2006) Pattern recognition. *Machine Learning* 128.

Böhmer W, Springenberg JT, Boedecker J, Riedmiller M and Obermayer K (2015) Autonomous learning of state representations for control: An emerging field aims to autonomously learn state representations for reinforcement learning agents from their real-world sensor observations. *KI-Künstliche Intelligenz* 29(4): 353–362.

Bojarski M, Choromanska A, Choromanski K, Firner B, Jackel L, Muller U and Zieba K (2016) Visualbackprop: visualizing cnns for autonomous driving. *arXiv preprint arXiv:1611.05418* .

Cadena C, Carlone L, Carrillo H, Latif Y, Scaramuzza D, Neira J, Reid I and Leonard JJ (2016) Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics* 32(6): 1309–1332.

Cambria E and White B (2014) Jumping nlp curves: a review of natural language processing research [review article]. *Computational Intelligence Magazine, IEEE* 9(2): 48–57.

Chen C, Seff A, Kornhauser A and Xiao J (2015) Deepdriving: Learning affordance for direct perception in autonomous driving. In: *Proceedings of the IEEE International Conference on Computer Vision*. pp. 2722–2730.

Chen LC, Papandreou G, Kokkinos I, Murphy K and Yuille AL (2016a) Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915* .

Chen X, Huimin M, Ji W, Bo L and Tian X (2016b) Multi-view 3d object detection network for autonomous driving. *arXiv preprint arXiv:1611.07759* .

Chen X, Kundu K, Zhang Z, Ma H, Fidler S and Urtasun R (2016c) Monocular 3d object detection for autonomous driving. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 2147–2156.

Chen Z, Lam O, Jacobson A and Milford M (2014) Convolutional neural network-based place recognition. *arXiv preprint arXiv:1411.1509* .

Chung JJ, Lawrance NR and Sukkarieh S (2015) Learning to soar: Resource-constrained exploration in reinforcement learning. *The International Journal of Robotics Research* 34(2): 158–172.

Corke PI (1993) Visual control of robot manipulators-a review.

Visual servoing 7: 1–31.

Crétual A and Chaumette F (2001a) Application of motion-based visual servoing to target tracking. *The International Journal of Robotics Research* 20(11): 878–890.

Crétual A and Chaumette F (2001b) Visual servoing based on image motion. *The International Journal of Robotics Research* 20(11): 857–877.

De Luca A, Oriolo G and Giordano PR (2008) Feature depth observation for image-based visual servoing: Theory and experiments. *The International Journal of Robotics Research* 27(10): 1093–1116.

Deng J, Dong W, Socher R, Li LJ, Li K and Fei-Fei L (2009) Imagenet: A large-scale hierarchical image database. In: *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, pp. 248–255.

Deng L, Hinton G and Kingsbury B (2013) New types of deep neural network learning for speech recognition and related applications: An overview. In: *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, pp. 8599–8603.

Dequaire J, Rao D, Ondruska P, Wang D and Posner I (2016) Deep tracking on the move: Learning to track the world from a moving vehicle using recurrent neural networks. *arXiv preprint arXiv:1609.09365* .

Donahue J, Jia Y, Vinyals O, Hoffman J, Zhang N, Tzeng E and Darrell T (2014) Decaf: A deep convolutional activation feature for generic visual recognition. In: *ICML*. pp. 647–655.

Duan Y, Chen X, Houthooft R, Schulman J and Abbeel P (2016) Benchmarking deep reinforcement learning for continous control. In: *International Conference on Machine Learning (ICML-16)*.

Eitel A, Springenberg JT, Spinello L, Riedmiller M and Burgard W (2015) Multimodal deep learning for robust rgb-d object recognition. In: *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, pp. 681–687.

Engelcke M, Rao D, Wang DZ, Tong CH and Posner I (2016) Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. *arXiv preprint arXiv:1609.06666* .

Fang Y, Liu X and Zhang X (2012) Adaptive active visual servoing of nonholonomic mobile robots. *Industrial Electronics, IEEE Transactions on* 59(1): 486–497.

Felzenszwalb PF, Girshick RB and McAllester D (2010) Cascade object detection with deformable part models. In: *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*. IEEE, pp. 2241–2248.

Finn C and Levine S (2016) Deep visual foresight for planning robot motion. *arXiv preprint arXiv:1610.00696* .

Finn C, Levine S and Abbeel P (2016) Guided cost learning: Deep inverse optimal control via policy optimization .

Finn C, Tan XY, Duan Y, Darrell T, Levine S and Abbeel P (2015) Deep spatial autoencoders for visuomotor learning. *reconstruction* 117(117): 240.

