Deep Reinforcement Learning in Computer Vision: A Comprehensive Survey

Ngan Le** Vidhiwar Singh Rathour* Kashu Yamazaki*
Khoa Luu Marios Savvides

April 14, 2022

Abstract

Deep reinforcement learning augments the reinforcement learning framework and utilizes the powerful representation of deep neural networks. Recent works have demonstrated the remarkable successes of deep reinforcement learning in various domains including finance, medicine, healthcare, video games, robotics, and computer vision. In this work, we provide a detailed review of recent and state-of-the-art research advances of deep reinforcement learning in computer vision. We start with *comprehending the theories* of deep learning, reinforcement learning, and deep reinforcement learning. We then *propose a categorization* of deep reinforcement learning methodologies and *discuss their advantages and limitations*. In particular, we divide deep reinforcement learning into *seven main categories* according to their applications in computer vision, i.e. (i) landmark localization (ii) object detection; (iii) object tracking; (iv) registration on both 2D image and 3D image volumetric data (v) image segmentation; (vi) videos analysis; and (vii) other applications. Each of these categories is further analyzed with reinforcement learning techniques, network design, and performance. Moreover, we provide a comprehensive analysis of the existing publicly available datasets and examine source code availability. Finally, we present some open issues and discuss future research directions on deep reinforcement learning in computer vision.

1 Introduction

Reinforcement learning (RL) is a machine learning technique for learning a sequence of actions in an interactive environment by trial and error that maximizes the expected reward [351]. Deep Reinforcement Learning (DRL) is the combination of *Reinforcement Learning* and *Deep Learning* (DL) and it has become one of the most intriguing areas of artificial intelligence today. DRL can solve a wide range of complex real-world decision-making problems with human-like intelligence that were previously intractable. DRL was selected by [316], [106] as one of ten breakthrough techniques in 2013 and 2017, respectively.

The past years have witnessed the rapid development of DRL thanks to its amazing achievement in solving challenging decision-making problems in the real world. DRL has

been successfully applied into many domains including games, robotics, autonomous driving, healthcare, natural language processing, and computer vision. In contrast to supervised learning which requires large labeled training data, DRL samples training data from an environment. This opens up many machine learning applications where big labeled training data does not exist.

Far from supervised learning, DRL-based approaches focus on solving sequential decision-making problems. They aim at deciding, based on a set of experiences collected by interacting with the environment, the sequence of actions in an uncertain environment to achieve some targets. Different from supervised learning where the feedback is available after each system action, it is simply a scalar value that may be delayed in time in the DRL framework. For example, the success or failure of the entire system is reflected after a sequence of actions. Furthermore, the supervised learning model is updated based on the loss/error of the output and there is no mechanism to get the correct value when it is wrong. This is addressed by policy gradients in DRL by assigning gradients without a differentiable loss function. This aims at teaching a model to try things out randomly and learn to do correct things more.

Many survey papers in the field of DRL including [13] [97] [414] have been introduced recently. While [13] covers central algorithms in DRL, [97] provides an introduction to DRL models, algorithms, and techniques, where particular focus is the aspects related to generalization and how DRL can be used for practical applications. Recently, [414] introduces a survey, which discusses the broad applications of RL techniques in healthcare domains ranging from dynamic treatment regimes in chronic diseases and critical care, an automated medical diagnosis from both unstructured and structured clinical data, to many other control or scheduling domains that have infiltrated many aspects of a healthcare system. Different from the previous work, our survey focuses on how to implement DRL in various computer vision applications such as landmark detection, object detection, object tracking, image registration, image segmentation, and video analysis.

Our goal is to provide our readers good knowledge about the principle of RL/DRL and thorough coverage of the latest examples of how DRL is used for solving computer vision tasks. We structure the rest of the paper as follows: we first introduce fundamentals of Deep Learning (DL) in section 2 including Multi-Layer Perceptron (MLP), Autoencoder, Deep Belief Network, Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs). Then, we present the theories of RL in section 3, which starts with the Markov Decision Process (MDP) and continues with value function and Q-function. In the end of section 3, we introduce various techniques in RL under two categories of model-based and model-free RL. Next, we introduce DRL in section 4 with main techniques in both value-based methods, policy gradient methods, and actor-critic methods under model-based and model-free categories. The application of DRL in computer vision will then be introduced in sections 5, 6, 7, 8, 9, 10, 11 corresponding respectively to DRL in landmark detection, DRL in object detection, DRL in object tracking, DRL in image registration, DRL in image segmentation, DRL in video analysis and other applications of DRL. Each application category first starts with a problem introduction and then state-of-the-art approaches in the field are discussed and compared through a summary table. We are going to discuss some

future perspectives in section 12 including challenges of DRL in computer vision and the recent advanced techniques.

2 Introduction to Deep Learning

2.1 Multi-Layer Perceptron (MLP)

Deep learning models, in simple words, are large and deep artificial neural networks. Let us consider the simplest possible neural network which is called "**neuron**" as illustrated in Fig. 1. A computational model of a single neuron is called a perceptron which consists of one or more inputs, a processo

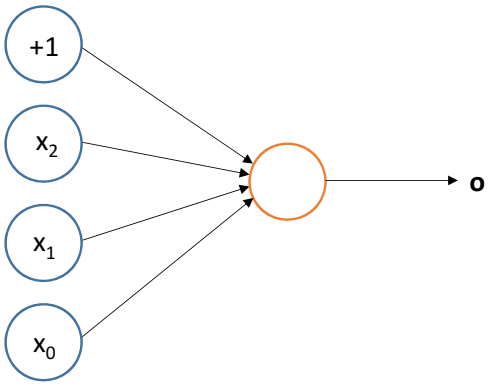

Figure 1: An example of one neuron which takes input $\mathbf{x} = [x_1, x_2, x_3]$, the intercept term $+1$ as bias, and the output $\mathbf{o}$.

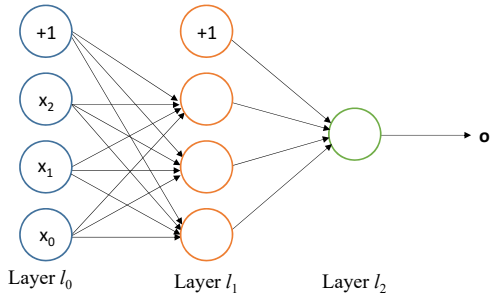

Figure 2: An example of multi-layer perceptron network (MLP)

In this example, the neuron is a computational unit that takes $\mathbf{x} = [x_0, x_1, x_2]$ as input, the intercept term $+1$ as bias $\mathbf{b}$, and the output $\mathbf{o}$. The goal of this simple network is to learn a function $f : \mathrm{R}^{\mathrm{N}} \rightarrow \mathrm{R}^{\mathrm{M}}$ where N is the number of dimensions for input $\mathbf{x}$ and M is the number of dimensions for output which is computed as $\mathbf{o} = f(\mathbf{x}, \theta)$, where θ is a set of

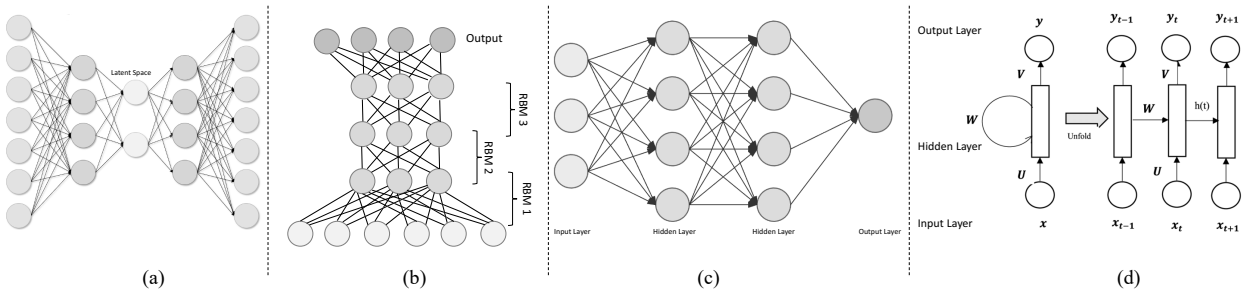

Figure 3: An illustration of various DL architectures. (a): Autoencoder (AE); (b): Deep Belief Network; (c): Convolutional Neural Network (CNN); (d): Recurrent Neural Network (RNN).

weights and are known as weights $\theta = \{w_i\}$. Mathematically, the output $\mathbf{o}$ of a one neuron is defined as:

$$\mathbf{o} = f(\mathbf{x}, \theta) = \sigma\left(\sum_{i=1}^{N} w_i x_i + b\right) = \sigma(\mathbf{W}^T \mathbf{x} + b) \tag{1}$$

In this equation, σ is the point-wise non-linear activation function. The common non-linear activation functions for hidden units are hyperbolic tangent ($Tanh$), sigmoid, softmax, ReLU, and LeakyReLU. A typical multi-layer perception (MLP) neural network is composed of one input layer, one output layer, and many hidden layers. Each layer may contain many units. In this network, $\mathbf{x}$ is the input layer, $\mathbf{o}$ is the output layer. The middle layer is called the hidden layer. In Fig. 2(b), MLP contains 3 units of the input layer, 3 units of the hidden layer, and 1 unit of the output layer.

In general, we consider a MLP neural network with L hidden layers of units, one layer of input units and one layer of output units. The number of input units is N, output units is M, and units in hidden layer l^{th} is N^l. The weight of the j^{th} unit in layer l^{th} and the i^{th} unit in layer $(l+1)^{th}$ is denoted by w_{ij}^l. The activation of the i^{th} unit in layer l^{th} is $\mathbf{h}_i^l$.

2.2 Autoencoder

Autoencoder is an unsupervised algorithm used for representation learning, such as feature selection or dimension reduction. A gentle introduction to Variational Autoencoder (VAE) is given in [11] and VAE framework is illustrated in Fig.3(a). In general, VAE aims to learn a parametric latent variable model by maximizing the marginal log-likelihood of the training data.

2.3 Deep Belief Network

Deep Belief Network (DBN) and Deep Autoencoder are two common unsupervised approaches that have been used to initialize the network instead of random initialization.

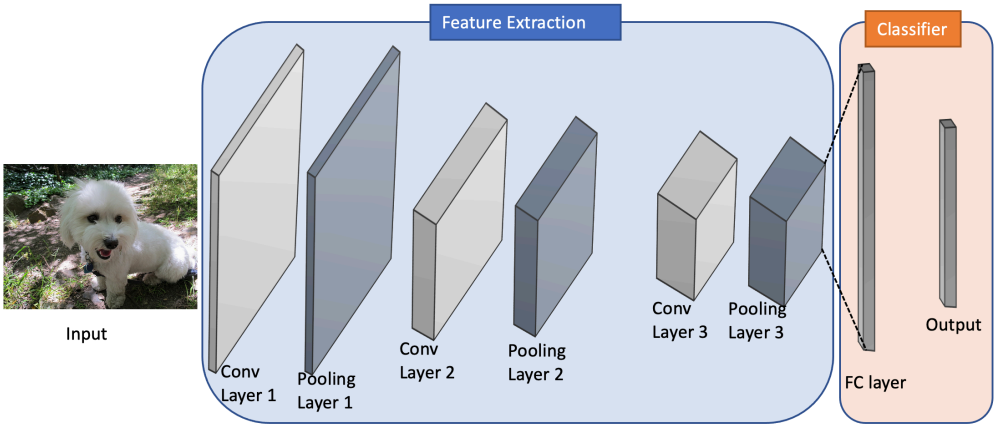

Figure 4: Architecture of a typical convolutional network for image classification containing three basic layers: convolution layer, pooling layer and fully connected layer

While Deep Autoencoder is based on Autoencoder, Deep Belief Networks is based on Restricted Boltzmann Machine (RBM), which contains a layer of input data and a layer of hidden units that learn to represent features that capture high-order correlations in the data as illustrated in Fig.3(b).

2.4 Convolutional Neural Networks (CNN)

Convolutional Neural Network (CNN) [204] [203] is a special case of fully connected MLP that implements weight sharing for processing data. CNN uses the spatial correlation of the signal to utilize the architecture in a more sensible way. Their architecture, somewhat inspired by the biological visual system, possesses two key properties that make them extremely useful for image applications: spatially shared weights and spatial pooling. These kinds of networks learn features that are shift-invariant, i.e., filters that are useful across the entire image (due to the fact that image statistics are stationary). The pooling layers are responsible for reducing the sensitivity of the output to slight input shifts and distortions, and increasing the reception field for next layers. Since 2012, one of the most notable results in Deep Learning is the use of CNN to obtain a remarkable improvement in object recognition in ImageNet classification challenge [72] [187].

A typical CNN is composed of multiple stages, as shown in Fig. 3(c). The output of each stage is made of a set of 2D arrays called feature maps. Each feature map is the outcome of one convolutional (and an optional pooling) filter applied over the full image. A point-wise non-linear activation function is applied after each convolution. In its more general form, a CNN can be written as

$$\begin{aligned} \mathbf{h}^0 =& \mathbf{x} \\ \mathbf{h}^l =& pool^l(\sigma_l(\mathbf{w}^l\mathbf{h}^{l-1} + \mathbf{b}^l)), \forall l \in 1, 2, ...L \\ \mathbf{o} =& \mathbf{h}^L \end{aligned} \tag{2}$$

where $\mathbf{w}^l, \mathbf{b}^l$ are trainable parameters as in MLPs at layer l^{th}. $\mathbf{x} \in \mathrm{R}^{c \times h \times w}$ is vectorized from an input image with c being the color channels, h the image height and w the image width. $\mathbf{o} \in \mathrm{R}^{n \times h' \times w'}$ is vectorized from an array of dimension $h' \times w'$ of output vector (of dimension n). $pool^l$ is a (optional) pooling function at layer l^{th}.

Compared to traditional machine learning methods, CNN has achieved state-of-the-art performance in many domains including image understanding, video analysis and audio/speech recognition. In *image understanding* [404], [426], CNN outperforms human capacities [39]. *Video analysis* [422], [217] is another application that turns the CNN model from a detector [374] into a tracker [94]. As a special case of *image segmentation* [194], [193], *saliency detection* is another computer vision application that uses CNN [381], [213]. In addition to the previous applications, *pose estimation* [290], [362] is another interesting research that uses CNN to estimate human-body pose. *Action recognition* in both still images and videos is a special case of recognition and is a challenging problem. [110] utilizes CNN-based representation of contextual information in which the most representative secondary region within a large number of object proposal regions, together with the contextual features, is used to describe the primary region. CNN-based action recognition in video sequences is reviewed in [420]. *Text detection and recognition* using CNN is the next step of optical character recognition (OCR) [406] and word spotting [160]. Not only in computer vision, CNN has been successfully applied into other domains such as *speech recognition and speech synthesis* [274], [283], biometrics [242], [85], [281], [350],[304], [261], biomedical [191], [342], [192], [411].

2.5 Recurrent Neural Networks (RNN)

RNN is an extremely powerful sequence model and was introduced in the early 1990s [172]. A typical RNN contains three parts, namely, sequential input data ($\mathbf{x}_t$), hidden state ($\mathbf{h}_t$) and sequential output data ($\mathbf{y}_t$) as shown in Fig. 3(d).

RNN makes use of sequential information and performs the same task for every element of a sequence where the output is dependent on the previous computations. The activation of the hidden states at time-step t is computed as a function f of the current input symbol $\mathbf{x_t}$ and the previous hidden states $\mathbf{h_{t-1}}$. The output at time t is calculated as a function g of the current hidden state $\mathbf{h_t}$ as follows

$$\begin{aligned} \mathbf{h}_t &= f(\mathbf{U}\mathbf{x}_t + \mathbf{W}\mathbf{h}_{t-1}) \\ \mathbf{y}_t &= g(\mathbf{V}\mathbf{h}_t) \end{aligned} \tag{3}$$

where $\mathbf{U}$ is the input-to-hidden weight matrix, $\mathbf{W}$ is the state-to-state recurrent weight matrix, $\mathbf{V}$ is the hidden-to-output weight matrix. f is usually a logistic sigmoid function or a hyperbolic tangent function and g is defined as a softmax function.

Most works on RNN have made use of the method of backpropagation through time (BPTT) [318] to train the parameter set $(\mathbf{U}, \mathbf{V}, \mathbf{W})$ and propagate error backward through time. In classic backpropagation, the error or loss function is defined as

$$E(\mathbf{y'}, \mathbf{y}) = \sum_t ||\mathbf{y'}_t - \mathbf{y}_t||^2 \tag{4}$$

where $\mathbf{y}_t$ is the prediction and $\mathbf{y'}_t$ is the labeled groundtruth.

For a specific weight $\mathbf{W}$, the update rule for gradient descent is defined as $\mathbf{W}^{new} = \mathbf{W} - \gamma \frac{\partial E}{\partial \mathbf{W}}$, where γ is the learning rate. In RNN model, the gradients of the error with respect to our parameters $\mathbf{U}$, $\mathbf{V}$ and $\mathbf{W}$ are learned using Stochastic Gradient Descent (SGD) and chain rule of differentiation.

The difficulty of training RNN to capture long-term dependencies has been studied in [26]. To address the issue of learning long-term dependencies, Hochreiter and Schmidhuber [139] proposed Long Short-Term Memory (LSTM), which can maintain a separate memory cell inside it that updates and exposes its content only when deemed necessary. Recently, a Gated Recurrent Unit (GRU) was proposed by [51] to make each recurrent unit adaptively capture dependencies of different time scales. Like the LSTM unit, the GRU has gating units that modulate the flow of information inside the unit but without having separate memory cells.

Several variants of RNN have been later introduced and successfully applied to wide variety of tasks, such as natural language processing [257], [214], speech recognition [115], [54], machine translation [175], [241], question answering [138], image captioning [247], [78], and many more.

3 Basics of Reinforcement Learning

This section serves as a brief introduction to the theoretical models and techniques in RL. In order to provide a quick overview of what constitutes the main components of RL methods, some fundamental concepts and major theoretical problems are also clarified. RL is a kind of machine learning method where agents learn the optimal policy by trial and error. Unlike supervised learning, the feedback is available after each system action, it is simply a scalar value that may be delayed in time in RL framework, for example, the success or failure of the entire system is reflected after a sequence of actions. Furthermore, the supervised learning model is updated based on the loss/error of the output and there is no mechanism to get the correct value when it is wrong. This is addressed by policy gradients in RL by assigning gradients without a differentiable loss function which aims at teaching a model to try things out randomly and learn to do correct things more.

Inspired by behavioral psychology, RL was proposed to address the sequential decision-making problems which exist in many applications such as games, robotics, healthcare, smart grids, stock, autonomous driving, etc. Unlike supervised learning where the data is given, an artificial agent collects experiences (data) by interacting with its environment in RL framework. Such experience is then gathered to optimize the cumulative rewards/utilities.

In this section, we focus on how the RL problem can be formalized as an agent that can make decisions in an environment to optimize some objectives presented under reward functions. Some key aspects of RL are: (i) Address the sequential decision making; (ii) There is no supervisor, only a reward presented as scalar number; and (iii) The feedback is highly delayed. Markov Decision Process (MDP) is a framework that has commonly been used to solve most RL problems with discrete actions, thus we will first discuss MDP in this section.

We then introduce value function and how to categorize RL into model-based or model-free methods. At the end of this section, we discuss some challenges in RL.

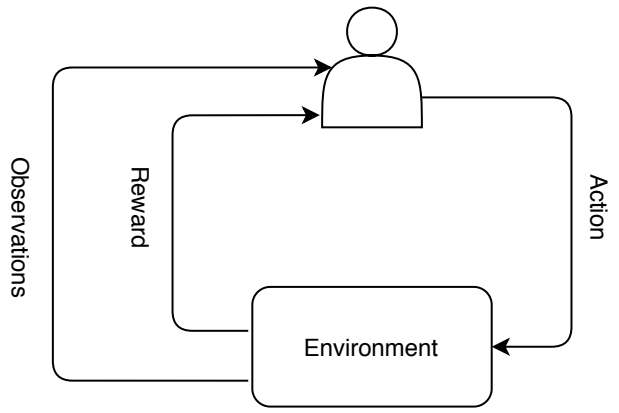

Figure 5: An illustration of agent-environment interaction in RL

3.1 Markov Decision Process

The standard theory of RL is defined by a Markov Decision Process (MDP), which is an extension of the Markov process (also known as the Markov chain). Mathematically, the Markov process is a discrete-time stochastic process whose conditional probability distribution of the future states only depends upon the present state and it provides a framework to model decision-making situations. An MDP is typically defined by five elements as follows:

- S: a set of *state* or observation space of an environment. s_0 is starting state.
- $\mathcal{A}$: set of *actions* the agent can choose.
- T: a *transition probability* function $T(s_{t+1}|s_t, a_t)$, specifying the probability that the environment will transition to state $s_{t+1} \in S$ if the agent takes action $a_t \in \mathcal{A}$ in state $s_t \in S$.
- R: a *reward* function where $r_{t+1} = R(s_t, s_{t+1})$ is a reward received for taking action a_t at state s_t and transfer to the next state s_{t+1}.
- γ: a discount factor.

Considering MDP(S, $\mathcal{A}$, γ, T, R), the agent chooses an action a_t according to the policy $\pi(a_t|s_t)$ at state s_t. Notably, agent's algorithm for choosing action a given current state s, which in general can be viewed as distribution $\pi(a|s)$, is called a *policy* (strategy). The environment receives the action, produces a reward r_{t+1} and transfers to the next state s_{t+1} according to the transition probability $T(s_{t+1}|s_t, a_t)$. The process continues until the agent reaches a terminal state or a maximum time step. In RL framework, the tuple $(s_t, a_t, r_{t+1}, s_{t+1})$ is called *transition*. Several sequential transitions are usually referred to as

roll-out. Full sequence $(s_0, a_0, r_1, s_1, a_1, r_2, ...)$ is called a *trajectory*. Theoretically, trajectory is infinitely long, but the episodic property holds in most practical cases. One trajectory of some finite length τ is called an *episode*. For given MDP and policy π, the probability of observing $(s_0, a_0, r_1, s_1, a_1, r_2, ...)$ is called *trajectory distribution* and is denoted as:

$$\mathcal{T}_\pi = \prod_t \pi(a_t|s_t)T(s_{t+1}|s_t, a_t) \tag{5}$$

The objective of RL is to find the *optimal policy* π^* for the agent that maximizes the cumulative reward, which is called *return*. For every episode, the return is defined as the weighted sum of immediate rewards:

$$\mathcal{R} = \sum_{t=0}^{\tau-1} \gamma^t r_{t+1} \tag{6}$$

Because the policy induces a trajectory distribution, the *expected reward* maximization can be written as:

$$\mathbb{E}_{\mathcal{T}_\pi} \sum_{t=0}^{\tau-1} r_{t+1} \to \max_\pi \tag{7}$$

Thus, given MDP and policy π, the *discounted expected reward* is defined:

$$\mathcal{G}(\pi) = \mathbb{E}_{\mathcal{T}_\pi} \sum_{t=0}^{\tau-1} \gamma^t r_{t+1} \tag{8}$$

The goal of RL is to find an *optimal policy* π^*, which maximizes the discounted expected reward, i.e. $\mathcal{G}(\pi) \to \max_\pi$.

3.2 Value and Q- functions

The value function is applied to evaluate how good it is for an agent to utilize policy π to visit state s. The concept of "good" is defined in terms of expected return, i.e. future rewards that can be expected to receive in the future and it depends on what actions it will take. Mathematically, the value is the expectation of return, and value approximation is obtained by Bellman expectation equation as follows:

$$V^\pi(s_t) = \mathbb{E}[r_{t+1} + \gamma V^\pi(s_{t+1})] \tag{9}$$

$V^\pi(s_t)$ is also known as state-value function, and the expectation term can be expanded as a product of policy, transition probability, and return as follows:

$$V^\pi(s_t) = \sum_{a_t \in \mathcal{A}} \pi(a_t|s_t) \sum_{s_{t+1} \in S} T(s_{t+1}|s_t, a_t)[R(s_t, s_{t+1}) + \gamma V^\pi(s_{t+1})] \tag{10}$$

This equation is called the Bellman equation. When the agent always selects the action according to the optimal policy π^* that maximizes the value, the Bellman equation can be

expressed as follows:

$$\begin{aligned} V^*(s_t) &= \max_{a_t} \sum_{s_{t+1} \in S} T(s_{t+1}|s_t, a_t)[R(s_t, s_{t+1}) + \gamma V^*(s_{t+1})] \\ &\triangleq \max_{a_t} Q^*(s_t, a_t) \end{aligned} \tag{11}$$

However, obtaining optimal value function V^* does not provide enough information to reconstruct some optimal policy π^* because the real-world environment is complicated. Thus, a quality function (Q-function) is also called the action-value function under policy π. The Q-function is used to estimate how good it is for an agent to perform a particular action (a_t) in a state (s_t) with a policy π and it is introduced as:

$$Q^\pi(s_t, a_t) = \sum_{s_{t+1}} T(s_{t+1}|s_t, a_t)[R(s_t, s_{t+1}) + \gamma V^\pi(s_{t+1})] \tag{12}$$

Unlike value function which specifies the goodness of a state, a Q-function specifies the goodness of action in a state.

3.3 Category

In general, RL can be divided into either model-free or model-based methods. Here, "model" is defined by the two quantity: transition probability function $T(s_{t+1}|s_t, a_t)$ and the reward function $R(s_t, s_{t+1})$.

3.3.1 Model-based RL

Model-based RL is an approach that uses a learnt model, i.e. $T(s_{t+1}|s_t, a_t)$ and reward function $R(s_t, s_{t+1})$ to predict the future action. There are four main model-based techniques as follows:

- **Value Function:** The objective of value function methods is to obtain the best policy by maximizing the value functions in each state. A value function of a RL problem can be defined as in Eq.10 and the optimal state-value function is given in Eq.11 which are known as Bellman equations. Some common approaches in this group are Differential Dynamic Programming [208], [266], Temporal Difference Learning [249], Policy Iteration [334] and Monte Carlo [137].

- **Transition Models:** Transition models decide how to map from a state s, taking action a to the next state (s') and it strongly affects the performance of model-based RL algorithms. Based on whether predicting the future state s' is based on the probability distribution of a random variable or not, there are two main approaches in this group: stochastic and deterministic. Some common methods for deterministic models are decision trees [280] and linear regression [265]. Some common methods for stochastic models are Gaussian processes [71], [1], [12], Expectation-Maximization [59] and dynamic Bayesian networks [280].

- **Policy Search:** Policy search approach directly searches for the optimal policy by modifying its parameters, whereas the value function methods indirectly find the actions that maximize the value function at each state. Some of the popular approaches in this group are: gradient-based [87], [267], information theory [1], [189] and sampling based [21].

- **Return Functions:** Return functions decide how to aggregate rewards or punishments over an episode. They affect both the convergence and the feasibility of the model. There are two main approaches in this group: discounted returns functions [21], [75], [393] and averaged returns functions [34], [3]. Between the two approaches, the former is the most popular which represents the uncertainty about future rewards. While small discount factors provide faster convergence, its solution may not be optimal.

In practice, transition and reward functions are rarely known and hard to model. The comparative performance among all model-based techniques is reported in [385] with over 18 benchmarking environments including noisy environments. The Fig.6 summarizes different model-based RL approaches.

3.3.2 Model-free methods

Learning through the experience gained from interactions with the environment, i.e. model-free method tries to estimate the t. discrete problems transition probability function and the reward function from the experience to exploit them in acquisition of policy. Policy gradient and value-based algorithms are popularly used in model-free methods.

- **The policy gradient methods**: In this approach, RL task is considered as optimization with stochastic first-order optimization. Policy gradient methods directly optimize the discounted expected reward, i.e. $\mathcal{G}(\pi) \rightarrow \max_{\pi}$ to obtains the optimal policy π^* without any additional information about MDP. To do so, approximate estimations of the gradient with respect to policy parameters are used. Take [392] as an example, policy gradient parameterizes the policy and updates parameters θ,

$$\mathcal{G}^{\theta}(\pi) = \mathbb{E}_{\mathcal{T}_{\phi}} \sum_{t=0} log(\pi_{\theta}(a_t|s_t))\gamma^t \mathcal{R} \tag{13}$$

  where $\mathcal{R}$ is the total accumulated return and defined in Eq. 6. Common used policies are Gibbs policies [20], [352] and Gaussian policies [294]. Gibbs policies are used in discrete problems whereas Gaussian policies are used in continuous problems.

- **Value-based methods**: In this approach, the optimal policy π^* is implicitly conducted by gaining an approximation of optimal Q-function $Q^*(s, a)$. In value-based methods, agents update the value function to learn suitable policy while policy-based RL agents learn the policy directly. To do that, Q-learning is a typical value-based method. The update rule of Q-learning with learning rate λ is defined as:

$$Q(s_t, a_t) = Q(s_t, a_t) + \lambda \delta_t \tag{14}$$

Table 1: Comparison between model-based RL and model-free RL

Factors	**Model-based RL**	**Model-free RL**
Number of iterations between agent and environment	Small	Big
Convergence	Fast	Slow
Prior knowledge of transitions	Yes	No
Flexibility	Strongly depend on a learnt model	Adjust based on trials and errors

where $\delta_t = R(s_t, s_{t+1}) + \gamma \arg\max_a Q(s_{t+1}, a) - Q(s_t, a)$ is the temporal difference (TD) error.