Garcia CE, Prett DM and Morari M (1989) Model predictive control: theory and practicea survey. *Automatica* 25(3): 335–348.

Geng X, Zhang M, Bruce J, Caluwaerts K, Vespignani M, SunSpiral V, Abbeel P and Levine S (2016) Deep reinforcement learning for tensegrity robot locomotion. *arXiv preprint arXiv:1609.09049* .

Girshick R, Donahue J, Darrell T and Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 580–587.

Girshick R, Iandola F, Darrell T and Malik J (2015) Deformable part models are convolutional neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 437–446.

Giusti A, Guzzi J, Cirean DC, He FL, Rodrguez JP, Fontana F, Faessler M, Forster C, Schmidhuber J, Caro GD, Scaramuzza D and Gambardella LM (2016) A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters* 1(2): 661–667.

Goeddel R and Olson E (2016) Learning semantic place labels from occupancy grids using cnns. In: *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, pp. 3999–4004.

Grimmett H, Paul R, Triebel R and Posner I (2013) Knowing when we don't know: Introspective classification for mission-critical decision making. In: *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, pp. 4531–4538.

Grimmett H, Triebel R, Paul R and Posner I (2015) Introspective classification for robot perception. *The International Journal of Robotics Research* .

Gu S, Holly E, Lillicrap T and Levine S (2016a) Deep reinforcement learning for robotic manipulation. *arXiv preprint arXiv:1610.00633* .

Gu S, Lillicrap T, Sutskever I and Levine S (2016b) Continuous deep q-learning with model-based acceleration. In: *International Conference on Machine Learning (ICML-16)*.

Gupta S, Girshick R, Arbeláez P and Malik J (2014) Learning rich features from rgb-d images for object detection and segmentation. In: *European Conference on Computer Vision*. Springer, pp. 345–360.

Hadsell R, Sermanet P, Ben J, Erkan A, Scoffier M, Kavukcuoglu K, Muller U and LeCun Y (2009) Learning long-range vision for autonomous off-road driving. *Journal of Field Robotics* 26(2): 120–144.

Hausknecht M and Stone P (2015) Deep recurrent q-learning for partially observable mdps. In: *2015 AAAI Fall Symposium Series*.

He K, Zhang X, Ren S and Sun J (2016) Deep residual learning for image recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 770–778.

Hochreiter S and Schmidhuber J (1997) Long short-term memory. *Neural computation* 9(8): 1735–1780.

Ioffe S and Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *Proceedings of The 32nd International Conference on Machine Learning*. pp. 448–456.

Jégou H, Douze M, Schmid C and Pérez P (2010) Aggregating local descriptors into a compact image representation. In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, pp. 3304–3311.

Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S and Darrell T (2014) Caffe: Convolutional architecture for fast feature embedding. In: *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, pp. 675–678.

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

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

Kendall A, Grimes M and Cipolla R (2015) Posenet: A convolutional network for real-time 6-dof camera relocalization. In: *Proceedings of the IEEE International Conference on Computer Vision*. pp. 2938–2946.

Kim H, Jordan MI, Sastry S and Ng AY (2003) Autonomous helicopter flight via reinforcement learning. In: *Advances in neural information processing systems*. p. None.

Kober J, Bagnell JA and Peters J (2013) Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research* : 0278364913495721.

Kragic D, Christensen HI et al. (2002) Survey on visual servoing for manipulation. *Computational Vision and Active Perception Laboratory, Fiskartorpsv* 15.

Kretzschmar H, Spies M, Sprunk C and Burgard W (2016) Socially compliant mobile robot navigation via inverse reinforcement learning. *The International Journal of Robotics Research* 35(11): 1289–1307.

Krizhevsky A, Sutskever I and Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. pp. 1097–1105.

Lange S and Riedmiller M (2010) Deep auto-encoder neural networks in reinforcement learning. In: *Neural Networks (IJCNN), The 2010 International Joint Conference on*. IEEE, pp. 1–8.

Lange S, Riedmiller M and Voigtlander A (2012) Autonomous reinforcement learning on raw visual input data in a real world application. In: *Neural Networks (IJCNN), The 2012 International Joint Conference on*. IEEE, pp. 1–8.

LeCun Y, Bengio Y and Hinton G (2015) Deep learning. *Nature* 521(7553): 436–444.

LeCun Y, Boser BE, Denker JS, Henderson D, Howard R, Hubbard WE and Jackel LD (1990) Handwritten digit recognition with a back-propagation network. In: *Advances in Neural Information Processing Systems*. pp. 396–404.