Target at self-play Chess, [394] investigates inasmuch it is possible to leverage the qualitative feedback for learning an evaluation function for the game. [319] provides the comparison of learning of linear evaluation functions between using preference learning and using least-squares temporal difference learning, from samples of game trajectories. The value-based methods depend on a specific, optimal policy, thus it is hard for transfer learning.

- **Actor-critic** is an improvement of policy gradient with an value-based critic Γ, thus, Eq.13 is rewritten as:

$$\mathcal{G}^{\theta}(\pi) = \mathbb{E}_{\mathcal{T}_{\phi}} \sum_{t=0} log(\pi_{\theta}(a_t|s_t))\gamma^t \Gamma_t \tag{15}$$

The critic function Γ can be defined as $Q^{\pi}(s_t, a_t)$ or $Q^{\pi}(s_t, a_t) - V_t^{\pi}$ or $R[s_{t-1}, s_t] + V_{t+1}^{\pi} - V_t^{\pi}$

Actor-critic methods are combinations of actor-only methods and critic-only methods. Thus, actor-critic methods have been commonly used RL. Depend on reward setting, there are two groups of actor-critic methods, namely discounted return [282], [30] and average return [289], [31]. The comparison between model-based and model-free methods is given in Table 1.

4 Introduction to Deep Reinforcement Learning

DRL, which was proposed as a combination of RL and DL, has achieved rapid developments, thanks to the rich context representation of DL. Under DRL, the aforementioned value and policy can be expressed by neural networks which allow dealing with a continuous state or action that was hard for a table representation. Similar to RL, DRL can be categorized into model-based algorithms and model-free algorithms which will be introduced in this section.

4.1 Model-Free Algorithms

There are two approaches, namely, Value-based DRL methods and Policy gradient DRL methods to implement model-free algorithms.

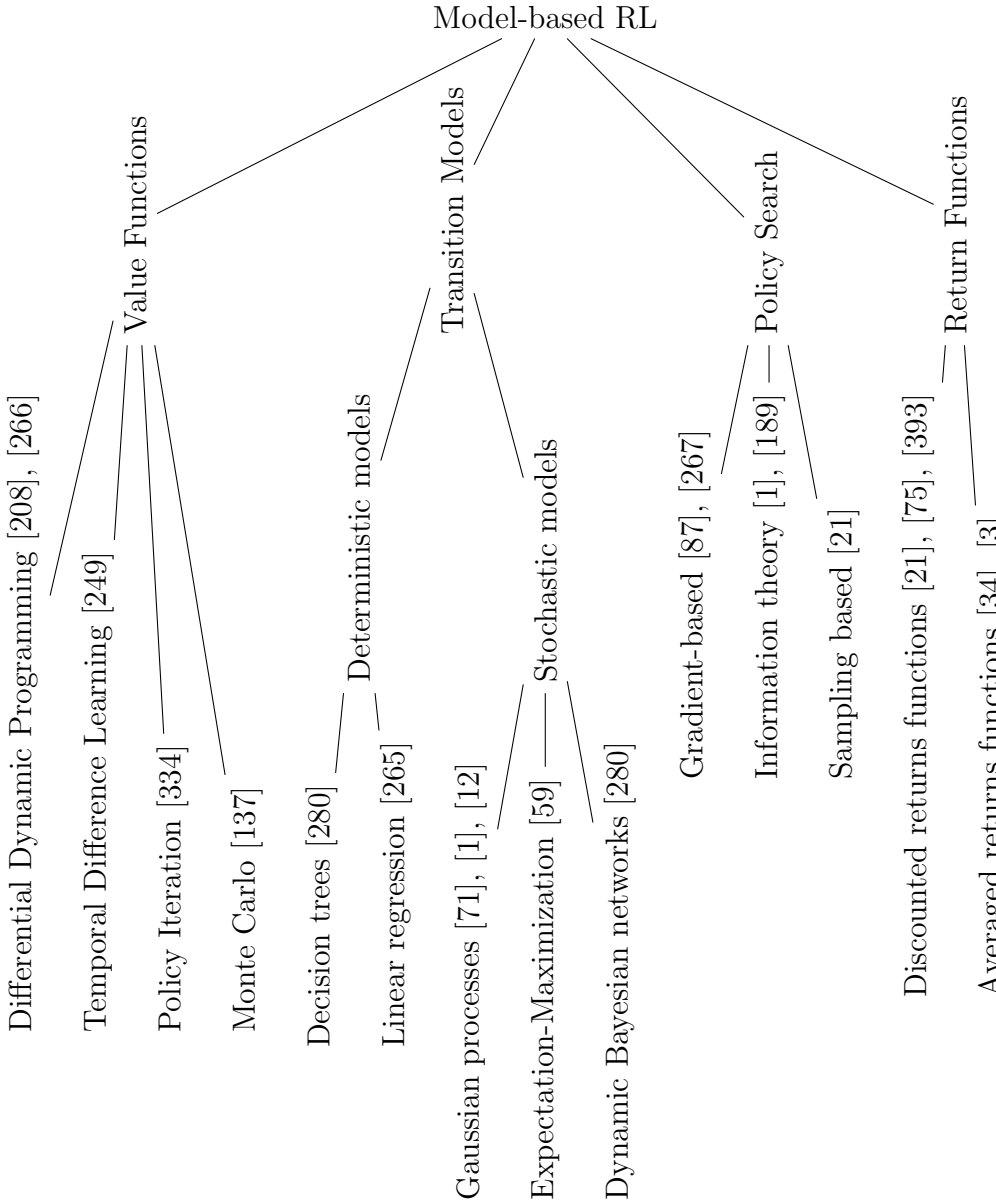

Figure 6: Summarization of model-based RL approaches

4.1.1 Value-based DRL methods

Deep Q-Learning Network (DQN): Deep Q-learning [264] (DQN) is the most famous DRL model which learns policies directly from high-dimensional inputs by CNNs. In DQN, input is raw pixels and output is a quality function to estimate future rewards as given in Fig.7. Take regression problem as an instance. Let y denote the target of our regression task, the regression with input (s, a), target $y(s, a)$ and the MSE loss function is as:

$$\begin{aligned} \mathcal{L}^{\mathcal{DQN}} &= \mathcal{L}(y(s_t, a_t), Q^*(s_t, a_t, \theta_t)) \\ &= ||y(s_t, a_t) - Q^*(s_t, a_t, \theta_t)||^2 \\ y(s_t, a_t) &= R(s_t, s_{t+1}) + \gamma \max_{a_{t+1}} Q^*(s_{t_1}, a_{t+1}, \theta_t) \end{aligned} \tag{16}$$

Where θ is vector of parameters, $\theta \in \mathbb{R}^{|S||R|}$ and s_{t+1} is a sample from $T(s_{t+1}|s_t, a_t)$ with input of (s_t, a_t).

Minimizing the loss function yields a gradient descent step formula to update θ as follows:

$$\theta_{t+1} = \theta_t - \alpha_t \frac{\partial \mathcal{L}^{\mathcal{DQN}}}{\partial \theta} \tag{17}$$

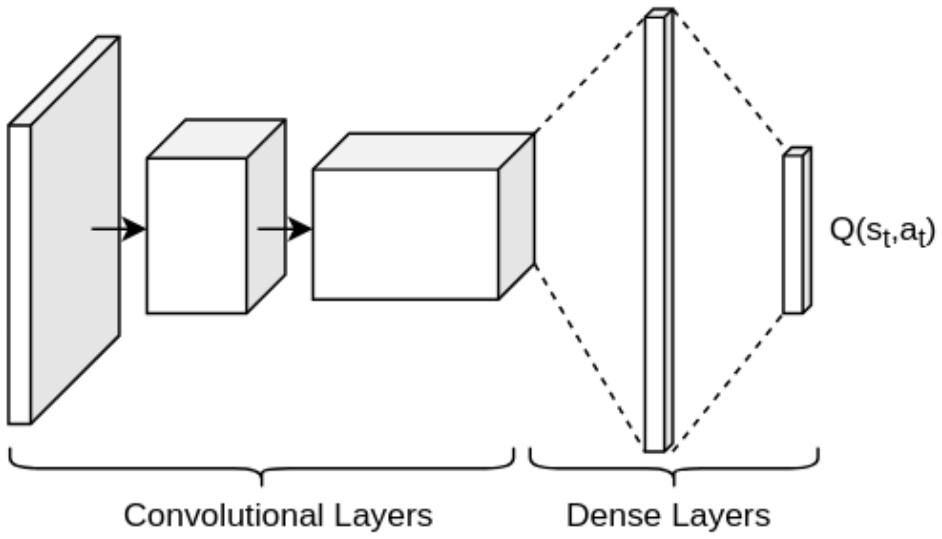

Figure 7: Network structure of Deep Q-Network (DQN), where Q-values Q(s,a) are generated for all actions for a given state.

Double DQN: In DQN, the values of Q^* in many domains were leading to overestimation because of max. In Eq.16, $y(s, a) = R(s, s') + \gamma \max_{a'} Q^*(s', a', \theta)$ shifts Q-value estimation towards either to the actions with high reward or to the actions with overestimating approximation error. Double DQN [370] is an improvement of DQN that combines double Q-learning [130] with DQN and it aims at reducing observed overestimation with better

performance. The idea of Double DQN is based on separating action selection and action evaluation using its own approximation of Q^* as follows:

$$\max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1}; \theta) = Q^*(s_{t+1}, \arg\max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1}; \theta_1); \theta_2) \tag{18}$$

Thus

$$y = R(s_t, s_{t+1}) + \gamma Q^*(s_{t+1}, \arg\max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1}; \theta_1); \theta_2) \tag{19}$$

The easiest and most expensive implementation of double DQN is to run two independent DQNs as follows:

$$\begin{aligned} y_1 = R(s_t, s_{t+1}) + \\ \gamma Q_1^*(s_{t+1}, \arg\max_{a_{t+1}} Q_2^*(s_{t+1}, a_{t+1}; \theta_2); \theta_1) \\ y_2 = R(s_t, s_{t+1}) + \\ \gamma Q_2^*(s_{t+1}, \arg\max_{a_{t+1}} Q_1^*(s_{t+1}, a_{t+1}; \theta_1); \theta_2) \end{aligned} \tag{20}$$

Dueling DQN: In DQN, when the agent visits an unfavorable state, instead of lowering its value V^*, it remembers only low pay-off by updating Q^*. In order to address this limitation, Dueling DQN [390] incorporates approximation of V^* explicitly in a computational graph by introducing an advantage function as follows:

$$A^\pi(s_t, a_t) = Q^\pi(s_t, a_t) - V^\pi(s_t) \tag{21}$$

Therefore, we can reformulate Q-value: $Q^*(s, a) = A^*(s, a) + V^*(s)$. This implies that after DL the feature map is decomposed into two parts corresponding to $V^*(v)$ and $A^*(s, a)$ as illustrated in Fig.8. This can be implemented by splitting the fully connected layers in the DQN architecture to compute the advantage and state value functions separately, then combining them back into a single Q-function. An interesting result has shown that Dueling DQN obtains better performance if it is formulated as:

$$Q^*(s_t, a_t) = V^*(s_t) + A^*(s_t, a_t) - \max_{a_{t+1}} A^*(s_t, a_{t+1}) \tag{22}$$

In practical implementation, averaging instead of maximum is used, i.e.

$$Q^*(s_t, a_t) = V^*(s_t) + A^*(s_t, a_t) - \text{mean}_{a_{t+1}} A^*(s_t, a_{t+1})$$

Furthermore, to address the limitation of memory and imperfect information at each decision point, Deep Recurrent Q-Network (DRQN) [131] employed RNNs into DQN by replacing the first fully-connected layer with an RNN. Multi-step DQN [68] is one of the most popular improvements of DQN by substituting one-step approximation with N-steps.

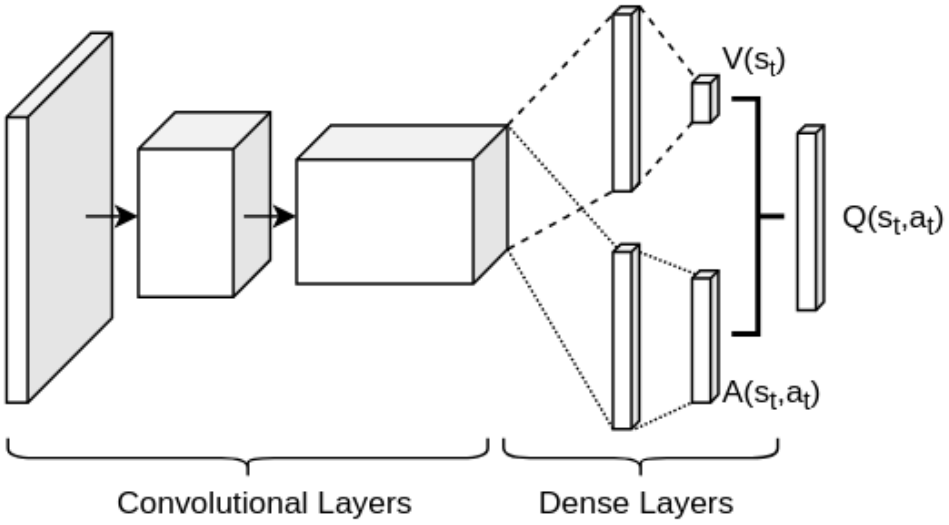

Figure 8: Network structure of Dueling DQN, where value function $V(s)$ and advantage function $A(s, a)$ are combined to predict Q-values $Q(s, a)$ for all actions for a given state.

4.1.2 Policy gradient DRL methods

Policy Gradient Theorem: Different from value-based DRL methods, policy gradient DRL optimizes the policy directly by optimizing the following objective function which is defined as a function of θ.

$$\mathcal{G}(\theta) = \mathbb{E}_{\mathcal{T}\sim\pi_\theta} \sum_{t=1} \gamma^{t-1} R(s_{t-1}, s_t) \rightarrow \max_\theta \tag{23}$$

For any MDP and differentiable policy π_θ, the gradient of objective Eq.23 is defined by policy gradient theorem [353] as follows:

$$\bigtriangledown_\theta \mathcal{G}(\theta) = \mathbb{E}_{\mathcal{T}\sim\pi_\theta} \sum_{t=0} \gamma^t Q^\pi(s_t, a_t) \bigtriangledown_\theta \log\pi_\theta(a_t|s_t) \tag{24}$$

REINFORCE: REINFORCE was introduced by [392] to approximately calculate the gradient in Eq.24 by using Monte-Carlo estimation. In REINFORCE approximate estimator, Eq.24 is reformulated as:

$$\bigtriangledown_\theta \mathcal{G}(\theta) \approx \sum_{\mathcal{T}}^{N} \sum_{t=0} \gamma^t \bigtriangledown_\theta \log\pi_\theta(a_t|s_t)(\sum_{t'=t} \gamma^{t'-t} R(s_{t'}, s_{t'+1})) \tag{25}$$

where $\mathcal{T}$ is trajectory distribution and defined in Eq.5. Theoretically, REINFORCE can be straightforwardly applied into any parametric $\pi_{theta}(a|s)$. However, it is impractical to use because of its time-consuming nature for convergence and local optimums problem. Based on the observation that the convergence rate of stochastic gradient descent directly depends

on the variance of gradient estimation, the variance reduction technique was proposed to address naive REINFORCE's limitations by adding a term that reduces the variance without affecting the expectation.

4.1.3 Actor-Critic DRL algorithm

Both value-based and policy gradient algorithms have their own pros and cons, i.e. policy gradient methods are better for continuous and stochastic environments, and have a faster convergence whereas, value-based methods are more sample efficient and steady. Lately, actor-critic [182] [262] was born to take advantage from both value-based and policy gradient while limiting their drawbacks. Actor-critic architecture computes the policy gradient using a value-based critic function to estimate expected future reward. The principal idea of actor-critic is to divide the model into two parts: (i) computing an action based on a state and (ii) producing the Q values of the action. As given in Fig.9, the actor takes as input the state s_t and outputs the best action a_t. It essentially controls how the agent behaves by learning the optimal policy (policy-based). The critic, on the other hand, evaluates the action by computing the value function (value-based). The most basic actor-critic method (beyond the tabular case) is naive policy gradients (REINFORCE). The relationship between actor-critic is similar to kid-mom. The kid (actor) explores the environment around him/her with new actions i.e. tough fire, hit a wall, climb a tree, etc while the mom (critic) watches the kid and criticizes/compliments him/her. The kid then adjusts his/her behavior based on what his/her mom told. When the kids get older, he/she can realize which action is bad/good.

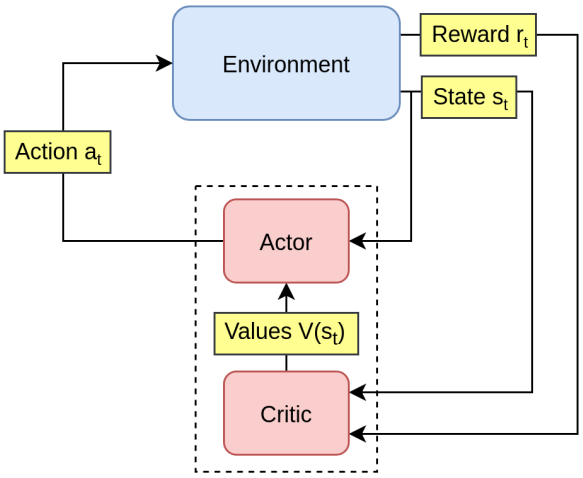

Figure 9: Flowchart showing the structure of actor critic algorithm.

Advantage Actor-Critic (A2C) Advantage Actor-Critic (A2C) [263] consist of two neural networks i.e. actor network $\pi_\theta(a_t|s_t)$ representing for policy and critic network V^π_ω with parameters ω approximately estimating actor's performance. In order to determine how much better, it is to take a specific action compared to the average, an advantage value is

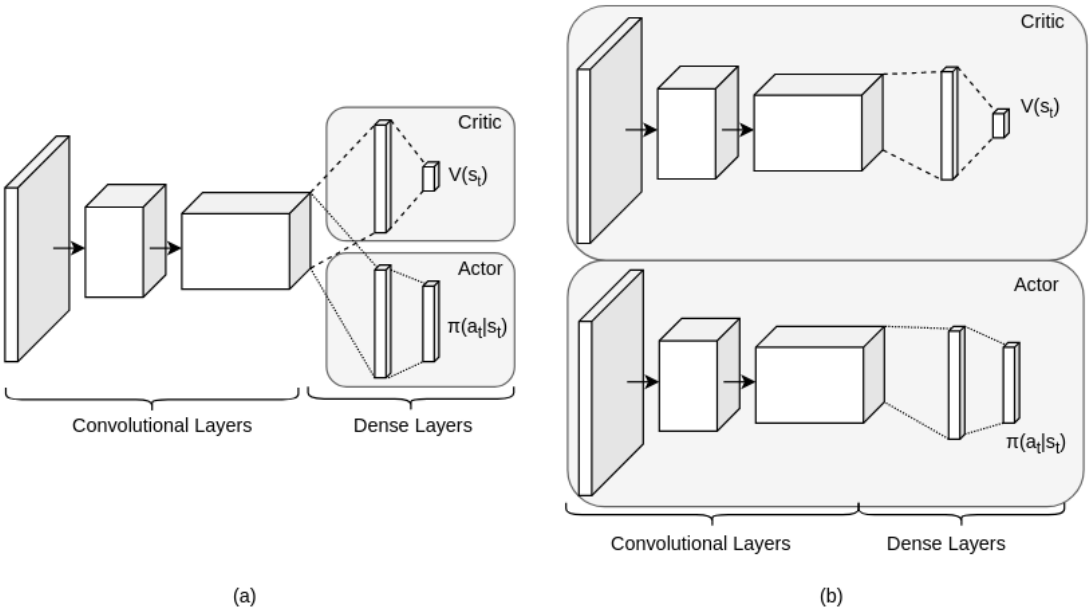

Figure 10: An illustration of Actor-Critic algorithm in two cases: sharing parameters (a) and not sharing parameters (b).

defined as:

$$A^{\pi}(s_t, a_t) = Q^{\pi}(s_t, a_t) - V^{\pi}(s_t) \tag{26}$$

Instead of constructing two neural networks for both the Q value and the V value, using the Bellman optimization equation, we can rewrite the advantage function as:

$$A^{\pi}(s_t, a_t) = R(s_t, s_{t+1}) + \gamma V^{\pi}_{\omega}(s_{t+1}) - V^{\pi}_{\omega}(s_t) \tag{27}$$

For given policy π, its value function can be obtained using point iteration for solving:

$$V^{\pi}(s_t) = \mathbb{E}_{a_t \sim \pi(a_t|s_t)} \mathbb{E}_{s_{t+1} \sim T(s_{t+1}|a_t, s_t)} (R(s_t, s_{t+1}) + \gamma V^{\pi}(s_{t+1})) \tag{28}$$

Similar to DQN, on each update a target is computed using current approximation:

$$y = R(s_t, s_{t+1}) + \gamma V^{\pi}_{\omega}(s_{t+1}) \tag{29}$$

At time step t, the A2C algorithm can be implemented as following steps:

- Step 1: Compute advantage function using Eq.27.
- Step 2: Compute target using Eq.29.
- Step 3: Compute critic loss with MSE loss: $\mathcal{L} = \frac{1}{B} \sum_T ||y - V^{\pi}(s_t))||^2$, where B is batch size and $V^{\pi}(s_t)$ is defined in Eq.28.

- Step 4: Compute critic gradient: $\bigtriangledown^{critic} = \frac{\partial \mathcal{L}}{\partial \omega}$.
- Step 5: Compute actor gradient: $\bigtriangledown^{actor} = \frac{1}{B} \sum_T \bigtriangledown_\theta \log \pi(a_t|s_t) A^\pi(s_t, a_t)$

Asynchronous Advantage Actor Critic (A3C) Besides A2C, there is another strategy to implement an Actor-Critic agent. Asynchronous Advantage Actor-Critic (A3C) [263] approach does not use experience replay because this requires a lot of memory. Instead, A3C asynchronously executes different agents in parallel on multiple instances of the environment. Each worker (copy of the network) will update the global network asynchronously. Because of the asynchronous nature of A3C, some workers (copy of the agents) will work with older values of the parameters. Thus the aggregating update will not be optimal. On the other hand, A2C synchronously updates the global network. A2C waits until all workers finished their training and calculated their gradients to average them, to update the global network. In order to update the entire network, A2C waits for each actor to finish their segment of experience before updating the global parameters. As a consequence, the training will be more cohesive and faster. Different from A3C, each worker in A2C has the same set of weights since and A2C updates all their workers at the same time. In short, A2C is an alternative to the synchronous version of the A3C. In A2C, it waits for each actor to finish its segment of experience before updating, averaging over all of the actors. In a practical experiment, this implementation is more effectively uses GPUs due to larger batch sizes. The structure of an actor-critic algorithm can be divided into two types depending on parameter sharing as illustrated in Fig.10.

In order to overcome the limitation of speed, GA3C [16] was proposed and it achieved a significant speedup compared to the original CPU implementation. To more effectively train A3C, [141] proposed FFE which forces random exploration at the right time during a training episode, that can lead to improved training performance.

4.2 Model-Based Algorithms

We have discussed so far model-free methods including the value-based approach and policy gradient approach. In this section, we focus on the model-based approach, that deals with the dynamics of the environment by learning a transition model that allows for simulation of the environment without interacting with the environment directly. In contrast to model-free approaches, model-based approaches are learned from experience by a function approximation. Theoretically, no specific prior knowledge is required in model-based RL/DRL but incorporating prior knowledge can help faster convergence and better-trained model, speed up training time as well as the number of training samples. While using raw data with pixel, it is difficult for model-based RL to work on high dimensional and dynamic environments. This is addressed in DRL by embedding the high-dimensional observations into a lower-dimensional space using autoencoders [95]. Many DRL approaches have been based on scaling up prior work in RL to high-dimensional problems. A good overview of model-based RL for high-dimensional problems can be found in [297] which partition model-based DRL into three categories: explicit planning on given transitions, explicit planning on

learned transitions, and end-to-end learning of both planning and transitions. In general, DRL targets training DNNs to approximate the optimal policy π^* together with optimal value functions V^* and Q^*. In the following, we will cover the most common model-based DRL approaches including value function and policy search methods.

4.2.1 Value function

We start this category with DQN [264] which has been successfully applied to classic Atari and illustrated in Fig.7. DQN uses CNNs to deal with high dimensional state space like pixels, to approximate the Q-value function.

Monte Carlo tree search (MCTS) MCTS [62] is one of the most popular methods to look-ahead search and it is combined with a DNN-based transition model to build a model-based DRL in [9]. In this work, the learned transition model predicts the next frame and the rewards one step ahead using the input of the last four frames of the agent's first-person-view image and the current action. This model is then used by the Monte Carlo tree search algorithm to plan the best sequence of actions for the agent to perform.

Value-Targeted Regression (UCRL-VTR) Alex, et al. proposed model-based DRL for regret minimization [167]. In their work, a set of models, that are 'consistent' with the data collected, is constructed at each episode. The consistency is defined as the total squared error, whereas the value function is determined by solving the optimistic planning problem with the constructed set of models

4.2.2 Policy search

Policy search methods aim to directly find policies by means of gradient-free or gradient-based methods.

Model-Ensemble Trust-Region Policy Optimization (ME-TRPO) ME-TRPO [190] is mainly based on Trust Region Policy Optimization (TRPO) [327] which imposes a trust region constraint on the policy to further stabilize learning.

Model-Based Meta-Policy-Optimization (MB-MPO) MB-MPO [58] addresses the performance limitation of model-based DRL compared against model-free DRL when learning dynamics models. MB-MPO learns an ensemble of dynamics models, a policy that can quickly adapt to any model in the ensemble with one policy gradient step. As a result, the learned policy exhibits less model bias without the need to behave conservatively.

A summary of both model-based and model-free DRL algorithms is given in Table 2. In this Table, we also categorized DRL techniques into either on-policy or off-policy. In on-policy RL, it allows the use of older samples (collected using the older policies) in the calculation. The policy π^k is updated with data collected by π^k itself. In off-policy RL, the data is assumed to be composed of different policies $\pi^0, \pi^0, ..., \pi^k$. Each policy has its own data collection, then the data collected from π^0, π^1, ..., π^k is used to train π^{k+1}.

Table 2: Summary of model-based and model-free DRL algorithms consisting of value-based and policy gradient methods.

DRL Algorithms	Description	Category
DQN [264]	Deep Q Network	Value-based Off-policy
Double DQN [370]	Double Deep Q Network	Value-based Off-policy
Dueling DQN [390]	Dueling Deep Q Network	Value-based Off-policy
MCTS [9]	Monte Carlo tree search	Value-based On-policy
UCRL-VTR[167]	optimistic planning problem	Value-based Off-policy
DDPG [223]	DQN with Deterministic Policy Gradient	Policy gradient Off-policy
TRPO [327]	Trust Region Policy Optimization	Policy gradient On-policy
PPO [328]	Proximal Policy Optimization	Policy gradient On-policy
ME-TRPO [190]	Model-Ensemble Trust-Region Policy Optimization	Policy gradient On-policy
MB-MPO [58]	Model-Based Meta- Policy-Optimization	Policy gradient On-policy
A3C [263]	Asynchronous Advantage Actor Critic	Actor Critic On-Policy
A2C [263]	Advantage Actor Critic	Actor Critic On-Policy

4.3 Good practices

Inspired by Deep Q-learning [264], we discuss some useful techniques that are used during training an agent in DRL framework in practices.

Experience replay Experience replay [417] is a useful part of off-policy learning and is often used while training an agent in RL framework. By getting rid of as much information as possible from past experiences, it removes the correlations in training data and reduces the oscillation of the learning procedure. As a result, it enables agents to remember and re-use past experiences sometimes in many weights updates which increases data efficiency.

Minibatch learning Minibatch learning is a common technique that is used together with experience replay. Minibatch allows learning more than one training sample at each step, thus, it makes the learning process robust to outliers and noise.

Target Q-network freezing As described in [264], two networks are used for the training process. In target Q-network freezing: one network interacts with the environment and another network plays the role of a target network. The first network is used to generate target Q-values that are used to calculate losses. The weights of the second network i.e. target network are fixed and slowly updated to the first network [224].

Reward clipping A reward is the scalar number provided by the environment and it aims at optimizing the network. To keep the rewards in a reasonable scale and to ensure proper learning, they are clipped to a specific range (-1 ,1). Here 1 refers to as positive reinforcement or reward and -1 is referred to as negative reinforcement or punishment.

Model-based v.s. model-free approach Whether the model-free or model-based approaches is chosen mainly depends on the model architecture i.e. policy and value function.

5 DRL in Landmark Detection

Autonomous landmark detection has gained more and more attention in the past few years. One of the main reasons for this increased inclination is the rise of automation for evaluating data. The motivation behind using an algorithm for landmarking instead of a person is that manual annotation is a time-consuming tedious task and is prone to errors. Many efforts have been made for the automation of this task. Most of the works that were published for this task using a machine learning algorithm to solve the problem. [64] proposed a regression forest-based method for detecting landmark in a full-body CT scan. Although the method was fast it was less accurate when dealing with large organs. [101] extended the work of [64] by adding statistical shape priors that were derived from segmentation masks with cascade regression.