Lee D, Ryan T and Kim HJ (2012) Autonomous landing of a vtol uav on a moving platform using image-based visual servoing. In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, pp. 971–976.

Lenz I, Knepper R and Saxena A (2015a) Deepmpc: Learning deep latent features for model predictive control. In: *Robotics Science and Systems (RSS)*.

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

Lenz IN (2016) *Deep Learning For Robotics*. PhD Thesis, Cornell University.

León B, Ulbrich S, Diankov R, Puche G, Przybylski M, Morales A, Asfour T, Moisio S, Bohg J, Kuffner J et al. (2010) Opengrasp: a toolkit for robot grasping simulation. In: *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*. Springer, pp. 109–120.

Levine S and Abbeel P (2014) Learning neural network policies with guided policy search under unknown dynamics. In: *Advances in Neural Information Processing Systems*. pp. 1071–1079.

Levine S, Finn C, Darrell T and Abbeel P (2016a) End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research* 17(39): 1–40.

Levine S and Koltun V (2013) Guided policy search. In: *ICML (3)*. pp. 1–9.

Levine S, Pastor P, Krizhevsky A and Quillen D (2016b) Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *arXiv preprint arXiv:1603.02199* .

Li B (2016) 3d fully convolutional network for vehicle detection in point cloud. *arXiv preprint arXiv:1611.08069* .

Li B, Zhang T and Xia T (2016) Vehicle detection from 3d lidar using fully convolutional network. In: *Proceedings of Robotics: Science and Systems*. AnnArbor, Michigan.

Liao Y, Huang L, Wang Y, Kodagoda S, Yu Y and Liu Y (2016a) Parse geometry from a line: Monocular depth estimation with partial laser observation. *arXiv preprint arXiv:1611.02174* .

Liao Y, Kodagoda S, Wang Y, Shi L and Liu Y (2016b) Understand scene categories by objects: A semantic regularized scene classifier using convolutional neural networks. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 2318–2325.

Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D and Wierstra D (2015) Continuous control with deep reinforcement learning. In: *Proceedings of the International Conference on Learning Representations (ICLR)*.

Lin TY, Cui Y, Belongie S and Hays J (2015) Learning deep representations for ground-to-aerial geolocalization. In: *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, pp. 5007–5015.

Liu M, Pradalier C, Chen Q and Siegwart R (2010) A bearing-only 2d/3d-homing method under a visual servoing framework. In: *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, pp. 4062–4067.

Liu M, Pradalier C, Pomerleau F and Siegwart R (2012) Scale-only visual homing from an omnidirectional camera. In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, pp. 3944–3949.

Liu M, Pradalier C and Siegwart R (2013) Visual homing from scale with an uncalibrated omnidirectional camera. *Robotics, IEEE Transactions on* 29(6): 1353–1365.

Liu RZ, Tang YY and Fang B (2014) Topological coding and

its application in the refinement of sift. *IEEE transactions on cybernetics* 44(11): 2155–2166.

Liu W, Anguelov D, Erhan D, Szegedy C and Reed S (2016) Ssd: Single shot multibox detector. In: *European Conference on Computer Vision*. Springer.

Long J, Shelhamer E and Darrell T (2015) Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 3431–3440.

Long P, Liu W and Pan J (2016) Deep-learned collision avoidance policy for distributed multi-agent navigation. *arXiv preprint arXiv:1609.06838* .

Lowe DG (1999) Object recognition from local scale-invariant features. In: *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2. Ieee, pp. 1150–1157.

Lowe DG (2004) Distinctive image features from scale-invariant keypoints. *International journal of computer vision* 60(2): 91–110.

Lowry S, Sünderhauf N, Newman P, Leonard JJ, Cox D, Corke P and Milford MJ (2016) Visual place recognition: A survey. *IEEE Transactions on Robotics* 32(1): 1–19.

Maas AL, Hannun AY and Ng AY (2013) Rectifier nonlinearities improve neural network acoustic models. In: *in ICML Workshop on Deep Learning for Audio, Speech and Language Processing*. Citeseer.

Maturana D and Scherer S (2015a) 3d convolutional neural networks for landing zone detection from lidar. In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, pp. 3471–3478.

Maturana D and Scherer S (2015b) Voxnet: A 3d convolutional neural network for real-time object recognition. In: *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, pp. 922–928.

Miller AT and Allen PK (2004) Graspit! a versatile simulator for robotic grasping. *IEEE Robotics & Automation Magazine* 11(4): 110–122.

Mirowski P, Pascanu R, Viola F, Soyer H, Ballard A, Banino A, Denil M, Goroshin R, Sifre L, Kavukcuoglu K et al. (2016) Learning to navigate in complex environments. *arXiv preprint arXiv:1611.03673* .