In order to address the limitations of previous works on anatomy detection, [105] reformulated the detection problem as a behavior learning task for an artificial agent using MDP. By using the capabilities of DRL and scale-space theory [226], the optimal search strategies for finding anatomical structures are learned based on the image information at multiple scales. In their approach, the search starts at the coarsest scale level for capturing global context and continues to finer scales for capturing more local information. In their

RL configuration, the state of the agent at time t, $s_t = I(\vec{p}_t)$ is defined as an axis-aligned box of image intensities extracted from the image I and centered at the voxel-position $\vec{p}_t$ in image space. An action a_t allows the agent to move from any voxel position $\vec{p}_t$ to an adjacent voxel position $\vec{p}_{t+1}$. The reward function represents distance-based feedback, which is positive if the agent gets closer to the target structure and negative otherwise. In this work, a CNN is used to extract deep semantic features. The search starts with the coarsest scale level $M-1$, the algorithm tries to maximize the reward which is the change in distance between ground truth and predicted landmark location before and after the action of moving the scale window across the image. Upon convergence, the scale level is changed to $M-2$ and the search continued from the convergence point at scale level $M-1$. The process is repeated on the following scales until convergence on the finest scale. The authors performed experiments on 3D CT scans and obtained an average accuracy increase of 20-30% and lower distance error than the other techniques such as SADNN [104] and 3D-DL [427]

Focus on anatomical landmark localization in 3D fetal US images, [10] proposed and demonstrated use cases of several different Deep Q-Network RL models to train agents that can precisely localize target landmarks in medical scans. In their work, they formulate the landmark detection problem as an MDP of a goal-oriented agent, where an artificial agent is learned to make a sequence of decisions towards the target point of interest. At each time step, the agent should decide which direction it has to move to find the target landmark. These sequential actions form a learned policy forming a path between the starting point and the target landmark. This sequential decision-making process is approximated under RL. In this RL configuration, the environment is defined as a 3D input image, action A is a set of six actions $a_x+, a_x-, a_y+, a_y-, a_z+, a_z-$ corresponding to three directions, the state s is defined as a 3D region of interest (ROI) centered around the target landmark and the reward is chosen as the difference between the two Euclidean distances: the previous step and current step. This reward signifies whether the agent is moving closer to or further away from the desired target location. In this work, they also proposed a novel fixed- and multi-scale optimal path search strategy with hierarchical action steps for agent-based landmark localization frameworks.

Whereas pure policy or value-based methods have been widely used to solve RL-based localization problems, [7] adopts an actor-critic [262] based direct policy search method framed in a temporal difference learning approach. In their work, the state is defined as a function of the agent-position which allows the agent at any position to observe an $m \times m \times 3$ block of surrounding voxels. Similar to other previous work, the action space is $a_x+, a_x-, a_y+, a_y-, a_z+, a_z-$. The reward is chosen as a simple binary reward function, where a positive reward is given if an action leads the agent closer to the target landmark, and a negative reward is given otherwise. Far apart from the previous work, their approach proposes a non-linear policy function approximator represented by an MLP whereas the value function approximator is presented by another MLP stacked on top of the same CNN from the policy net. Both policy (actor) and value (critic) networks are updated by actor-critic learning. To improve the learning, they introduce a partial policy-based RL to enable solving the large problem of localization by learning the optimal policy on smaller partial domains.

The objective of the partial policy is to obtain multiple simple policies on the projections of the actual action space, where the projected policies can reconstruct the policy on the original action space.

Based on the hypothesis that the position of all anatomical landmarks is interdependent and non-random within the human anatomy and this is necessary as the localization of different landmarks requires learning partly heterogeneous policies, [377] concluded that one landmark can help to deduce the location of others. For collective gain, the agents share their accumulated knowledge during training. In their approach, the state is defined as RoI centered around the location of the agent. The reward function is defined as the relative improvement in Euclidean distance between their location at time t and the target landmark location. Each agent is considered as Partially Observable Markov Decision Process (POMDP) [107] and calculates its individual reward as their policies are disjoint. In order to reduce the computational load in locating multiple landmarks and increase accuracy through anatomical interdependence, they propose a collaborative multi-agent landmark detection framework (Collab-DQN) where DQN is built upon a CNN. The backbone CNN is shared across all agents while the policy-making fully connected layers are separate for each agent.

Table 3: Comparing various DRL-based landmark detection methods. The first group on Single Landmark Detection (SLD) and the second group for Multiple Landmark Detection (MLD)

Approaches	Year	Training Technique	Actions	Remarks	Performance	Datasets and source code
SLD [105]	2017	DQN	6 action: 2 per axis	State: an axis-aligned box centered at the voxel-position. Action: move from $\vec{p}_t$ to $\vec{p}_{t+1}$. Reward: distance-based feedback	Average accuracy increase 20-30%. Lower distance error than other techniques such as SADNN [104] and 3D-DL [427]	3D CT Scan
SLD [10]	2019	DQN, DDQN, Duel DQN and Duel DDQN	6 action: 2 per axis	Environment: 3D input image. State: 3D RoI centered around the target landmark. Reward: Euclidean distance between predicted points and groundtruth points.	Duel DQN performs the best on Right Cerebellum (FS), Left Cerebellum (FS, MS) Duel DDQN is the best on Right Cerebellum (MS) DQN performs the best on Cavum Septum Pellucidum(FS, MS)	Fetal head, ultrasound scans [219]. Code

SLD [7]	2019	Actor-Critic -based Partial -Policy RL	6 action: 2 per axis	State: a function of the agent-position. Reward: binary reward function. policy function: MLP. value function: MLP	Faster and better convergence, outperforms than other conventional actor-critic and Q-learning	CT volumes: Aortic valve. CT volumes: LAA seed-point. MR images: Vertebra centers [42].
MLD [377]	2019	Collab DQN	6 action: 2 per axis	State: RoI centred around the agent. Reward: relative improvement in Euclidean distance. Each Agent is a POMDP has its own reward. Collab-DQN: reduce the computational load	Colab DQN got better results than supervised CNN and DQN	Brain MRI landmark [158], Cardiac MRI landmark [70], Fetal brain landmark [10]. Code
MLD [161]	2020	DQN	6 action 2 per axis	State: 3D image patch. Reward: Euclidean distance and $\in [-1, 1]$. Backbone CNN is share among agents Each agent has it own Fully connected layer	Detection error increased as the degree of missing information increased Performance is affected by the choice of landmarks	3D Head MR images

Different from the previous works on RL-based landmark detection, which detect a single landmark,[161] proposed a multiple landmark detection approach to better time-efficient and more robust to missing data. In their approach, each landmark is guided by one agent. The MDP is models as follows: The state is defined as a 3D image patch. The reward, clipped in [-1, +1], is defined as the difference in the Euclidean distance between the landmark predicted in the previous time step and the target, and in the landmark predicted in the current time step and the target. The action space is defined as in other previous works i.e. there are 6 actions $a_x+, a_x-, a_y+, a_y-, a_z+, a_z-$ in the action space. To enable the agents to share the information learned by detecting one landmark for use in detecting other landmarks, hard parameter sharing from multi-task learning is used. In this work, the backbone network is shared among agents and each agent has its own fully connected layer.

Table 3 summarizes and compares all approaches for DRL in landmark detection, and a basic implementation of landmark detection using DRL has been shown in Fig. 11. The figure illustrates a general implementation of landmark detection with the help of DRL, where the state is the Region of interest (ROI) around the current landmark location cropped from the image, The actions performed by the DRL agent are responsible for shifting the ROI across the image forming a new state and the reward corresponds to the improvement in

euclidean distance between ground truth and predicted landmark location with iterations as used by [105],[7],[10],[377],[161].

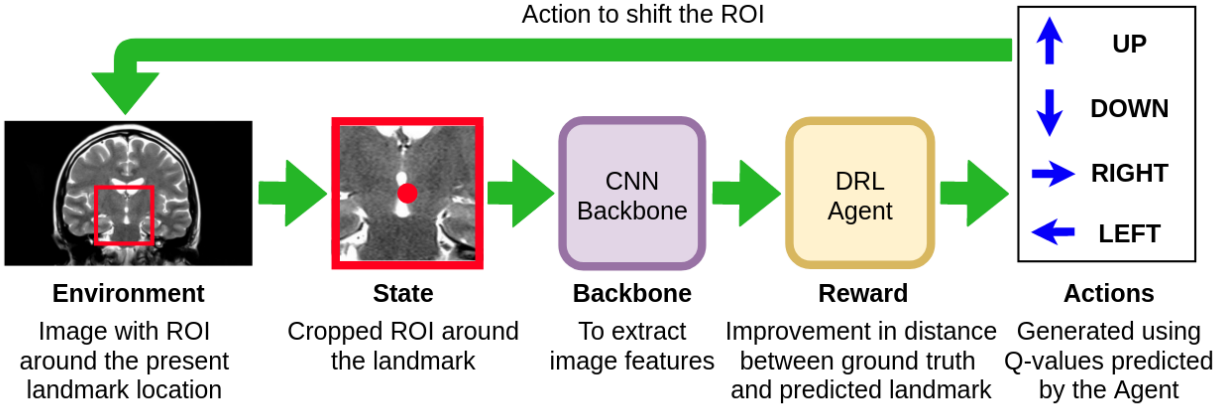

Figure 11: DRL implementation for landmark detection, The red point corresponds to the current landmark location and Red box is the Region of Interest (ROI) centered around the landmark, the actions of DRL agent shift the ROI across the image to maximize the reward corresponding to the improvement in distance between the ground truth and predicted landmark location.

6 DRL in Object Detection

Object detection is a task that requires the algorithm to find bounding boxes for all objects in a given image. Many attempts have been made towards object detection. A method for bounding box prediction for object detection was proposed by [109], in which the task was performed by extracting region proposals from an image and then feeding each of them to a CNN to classify each region. An improvement to this technique was proposed by [108], where they used the feature from the CNN to propose region proposals instead of the image itself, this resulted in fast detection. Further improvement was proposed by [309], where the authors proposed using a region proposal network (RPN) to identify the region of interest, resulting in much faster detection. Other attempts including focal loss [225] and Fast YOLO [332] have been proposed to address the imbalanced data problem in object detection with focal loss [225], and perform object detection in video on embedded devices in a real-time manner [332].

Considering MDP as the framework for solving the problem, [43] used DRL for active object localization. The authors considered 8 different actions (up, down, left, right, bigger, smaller, fatter, taller) to improve the fit of the bounding box around the object and additional action to trigger the goal state. They used a tuple of feature vector and history of actions for state and change in IOU across actions as a reward.

An improvement to [43] was proposed by [25], where the authors used a hierarchical approach for object detection by treating the problem of object detection as an MDP. In their method, the agent was responsible to find a region of interest in the image and then reducing the region of interest to find smaller regions from the previously selected region and hence forming a hierarchy. For the reward function, they used the change in Intersection over union (IOU) across the actions and used DQN as the agent. As described in their paper, two networks namely, Image-zooms and Pool45-crops with VGG-16 [340] backbone were used to extract the feature information that formed the state for DQN along with a memory vector of the last four actions.

Using a sequential search strategy, [251] proposed a method for object detection using DRL. The authors trained the model with a set of image regions where at each time step the agent returned fixate actions that specified a location in image for actor to explore next and the terminal state was specified by *done* action. The state consisted of a tuple three elements: the observed region history H_t, selected evidence region history E_t and fixate history F_t. The $fixate$ action was also a tuple of three elements: $fixate$ action, index of evidence region e_t and image coordinate of next fixate z_t. The *done* action consisted of: *done* action, index of region representing the detected output b_t and the detection confidence c_t. The authors defined the reward function that was sensitive to the detection location, the confidence at the final state and incurs a penalty for each region evaluation.

To map the inter-dependencies among the different objects, [170] proposed a tree-structured RL agent (Tree-RL) for object localization by considering the problem as an MDP. The authors in their implementation considered actions of two types: translation and scaling, where the scaling consisted of five actions whereas translation consisted of eight actions. In the specified work, the authors used the state as a concatenation of the feature vector of the current window, feature vector of the whole image, and history of taken actions. The feature vector were extracted from an ImageNet [72] [320] trained VGG-16 [340] model and for reward the change in IOU across an action was used. Tree-RL utilized a top-down tress search starting from the whole image where each window recursively takes the best action from each action group which further gives two new windows. This process is repeated recursively to find the object.

The task of breast lesion detection is a challenging yet very important task in the medical imaging field. A DRL method for active lesion detection in the breast was proposed by [246], where the authors formulated the problem as an MDP. In their formulation, a total of nine actions consisting of 6 translation actions, 2 scaling actions, and 1 trigger action were used. In the specified work, the change in dice coefficient across an action was used as the reward for scaling and translation actions, and for trigger action, the reward was $+\eta$ for dice coefficient greater than r_w and $-\eta$ otherwise, where η and r_w were the hyperparameters chosen by the authors. For network structure, ResNet [133] was used as the backbone and DQN as the agent.

Different from the previous methods, [386] proposed a method for multitask learning using DRL for object localization. The authors considered the problem as an MDP where the agent was responsible to perform a series of transformations on the bounding box using a series

of actions. Utilizing an RL framework the states consisted of feature vector and historical actions concatenated together, and a total of 8 actions for Bounding box transformation (left, right, up, down, bigger, smaller, fatter, and taller) were used. For reward the authors used the change in IOU between actions, the reward being 0 for an increase in IOU and -1 otherwise. For terminal action, however, the reward was 8 for IOU greater than 0.5 and -8 otherwise. The authors in the paper used DQN with multitask learning for localization and divided terminal action and 8 transformation actions into two networks and trained them together.

An improvement for the Region proposal networks that greedily select the ROIs was proposed by [295], where they used RL for the task. The authors in this paper used a two-stage detector similar to Fast and Faster R-CNN But used RL for the decision-making Process. For the reward, they used the normalized change in Intersection over Union (IOU).

Instead of learning a policy from a large set of data, [15] proposed a method for bounding box refinement (BAR) using RL. In the paper, once the authors have an inaccurate bounding box that is predicted by some algorithm they use the BAR algorithm to predict a series of actions for refinement of a bounding box. They considered a total of 8 actions (up, down, left, right, wider, taller, fatter, thinner) for bounding box transformation and considered the problem as a sequential decision-making problem (SDMP). They proposed an offline method called BAR-DRL and an online method called BAR-CB where training is done on every image. In BAR-DRL the authors trained a DQN over the states which consisted of features extracted from ResNet50 [133] [354] pretrained on ImageNet [72] [320] and a history vector of 10 actions. The Reward for BAR-DRL was 1 if the IOU increase after action and -3 otherwise. For BAR-CB they adapted the LinUCB [216] algorithm for an episodic scenario and considered The Histogram of Oriented Gradients (HOG) for the state to capture the outline and edges of the object of interest. The actions in the online method (BAR-CB) were the same as the offline method and the reward was 1 for increasing IOU and 0 otherwise. For both the implementations, the authors considered β as terminal IOU.

An improvement to sequential search strategy by [251] was proposed by [367], where they used a framework consisting of two modules, Coarse and fine level search. According to the authors, this method is efficient for object detection in large images (dimensions larger than 3000 pixels). The authors first performed a course level search on a large image to find a set of patches that are used by fine level search to find sub-patches. Both fine and coarse levels were conducted using a two-step episodic MDP, where The policy network was responsible for returning the probability distribution of all actions. In the paper, the authors considered the actions to be the binary action array (0,1) where 1 means that the agent would consider acquiring sub-patches for that particular patch. The authors in their implementation considered a number of patches and sub-patches as 16 and 4 respectively and used the linear combination of R_{acc} (detection recall) and R_{cost} which combines image acquisition cost and run-time performance reward.

Table 4: Comparing various DRL-based object detection methods

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets and source code
Active Object Localization [43]	2015	DQN	8 actions: up, down, left, right, bigger, smaller, fatter, taller	States: feature vector of observed region and action history. Reward: Change in IOU.	5 layer pre-trained CNN	Higher mAP as compared to methods that did not use region proposals like MultiBox [89], RegionLets [433], DetNet [356], and second best mAP as compared to R-CNN [109]	Pascal VOC-2007 [90], 2012 [91] Image Dataset.
Hierarchical Object Detection [25]	2016	DQN	5 actions: 1 action per image quarter and 1 at the center	States: current region and memory vector using Image-zooms and Pool45-crops. Reward: change in IOU.	VGG-16 [340]	Objects detected with very few region proposals per image	Pascal VOC-2007 Image Dataset [90]. Code
Visual Object Detection [251]	2016	Policy sampling and state transition algorithm	2 actions: fixate and done, where each is a tuple of three.	States: Observed region history, evidence region history and fixate history. Reward: sensitive to detection location	Deep NN [187]	Comparable mAP and lower run time as compared to other methods such as to exhaustive sliding window search(SW), exhaustive search over the CPMC and region proposal set(RP) [112] [366]	Pascal VOC 2012 Object detection challenge [91].
Tree-Structured Sequential Object Localization (Tree-RL) [170]	2016	DQN	13 actions: 8 translation, 5 scaling.	States: Feature vector of current region, and whole image. Reward: change in IOU.	CNN trained on ImageNet [72] [320]	Tree-RL with faster R-CNN outperformed RPN with fast R-CNN [108] in terms of AP and comparable results to Faster R-CNN [309]	Pascal VOC 2007 [90] and 2012 [91].

Active Breast Lesion Detection [246]	2017	DQN	9 actions: 6 translation, 2 scaling, 1 trigger	States: feature vector of current region, Reward: improvement in localization.	ResNet [133]	Comparable true positive and false positive proportions as compared to SL [253] and Ms-C [116], but with lesser mean inference time.	DCE-MRI and T1-weighted anatomical dataset [253]
Multitask object localization [386]	2018	DQN	8 actions: left, right, up, down, bigger, smaller, fatter and taller	States: feature vector, historical actions. Reward: change in IOU. different network for transformation actions and terminal actions.	Pretrained VGG-16 [340] with ImageNet [72] [320]	Better mAP as compared to MultiBox [89], Caicedo et al. [43] and second best to R-CNN [109].	Pascal VOC-2007 Image Dataset [90].
Bounding-Box Automated Refinement [15]	2020	DQN	8 actions: up, down, left, right, bigger, smaller, fatter, taller	Offline and online implementation States: feature vector for offline (BAR-DRL), HOG for online (BAR-CB). Reward: change in IOU	ResNet50 [133]	Better final IOU for boxes generated by methods such as RetinaNet [225].	Pascal VOC-2007 [90], 2012 [91] Image Dataset.
Efficient Object Detection in Large Images [367]	2020	DQN	binary action array: where 1 means that the agent would consider acquiring sub-patches for that particular patch	Course CPNet and fine FPNet level search. States: selected region. Reward: detection recall image acquisition cost. Policy: REINFORCE [351]	ResNet32 [133] for policy network. and YOLOv3 [306] with DarkNet-53 for Object detector	Higher mAP and lower run time as compared to other methods such as [99].	Caltech Pedestrian dataset (CPD) [77] Code

Organ Localization in CT [275]	2020	DQN	11 actions: 6 translation, 2 scaling, 3 deformation	States: region inside the Bounding box. Reward: change in IOU.	Architecture similar to [10]	Lower distance error for organ localization and run time as compared to other methods such as 3D-RCNN [409] and CNNs [152]	CT scans from the VISCERAL dataset [171]
Monocular 3D Object Detection [231]	2020	DQN [264]	15 actions, each modifies the 3D bounding box in a specific parameter	State: 3D bounding box parameters, 2D image of object cropped by 2D its detected bounding box. Reward: accuracy improvement after applying an action.	ResNet-101 [133]	Higher average precision (AP) compared to [268], [302], [210] and [35]	KITTI [102]

Localization of organs in CT scans is an important pre-processing requirement for taking the images of an organ, planning radiotherapy, etc. A DRL method for organ localization was proposed by [275], where the problem was formulated as an MDP. In the implementation, the agent was responsible for predicting a 3D bounding box around the organ. The authors used the last 4 states as input to the agent to stabilize the search and the action space consists of Eleven actions, 6 for the position of the bounding box, 2 for zoom in and zoom out the action, and last 3 for height, width, and depth. For Reward, they used the change the in Intersection over union (IOU) across an action.

Monocular 3D object detection is a problem where 3D bounding boxes of objects are required to be detected from a single 2D image. Even the sampling-based method is the SOTA approach, it has a huge flaw, in which most of the samples it generates do not overlap with the groundtruth. To leverage that method, [231] introduced Reinforced Axial Refinement Network (RARN) for monocular 3D object detection by utilizing an RL model to iteratively refining the sampled bounding box to be more overlapped with the groundtruth bounding box. Given a state having the coordinates of the 3D bounding box and image patch of the image, the model predicts an action out of a set of 15 actions to refine one of the bounding box coordinates in a direction at every timestep, the model is trained by DQN method with the immediate reward is the improvement in detection accuracy between every pair of timesteps. The whole pipeline, namely RAR-Net, was evaluated on the real-world KITTI dataset [102] and achieved state-of-the-art performance.

All these methods have been summarised and compared in Table 4, and a basic implementation of object detection using DRL has been shown in Fig. 12. The figure illustrates a general implementation of object detection using DRL, where the state is an image segment cropped using a bounding box produced by some other algorithm or previous iteration of DRL, actions predicted by the DRL agent predict a series of bounding box transforma-

tion to fit the object better, hence forming a new state and Reward is the improvement in Intersection over union (IOU) with iterations as used by [43],[25],[15],[386],[170],[275].

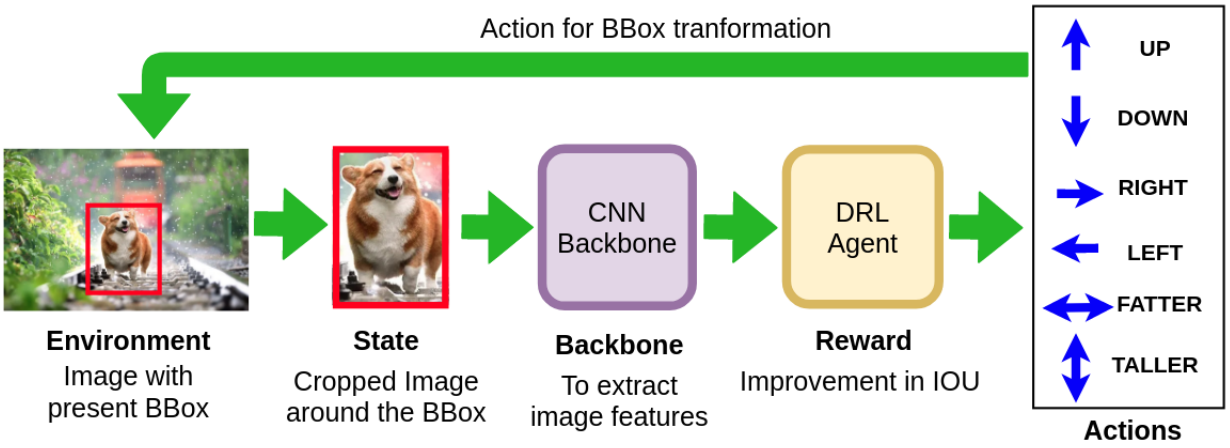

Figure 12: DRL implementation for object detection. The red box corresponds to the initial bounding box which for t=0 is predicted by some other algorithm or the transformed bounding box by previous iterations of DRL using the actions to maximize the improvement in IOU.

7 DRL in Object Tracking

Real-time object tracking has a large number of applications in the field of autonomous driving, robotics, security, and even in sports where the umpire needs accurate estimation of ball movement to make decisions. Object tracking can be divided into two main categories: Single object tracking (SOT) and Multiple object tracking (MOT).

Many attempts have been made for both SOT and MOT. SOT can be divided into two types, active and passive. In passive tracking it is assumed that the object that is being tracked is always in the camera frame, hence camera movement is not required. In active tracking, however, the decision to move the camera frame is required so that the object is always in the frame. Passive tracking has been performed by [397], [146], where [146] performed tracking for both single and multiple objects. The authors of these papers proposed various solutions to overcome common problems such as a change in lighting and occlusion. Active tracking is a little bit harder as compared to a passive one because additional decisions are required for camera movement. Some efforts towards active tracking include [74] [270] [178]. These solutions treat object detection and object tracking as two separate tasks and tend to fail when there is background noise.

An end-to-end active object tracker using DRL was proposed by [240], where the authors used CNNs along with an LSTM [139] in their implementation. They used the actor-critic algorithm [262] to calculate the probability distribution of different actions and the value

of state and used the object orientation and distance from the camera to calculate rewards. For experiments, the authors used VizDoom and Unreal Engine as the environment.

Another end-to-end method for SOT using sequential search strategy and DRL was proposed by [418]. The method included using an RNN along with REINFORCE [392] algorithm to train the network. The authors used a function $f(W_0)$ that takes in S_t and frame as input, where S_t is the object location for the first frame and is zero elsewhere. The output is fed to an LSTM module [139] with past hidden state h_t. The authors calculated the reward function by using insertion over union (IoU) and the difference between the average and max.

A deformable face tracking method that could predict bounding box along with facial landmarks in real-time was proposed by [118]. The dual-agent DRL method (DADRL) mentioned in the paper consisted of two agents: a tracking and an alignment agent. The problem of object tracking was formulated as an MDP where state consisted of image regions extracted by the bounding box and a total of 8 actions (left, right, up, down, scale-up, scale down, stop and continue) were used, where first six consists of movement actions used by tracking agent and last two for alignment agent. The tracking agent is responsible for changing the current observable region and the alignment agent determines whether the iteration should be terminated. For the tracking agent, the reward corresponded to the misalignment descent and for the alignment agent the reward was $+\eta$ for misalignment less than the threshold and $-\eta$ otherwise. The DADRL implementation also consisted of communicated message channels beside the tracking agent and the alignment agent. The tracking agent consisted of a VGG-M [340] backbone followed by a one-layer Q-Network and the alignment agent was designed as a combination of a stacked hourglass network with a confidence network. The two communicated message channels were encoded by a deconvolution layer and an LSTM unit [139] respectively.

Visual object tracking when dealing with deformations and abrupt changes can be a challenging task. A DRL method for object tracking with iterative shift was proposed by [308]. The approach (DRL-IS) consisted of three networks: The actor network, the prediction network, and the critic network, where all three networks shared the same CNN and a fully connected layer. Given the initial frame and bounding box, the cropped frame is fed to the CNNs to extract the features to be used as a state by the networks. The actions included continue, stop and update, stop and ignore, and restart. For continue, the bounding boxes are adjusted according to the output of the prediction network, for stop and update the iteration is stopped and the appearance feature of the target is updated according to the prediction network, for stop and ignore the updating of target appearance feature is ignored and restart means that the target is lost and the algorithm needs to start from the initial bounding box. The authors of the paper used reward as 1 for change in IoU greater than the threshold, 0 for change in IOU between + and - threshold, and -1 otherwise.

Considering the performance of actor-critic framework for various applications, [45] proposed an actor-critic [262] framework for real-time object tracking. The authors of the paper used a pre-processing function to obtain an image patch using the bounding box that is fed into the network to find the bounding box location in subsequent frames. For actions the

authors used $\triangle x$ for relative horizontal translation, $\triangle y$ for relative vertical translation, and $\triangle s$ for relative scale change, and for a reward they used 1 for IoU greater than a threshold and -1 otherwise. They proposed offline training and online tracking, where for offline training a pre-trained VGG-M [340] was used as a backbone, and the actor-critic network was trained using the DDPG approach [224].

An improvement to [45] for SOT was proposed by [84], where a visual tracker was formulated using DRL and an expert demonstrator. The authors treated the problem as an MDP, where the state consists of two consecutive frames that have been cropped using the bounding box corresponding to the former frame and used a scaling factor to control the offset while cropping. The actions consisted of four elements: $\triangle x$ for relative horizontal translation, $\triangle y$ for relative vertical translation, $\triangle w$ for width scaling, and $\triangle h$ for height scaling, and the reward was calculated by considering whether the IoU is greater than a threshold or not. For the agent architecture the authors used a ResNet-18 [133] as backbone followed by an LSTM unit [391][139] to encode past information, and performed training based on the on-policy A3C framework [262].

In MOT the algorithm is responsible to track trajectories of multiple objects in the given video. Many attempts have been made with MOT including [53], [55] and [143]. However, MOT is a challenging task because of environmental constraints such as crowding or object overlapping. MOT can be divided into two main techniques: Offline [53] and Online [55] [143]. In offline batch, tracking is done using a small batch to obtain tracklets and later all these are connected to obtain a complete trajectory. The online method includes using present and past frames to calculate the trajectory. Some common methods include Kalman filtering [177], Particle Filtering [284] or Markov decision [401]. These techniques however are prone to errors due to environmental constraints.

To overcome the constraints of MOT by previous methods, [401] proposed a method for MOT where the problem was approached as an MDP. The authors tracked each object in the frame through the Markov decision process, where each object has four states consisting: Active, Tracked, Lost, and Inactive. Object detection is the active state and when the object is in the lost state for a sufficient amount of time it is considered Inactive, which is the terminal state. The reward function in the implementation was learned through data by inverse RL problem [279].