Mnih V, Badia AP, Mirza M, Graves A, Lillicrap TP, Harley T, Silver D and Kavukcuoglu K (2016) Asynchronous methods for deep reinforcement learning. *arXiv preprint arXiv:1602.01783* .

Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D and Riedmiller M (2013) Playing atari with deep reinforcement learning. In: *NIPS Deep Learning Workshop*.

Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G et al. (2015) Human-level control through deep reinforcement learning. *Nature* 518(7540): 529–533.

Moreno FA, Blanco JL and Gonzalez-Jimenez J (2015) A constant-time slam back-end in the continuum between global mapping and submapping: application to visual stereo slam. *The International Journal of Robotics Research* : 1036–1056.

Mujika A (2016) Multi-task learning with deep model based

reinforcement learning. *arXiv preprint arXiv:1611.01457* .

Muller U, Ben J, Cosatto E, Flepp B and Cun YL (2005) Off-road obstacle avoidance through end-to-end learning. In: *Advances in neural information processing systems*. pp. 739–746.

Mur-Artal R, Montiel J and Tardós JD (2015) Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics* 31(5): 1147–1163.

Naseer T, Ruhnke M, Stachniss C, Spinello L and Burgard W (2015) Robust visual slam across seasons. In: *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, pp. 2529–2535.

Ng AY, Coates A, Diel M, Ganapathi V, Schulte J, Tse B, Berger E and Liang E (2006) Autonomous inverted helicopter flight via reinforcement learning. In: *Experimental Robotics IX*. Springer, pp. 363–372.

Ng AY, Russell SJ et al. (2000) Algorithms for inverse reinforcement learning. In: *Icml*. pp. 663–670.

Oliveira GL, Burgard W and Brox T (2016) Efficient deep models for monocular road segmentation. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 4885–4891.

Ondruska P and Posner I (2016) Deep tracking: Seeing beyond seeing using recurrent neural networks. *arXiv preprint arXiv:1602.00991* .

Ou Y, Kim DH, Kim P, Kim MJ and Julius AA (2013) Motion control of magnetized tetrahymena pyriformis cells by a magnetic field with model predictive control. *The International Journal of Robotics Research* 32(1): 129–140.

Pelossof R, Miller A, Allen P and Jebara T (2004) An svm learning approach to robotic grasping. In: *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 4. IEEE, pp. 3512–3518.

Pfeiffer M, Schaeuble M, Nieto J, Siegwart R and Cadena C (2016) From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots. *arXiv preprint arXiv:1609.07910* .

Pineda FJ (1987) Generalization of back-propagation to recurrent neural networks. *Physical review letters* 59(19): 2229.

Redmon J and Angelova A (2015) Real-time grasp detection using convolutional neural networks. In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, pp. 1316–1322.

Redmon J, Divvala S, Girshick R and Farhadi A (2016) You only look once: Unified, real-time object detection. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 779–788.

Ren S, He K, Girshick R and Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In: *Advances in Neural Information Processing Systems*. pp. 91–99.

Rublee E, Rabaud V, Konolige K and Bradski G (2011) Orb: An efficient alternative to sift or surf. In: *2011 International conference on computer vision*. IEEE, pp. 2564–2571.

Ruf A and Horaud R (1999) Visual servoing of robot manipulators part i: Projective kinematics. *The International Journal of Robotics Research* 18(11): 1101–1118.

Russell SJ, Norvig P, Canny JF, Malik JM and Edwards DD (2003) *Artificial intelligence: a modern approach*, volume 2. Prentice hall Upper Saddle River.

Sadeghi F and Levine S (2016) (cad)rl: Real single-image flight without a single real image. *arXiv preprint arXiv:1611.04201* .

Schmitz A, Bansho Y, Noda K, Iwata H, Ogata T and Sugano S (2014) Tactile object recognition using deep learning and dropout. In: *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, pp. 1044–1050.

Sermanet P, Hadsell R, Scoffier M, Grimes M, Ben J, Erkan A, Crudele C, Miller U and LeCun Y (2009) A multirange architecture for collision-free off-road robot navigation. *Journal of Field Robotics* 26(1): 52–87.

Sharif Razavian A, Azizpour H, Sullivan J and Carlsson S (2014) Cnn features off-the-shelf: An astounding baseline for recognition. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M et al. (2016) Mastering the game of go with deep neural networks and tree search. *Nature* 529(7587): 484–489.

Simonyan K and Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. *Proceedings of the International Conference on Learning Representations (ICLR)* .

Smola AJ and Schölkopf B (1998) *Learning with kernels*. Citeseer.

Srivastava N, Hinton GE, Krizhevsky A, Sutskever I and Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15(1): 1929–1958.

Sünderhauf N, Shirazi S, Dayoub F, Upcroft B and Milford M (2015a) On the performance of convnet features for place recognition. In: *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, pp. 4297–4304.

Sünderhauf N, Shirazi S, Jacobson A, Dayoub F, Pepperell E, Upcroft B and Milford M (2015b) Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. *Proceedings of Robotics: Science and Systems XII* .

Sutton RS and Barto AG (1998) *Reinforcement learning: An introduction*. MIT press.

Tai L, Li S and Liu M (2016) A deep-network solution towards model-less obstacle avoidance. In: *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, pp. 2759–2764.

Tai L and Liu M (2016a) A robot exploration strategy based on q-learning network. In: *Real-time Computing and Robotics (RCAR), 2016 IEEE International Conference on*. pp. 57–62.

Tai L and Liu M (2016b) A technical report on deep visual homing through omnidirectional camera. Technical report, Department of Mechanical and Biomedical Engineering, City University of Hong Kong, Kowloon Tong.

Tai L and Liu M (2016c) Towards cognitive exploration

through deep reinforcement learning for mobile robots. *arXiv preprint arXiv:1610.01733* .

Valada A, Spinello L, Burgard W, Agarwal P, Burgard W, Spinello L, Eitel A, Springerberg J, Spinello L, Riedmiller M et al. (2015) Deep feature learning for acoustics-based terrain classification. In: *Proc. of the International Symposium on Robotics Research (ISRR)*. Springer, pp. 1383–1432.

Varshavskaya P, Kaelbling LP and Rus D (2008) Automated design of adaptive controllers for modular robots using reinforcement learning. *The International Journal of Robotics Research* 27(3-4): 505–526.

Wang DZ and Posner I (2015) Voting for voting in online point cloud object detection. *Proceedings of the Robotics: Science and Systems, Rome, Italy* 1317.

Wang Z, de Freitas N and Lanctot M (2016) Dueling network architectures for deep reinforcement learning. In: *Proceedings of The 33rd International Conference on Machine Learning*.

Wulfmeier M, Ondruska P and Posner I (2015) Deep inverse reinforcement learning. *arXiv preprint arXiv:1507.04888* .

Wulfmeier M, Wang DZ and Posner I (2016) Watch this: Scalable cost-function learning for path planning in urban environments. In: *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, pp. 2089–2095.

Yang S, Maturana D and Scherer S (2016a) Real-time 3d scene layout from a single image using convolutional neural networks. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. pp. 2183–2189.

Yang S, Song Y, Kaess M and Scherer S (2016b) Pop-up slam: Semantic monocular plane slam for low-texture environments. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pp. 1222–1229.

Zhang F, Leitner J, Milford M, Upcroft B and Corke P (2015) Towards vision-based deep reinforcement learning for robotic motion control. In: *in Proceedings of the Australasian Conference on Robotics and Automation (ACRA)*.

Zhang F, Leitner J, Upcroft B and Corke P (2016a) Vision-based reaching using modular deep networks: from simulation to the real world. *arXiv preprint arXiv:1610.06781* .

Zhang LE, Ciocarlie M and Hsiao K (2011) Grasp evaluation with graspable feature matching. In: *RSS Workshop on Mobile Manipulation: Learning to Manipulate*.

Zhang M, McCarthy Z, Finn C, Levine S and Abbeel P (2016b) Learning deep neural network policies with continuous memory states. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 520–527.

Zhang T, Kahn G, Levine S and Abbeel P (2016c) Learning deep control policies for autonomous aerial vehicles with mpc-guided policy search. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. pp. 528–535.

Zheng S, Jayasumana S, Romera-Paredes B, Vineet V, Su Z, Du D, Huang C and Torr PH (2015) Conditional random fields as recurrent neural networks. In: *Proceedings of the IEEE International Conference on Computer Vision*. pp. 1529–1537.

Zhou B, Lapedriza A, Xiao J, Torralba A and Oliva A (2014) Learning deep features for scene recognition using places database. In: *Advances in neural information processing systems*. pp. 487–495.

Zhu Y, Mottaghi R, Kolve E, Lim JJ, Gupta A, Fei-Fei L and Farhadi A (2016) Target-driven visual navigation in indoor scenes using deep reinforcement learning. *arXiv preprint arXiv:1609.05143* .

Zitnick CL and Dollár P (2014) Edge boxes: Locating object proposals from edges. In: *Computer Vision–ECCV 2014*. Springer, pp. 391–405.