Previous approaches for MOT follow a tracking by detection technique that is prone to errors. An improvement was proposed by [307], where detection and tracking of the objects were carried out simultaneously. The authors used a collaborative Q-Network to track trajectories of multiple objects, given the initial position of an object the algorithm tracked the trajectory of that object in all subsequent frames. For actions the authors used $\triangle x$ for relative horizontal translation, $\triangle y$ for relative vertical translation, $\triangle w$ for width scaling, and $\triangle h$ for height scaling, and the reward consisted of values 1,0,-1 based on the IoU.

Another method for MOT was proposed by [168], where the authors used LSTM [139] and DRL to approach the problem of multi-object tracking. The method described in the paper used three basic components: a YOLO V2 [260] object detector, many single object

trackers, and a data association module. Firstly the YOLO V2 object detector is used to find objects in a frame, then each detected object goes through the agent which consists of CNN followed by an LSTM to encode past information for the object. The state consisted of the image patch and history of past 10 actions, where six actions (right, left, up, down, scale-up, scale down) were used for bounding box movement across the frame with a stop action for the terminal state. To provide reinforcement to the agent the reward was 1 if the IOU is greater than a threshold and 0 otherwise. In their experiments, the authors used VGG-16 [340] for CNN backbone and performed experiments on MOT benchmark [201] for people tracking.

Table 5: Comparing various DRL-based object tracking methods. The First group for Single object tracking (SOT) and the second group for multi-object tracking (MOT)

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets and Source code
End to end active object tracking [240]	2017	Actor-Critic (a3c) [262]	6 actions: turn left, turn right, turn left and move forward, turn right and move forward, move forward, no-op	Environment: virtual environment. Reward: calculated using object orientation and position. Tracking Using LSTM [139]	ConvNet-LSTM	Higher accumulated reward and episode length as compared to methods like MIL [17], Meanshift [60], KCF [134].	ViZDoom [176], Unreal Engine
DRL for object tracking [418]	2017	DRLT	None	State: feature vector, Reward: change in IOU use of LSTM [139] and REINFORCE [392]	YOLO network [305]	Higher area under curve (success rate Vs overlap threshold), precision and speed (fps) as compared to STUCK [126] and DLT [384].	Object tracking benchmark [397]. Code

Dual-agent deformable face tracker [118]	2018	DQN	8 actions: left, right, up, down, scale up, scale down, stop and continue.	States: image region using Bounding box. Reward: distance error. Facial landmark detection and tracking using LSTM [139]	VGG-M [340]	Lower normalized point to point error for landmarks and higher success rate for facial tracking as compared to ICCR [187], MDM [336], Xiao et al [32], etc.	Large-scale face tracking dataset, the 300-VW test set [336]
Tracking with iterative shift [308]	2018	Actor-critic [262]	4 actions: continue, stop and update, stop and ignore and restart	States: image region using bounding box. Reward: change in IOU. Three networks: actor, critic and prediction network	3 Layer CNN and FC layer	Higher area under curve for success rate Vs overlap threshold and precision Vs location error threshold as compared to CREST [345], ADNet [416], MDNet [273], HCFT [243], SINT [358], DeepSRDCF [67], and HDT [301]	OTB-2015 [398], TempleColor [220], and VOT-2016 Dataset [186]
Tracking with actor-critic [45]	2018	Actor-critic [262]	3 actions: $\triangle x$, $\triangle y$ and $\triangle s$	States: image region using bounding box. Reward: IOU greater then threshold. Offline training, online tracking	VGG-M [340]	Higher average precision score then PTAV [93], CFNet [368], ACFN [52], SiameFC [29], ECO-HC [67], etc.	OTB-2013 [397], OTB-2015 [398] and VOT-2016 dataset [186] Code

Visual tracking and expert demonstrator [84]	2019	Actor-critic (a3c) [262]	4 actions: $\triangle x$, $\triangle y$,$\triangle w$ and $\triangle h$	States: image region using bounding box. Reward: change in IOU. SOT using LSTM [391][139]	ResNet-18 [133]	Comparable success and precision scores as compared to LADCF [408], SiamRPN [209] and ECO [66]	GOT-10k [148], LaSOT [92], UAV123 [269], OTB-100 [397], VOT-2018 [185] and VOT-2019.
Object tracking by decision making [401]	2015	TLD Tracker [174]	7 actions: corresponding to moving the object between states such as Active, tracked, lost and Inactive	States: 4 states: Active, tracked, lost and Inactive. Reward: inverse RL problem [279]	None	Comparable multiple object tracking accuracy (MOTA) and multiple object tracking precision (MOTP) [28] as compared to DPNMS [296], TCODAL [18], SegTrack [259], MotiCon [200], etc	M0T15 dataset [201] Code
Collaborative multi object tracker [307]	2018	DQN	4 actions: $\triangle x$, $\triangle y$, $\triangle w$ and $\triangle h$	States: image region using bounding box. Reward: IOU greater then threshold. 2 networks: prediction and decision network	3 Layer CNN and FC Layer	Comparable multiple object tracking accuracy (MOTA) and multiple object tracking precision (MOTP) [28] as compared to SCEA [143], MDP [401], CDADDALpb [19], AMIR15 [321]	MOT15 [201] and MOT16 [258] datasets

Multi object tracking in video [168]	2018	DQN	6 actions: right, left, up, down, scale up, scale down	States: image region using bounding box. Reward: IOU greater then threshold. Detection using YOLO-V2 [260] for detector and LSTM [139] .	VGG-16 [340]	Comparable if not better multiple object tracking accuracy (MOTA) and multiple object tracking precision (MOTP) [28] as compared to RNN-LSTM [201], LP-SSVM [401], MDPSub-CNN [199], and SiameseCNN [123]	MOT15 Dataset [201]
Multi agent multi object tracker [169]	2019	DQN	9 actions: move right, move left, move up, move down, scale up, scale down, fatter, taller and stop	States: image region using bounding box. Reward: IOU greater then threshold. YOLO-V3 [306] for detection and LSTM [139].	VGG-16 [340]	Higher running time, and comparable if not better multiple object tracking accuracy (MOTA) and multiple object tracking precision (MOTP) [28] as compared to RNN-LSTM [201], LP-SSVM [401], MDPSub-CNN [199], and SiameseCNN [123]	MOT15 challenge benchmark [201].

To address the problems in existing tracking methods such as varying numbers of targets, non-real-time tracking, etc, [169] proposed a multi-object tracking algorithm based on a multi-agent DRL tracker (MADRL). In their object tracking pipeline the authors used YOLO-V3 [306] as object detector, where multiple detections produced by YOLO-V3 were filtered using the IOU and the selected results were used as multiple agents in multiple agent detector. The input agents were fed into a pre-trained VGG-16 [340] followed by an LSTM unit [139] that could share information across agents and return the actions encoded in a

9-dimensional vector(move right, move left, move up, move down, scale-up, scale down, aspect ratio change fatter, aspect ratio change taller and stop), also a reward function similar to [168] was used.

Various works in the field of object tracking have been summarized in Table 5, and a basic implementation of object tracking using DRL has been shown in Fig. 13. The figure illustrates a general implementation of object tracking in videos using DRL, where the state consists of two consecutive frames (F_t, F_{t+1}) with a bounding box for the first frame produced by another algorithm for the first iteration or by the previous iterations of DRL agent. The actions corresponds to the moving the bounding on the image to fit the object in frame F_{t+1}, hence forming a new state with frame F_{t+1} and frame F_{t+2} along with the bounding box for frame F_{t+1} predicted by previous iteration and reward corresponds to whether IOU is greater then a given threshold as used by [118],[308],[45], [84],[307],[168],[169].

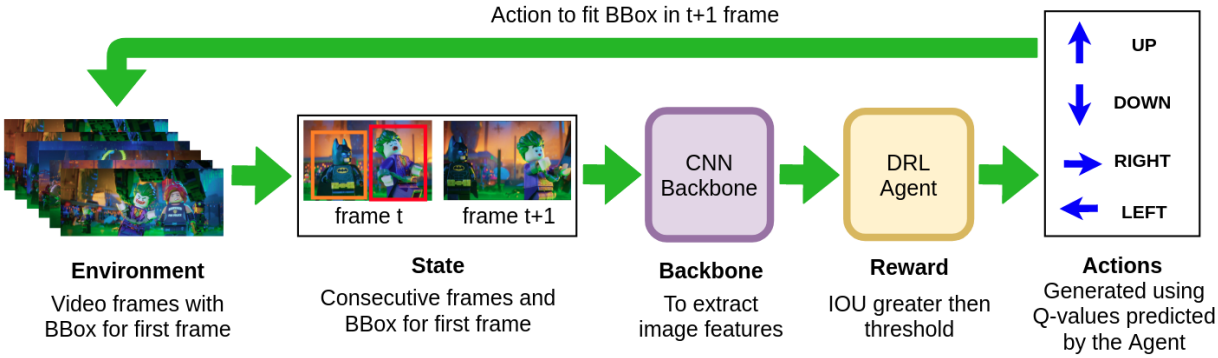

Figure 13: DRL implementation for object tracking. Here the state consists of two consecutive frames with bounding box locations for the first frame that is predicted by some object detection algorithm or by the previous iteration of DRL, the actions move the bounding box present in the first frame to fit the object in the second frame to maximize the reward which is the whether the IOU is greater than a given threshold or not.

8 DRL in Image Registration

Image registration is a very useful step that is performed on 3D medical images for the alignment of two or more images. The goal of 3D medical image registration is to find a correlation between two images from either different patients or the same patients at different times, where the images can be Computed Tomography (CT), Magnetic Resonance Imaging (MRI), or Positron Emission Tomography (PET). In the process, the images are brought to the same coordinate system and aligned with each other. The reason for image registration being a challenging task is the fact that the two images used may have a different coordinate system, scale, or resolution.

Many attempts have been made toward automated image registration. A multi-resolution strategy with local optimizers to perform 2D or 3D image registration was performed by

[359]. However, multi-resolution tends to fail with different field of views. Heuristic semi-global optimization schemes were proposed to solve this problem and used by [252] through simulated annealing and through genetic algorithm [317], However, their cost of computation was very high. A CNN-based approach to this problem was suggested by [256], and [79] proposed an optical flow method between 2D RGB images. A descriptor learned through a CNN was proposed by [395], where the authors encoded the posture and identity of a 3D object using the 2D image. Although all of these formulations produce satisfactory results yet, the methods could not be applied directly to 3D medical images.

To overcome the problems faced by previous methods, [238] proposed a method for improving probabilistic image registration via RL and uncertainty evaluation. The method involved predicting a regression function that predicts registration error from a set of features by using regression random forests (RRF) [37] method for training. The authors performed experiments on 3D MRI images and obtained an accuracy improvement of up to 25%.

Previous image registration methods are often customized to a specific problem and are sensitive to image quality and artifacts. To overcome these problems, [221] proposed a robust method using DRL. The authors considered the problem as an MDP where the goal is to find a set of transformations to be performed on the floating image to register it on the reference image. They used the gamma value for future reward decay and used the change in L2 Norm between the predicted transformation and ground truth transformation to calculate the reward. The authors also used a hierarchical approach to solve the problem with varying FOVs and resolutions.

Table 6: Comparing various DRL-based image registration methods.

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets
Image registration using uncertainity evaluation [238]	2013	DQN	Not specified	Probabilistic model using regression random forests (RRF) [37]	Not specified	Higher final Dice score (DSC) as compared to other methods like random seed selection and grid-based seed selection	3D MRI images from LONI Probabilistic Brain Atlas (LPBA40) Dataset

Robust Image registration [221]	2017	DQN	12 actions: corresponding to different transformations	States: current transformation. Reward: distance error.	5 Conv3D layers and 3 FC layers	Better success rate then ITK [153], Quasi-global [255] and Semantic registration[277]	Abdominal spine CT and CBCT dataset, Cardiac CT and CBCT
Multimodal image registration [244]	2017	Duel-DQN Double-DQN	Actions update the transformations on floating image	States: cropped 3D image. Duel-DQN for value estimation and Double DQN for updating weights.	Batch normalization followed by 5 Conv3D and 3 Maxpool layers	Lower Euclidean distance error as compared to methods like Hausdorff, ICP, DQN [264], Dueling [390], etc.	Thorax and Abdomen (ABD) dataset
Robust non-rigid agent-based registration [184]	2017	DQN	2n actions for n dimensional θ vector	States: fixed and moving image. Reward: change in transformation error. With Statistical deformation model and fuzzy action control.	Multi layer CNN, pooling and FC layers.	Higher Mean Dice score and lower Hausdorff distance as compared to methods like LCC-Demons [237] and Elastix [180].	MICCAI challenge PROMISE12 [227]
Robust Multimodal registration [349]	2018	Actor-Critic (a3c) [262]	8 actions: for different transformations	States: fixed and moving image. Reward: Distance error. Monte-carlo method with LSTM [139].	Multi layer CNN and FC layer	Comparable if not lower target registration error [96] as compared to methods like SIFT [239], Elastix [180], Pure SL, RL-matrix, RL-LME, etc.	CT and MR images

A multi-modal method for image registration was proposed by [244], where the authors used DRL for alignment of depth data with medical images. In the specified work Duel DQN was used as the agent for estimating the state value and the advantage function, and the cropped 3D image tensor of both data modalities was considered as the state. The

algorithm's goal was to estimate a transformation function that could align moving images to a fixed image by maximizing a similarity function between the fixed and moving image. A large number of convolution and pooling layer were used to extract high-level contextual information, batch normalization and concatenation of feature vector from last convolution layer with action history vector was used to solve the problem of oscillation and closed loops, and Double DQN architecture for updating the network weights was used.

Previous methods for image registration fail to cope with large deformations and variability in appearance. To overcome these issues [184] proposed a robust non-rigid agent-based method for image registration. The method involves finding a spatial transformation T_θ that can map the fixed image with the floating image using actions at each time step, that is responsible for optimizing θ. If the θ is a d dimensional vector then there will be 2d possible actions. In this work, a DQN was used as an agent for value estimation, along with a reward that corresponded to the change in θ distance between ground truth and predicted transformations across an action.

An improvement to the previous methods was proposed by [349], where the authors used a recurrent network with RL to solve the problem. Similar to [221], they considered the two images as a reference/fixed and floating/moving, and the algorithm was responsible for predicting transformation on the moving image to register it on a fixed image. In the specified work an LSTM [139] was used to encode past hidden states, Actor-critic [262] for policy estimation, and a reward function corresponding to distance between ground truth and transformed predicted landmarks were used.

Various methods in the field of Image registration have been summarized and compared in Table 6, and a basic implementation of image registration using DRL has been shown in Fig. 14. The figure illustrates a general implementation of image registration using DRL where the state consists of a fixed and floating image. The DRL agent predicts actions in form of a set of transformations on a floating image to register it onto the fixed image hence forming a new state and accepts reward in form of improvement in distance error between ground truth and predicted transformations with iterations as described by [349],[184],[221].

9 DRL in Image Segmentation

Image segmentation is one of the most extensively performed tasks in computer vision, where the algorithm is responsible for labeling each pixel position as foreground or background corresponding to the object being segmented in the image. Image segmentation has a wide variety of applications in medical, robotics, weather, etc. One of the earlier attempts with image segmentation includes [125]. With the improvement in detection techniques and introduction of CNN, new methods are introduced every year for image segmentation. Mask R-CNN [132] extended the work by Faster R-CNN [309] by adding a segmentation layer after the Bounding box has been predicted. Some earlier works include [109], [127], [128] etc. Most of these works give promising results in image segmentation. However, due to the supervised nature of CNN and R-CNN, these algorithms need a large amount of data. In fields like medical, the data is sometimes not readily available hence we needed a way to

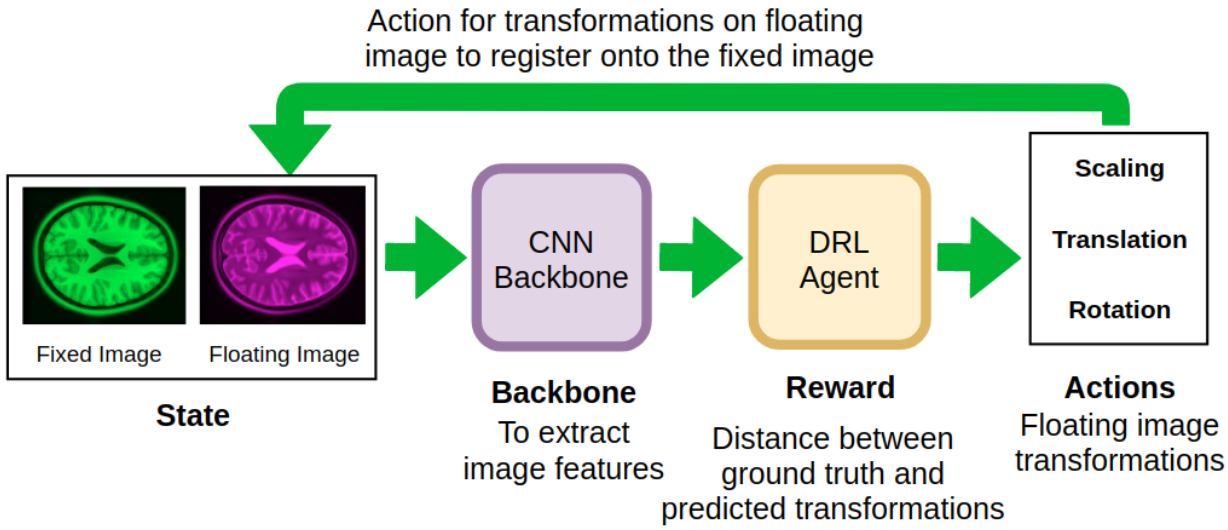

Figure 14: DRL implementation for image registration. The state consists of fixed and floating image and the actions in form of transformations are performed on the floating image so as to maximize reward by minimizing distance between the ground truth and predicted transformations.

train algorithms to perform a given task when there are data constraints. Luckily RL tends to shine when the data is not available in a large quantity.

One of the first methods for Image segmentation through RL was proposed by [324], where the authors proposed an RL framework for medical image segmentation. In their work, they used a Q-Matrix, where the actions were responsible for adjusting the threshold values to predict the mask and the reward was the normalized change in quality measure between action steps. [325] also used a similar technique of Tabular method.

To overcome the constraints of the previous method for segmentation, [310] proposed a method for indoor semantic segmentation through RL. In their paper, the authors proposed a sequential strategy using RL to combine binary object masks of different objects into a single multi-object segmentation mask. They formulated the binary mask in a Conditional Random Field Framework (CRF), and used a logistic regression version of AdaBoost [140] for classification. The authors considered the problem of adding multiple binary segmentation into one as an MDP, where the state consisted of a list of probability distributions of different objects in an image, and the actions correspond to the selection of object/background segmentation for a particular object in the sequential semantic segmentation. In the RL framework, the reward was considered in terms of pixel-wise frequency weighted Jaccard Index computed over the set of actions taken at any stage of an episode.

Interactive segmentation is the task of producing an interactive mask for objects in an image. Most of the previous works in this field greatly depend on the distribution of inputs which is user-dependent and hence produce inadequate results. An improvement was proposed by [343], where the authors proposed SeedNet, an automatic seed generation method

for robust interactive segmentation through RL. With the image and initial seed points, the algorithm is capable of generating additional seed points and image segmentation results. The implementation included Random Walk (RW) [114] as the segmentation algorithm and DQN for value estimation by considering the problem as an MDP. They used the current binary segmentation mask and image features as the state, the actions corresponded to selecting seed points in a sparse matrix of size 20×20(800 different actions were possible), and the reward consisted of the change in IOU across an action. In addition, the authors used an exponential IOU model to capture changes in IOU values more accurately.

Most of the previous work for image segmentation fail to produce satisfactory results when it comes to 3D medical data. An attempt on 3D medical image segmentation was done by [222], where the authors proposed an iteratively-refined interactive multi-agent method for 3D medical image segmentation. They proposed a method to refine an initial course segmentation produced by some segmentation methods using RL, where the state consisted of the image, previous segmentation probability, and user hint map. The actions corresponded to adjusting the segmentation probability for refinement of segmentation, and a relative cross-entropy gain-based reward to update the model in a constrained direction was used. In simple words, it is the relative improvement of previous segmentation to the current one. The authors utilized an asynchronous advantage actor-critic algorithm for determining the policy and value of the state.

Table 7: Comparing various DRL-based image segmentation methods

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets
Semantic Segmentation for indoor scenes[310]	2016	DQN	2 actions per object: object, background	States: current probability distribution. Reward: pixel-wise frequency weighted Jaccard index. Conditional Random Field Framework (CRF) and logistic regression version of AdaBoost [140] for classification.	Not Specified	Pixel-wise percentage jaccard index comparable to Gupta-L [121] and Gupta-P [120].	NYUD V2 dataset [338]

SeedNet [343]	2018	DQN, Double-DQN, Duel-DQN	800 actions: 2 per pixel	States: image features and segmentation mask. Reward: change in IOU. Random Walk (RW) [114] for segmentation algorithm.	Multi layer CNN	Better IOU then methods like FCN [236] and iFCN [407].	MSRA10K saliency dataset [49]
Iteratively refined multi agent segmentation [222]	2020	Actor-critic (a3c) [262]	1 action per voxel for adjusting segmentation probability	States: 3D image segmentation probability and hint map. Reward: cross entropy gain based framework.	R-net [378]	Better performance then methods like MinCut [183], DeepIGeoS (R-Net) [378] and InterCNN [36].	BraTS 2015[254], MM-WHS [432] and NCI-ISBI 2013 Challenge [33]
Multi-step medical image segmentation [360]	2020	Actor-critic (a3c) [262]	Actions control the position and shape of brush stroke to modify segmentation	States: image, segmentation mask and time step. Reward: change in distance error. Policy: DPG [339].	ResNet18 [133]	Higher Mean Dice score and lower Hausdorff distance then methods like Grab-Cut [315], PSPNet [425], FCN [236], U-Net [313], etc.	Prostate MR image dataset (PROMISE12, ISBI2013) and retinal fundus image dataset (REFUGE challenge dataset [285])
Anomaly Detection in Images [56]	2020	REINFORCE [392]	9 actions, 8 for directions to shift center of the extracted patch to, the last action is to switch to a random new image	Environment: input image to the model. State: observed patch from the image centered by predicted center of interest.	None	Superior performance in [27] and [337] on all metrics e.g. precision, recall and F1 when compared with U-Net [313] and baseline unsupervised method in [27] but only wins on recall in [44]	MVTec AD [27], NanoTWICE [44], CrackForest [337]

Further improvement in the results of medical image segmentation was proposed by [360]. The authors proposed a method for multi-step medical image segmentation using RL, where they used a deep deterministic policy gradient method (DDPG) based on actor-critic framework [262] and similar to Deterministic policy gradient (DPG) [339]. The authors used ResNet18 [133] as backbone for actor and critic network along with batch normalisation [157] and weight normalization with Translated ReLU [400]. In their MDP formulation, the state consisted of the image along with the current segmentation mask and step-index, and the reward corresponded to the change in mean squared error between the predicted segmentation and ground truth across an action. According to the paper the action was defined to control the position and shape of brush stroke used to modify the segmentation.

An example in image segmentation outside the medical field is [56] proposing to tackle the problem of anomalies detection and segmentation in images (i.e. damaged pins of an IC chip, small tears in woven fabric). [56] utilizes an additional module to attend only on a specific patch of the image centered by a predicted center instead of the whole image, this module helps a lot in reducing the imbalance between normal regions and abnormal locations. Given an image, this module, namely Neural Batch Sampling (NBS), starts from a random initiated center and recurrently moves that center by eight directions to the abnormal location in the image if it exists, and it has an additional action to stop moving the center when it has already converged to the anomaly location or there is not any anomaly can be observed. The NBS module is trained by REINFORCE algorithm [392] and the whole model is evaluated on multiple datasets e.g. MVTec AD [27], NanoTWICE [44], CrackForest [337].

Various works in the fields of Image segmentation have been summarised and compared in Table 7, and a basic implementation of image segmentation using DRL has been shown in Fig. 15. The figure shows a general implementation of image segmentation using DRL. The states consist of the image along with user hint (landmarks or segmentation mask by the other algorithm) for the first iteration or segmentation mask by the previous iteration. The actions are responsible for labeling each pixel as foreground and background and reward corresponds to an improvement in IOU with iterations as used by [343],[222].

10 DRL in Video Analysis

Object segmentation in videos is a very useful yet challenging task in computer vision field. Video object segmentation task focuses on labelling each pixel for each frame as foreground or background. Previous works in the field of video object segmentation can be divided into three main methods. unsupervised [288][402], weakly supervised [48][163] [419] and semi-supervised [41] [164][292].

A DRL-based framework for video object segmentation was proposed by [323], where the authors divided the image into a group of sub-images and then used the algorithm on each of the sub-image. They proposed a group of actions that can perform to change the local values inside each sub-image and the agent received reward based on the change in the quality of segmented object inside each sub-image across an action. In the proposed method deep belief network (DBN) [47] was used for approximating the Q-values.

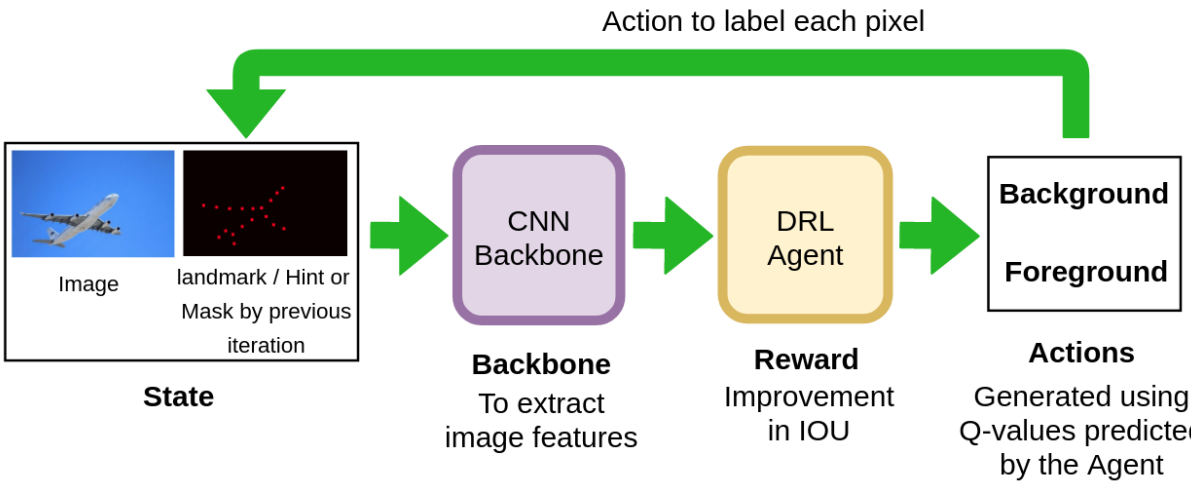

Figure 15: DRL implementation for Image segmentation. The state consists of the image to be segmented along with a user hint for t=0 or the segmentation mask by the previous iterations. The DRL agent performs actions by labeling each pixel as foreground and background to maximize the improvement in IOU over the iterations.

Surgical gesture recognition is a very important yet challenging task in the computer vision field. It is useful in assessing surgical skills and for efficient training of surgeons. A DRL method for surgical gesture classification and segmentation was proposed by [228]. The proposed method could work on features extracted by video frames or kinematic data frames collected by some means along with the ground truth labels. The problem of classification and segmentation was considered as an MDP, where the state was a concatenation of TCN [195][199] features of the current frame, 2 future frames a specified number of frames later, transition probability of each gesture computed from a statistical language model [311] and a one-hot encoded vector for gesture classes. The actions could be divided into two sub-actions, One to decide optimal step size and one for choosing gesture class, and the reward was adopted in a way that encouraging the agent to adopt a larger step and also penalizes the agent for errors caused by the action. The authors used Trust Region Policy Optimization (TRPO) [326] for training the policy and a spacial CNN [196] to extract features.

Earlier approaches for video object segmentation required a large number of actions to complete the task. An Improvement was proposed by [124], where authors used an RL method for object segmentation in videos. They proposed a reinforcement cutting-agent learning framework, where the cutting-agent consists of a cutting-policy network (CPN) and a cutting-execution network (CEN). The CPN learns to predict the object-context box pair, while CEN learns to predict the mask based on the inferred object-context box pair. The authors used MDP to solve the problem in a semi-supervised fashion. For the state of CPN the authors used the input frame information, the action history, and the segmentation mask provided in the first frame. The output boxes by CPN were input for the CEN. The actions for CPN network included 4 translation actions (Up, Down, Left, Right), 4 scaling

action (Horizontal shrink, Vertical shrink, Horizontal zoom, Vertical zoom), and 1 terminal action (Stop), and the reward corresponded to the change in IOU across an action. For the network architecture, a Fully-Convolutional DenseNet56 [166] was used as a backbone along with DQN as the agent for CPN and down-sampling followed by up-sampling architecture for CEN.

Unsupervised video object segmentation is an intuitive task in the computer vision field. A DRL method for this task was proposed by [111], where the authors proposed a motion-oriented unsupervised method for image segmentation in videos (MOREL). They proposed a two-step process to achieve the task in which first a representation of input is learned to understand all moving objects through unsupervised video object segmentation, Then the weights are transferred to the RL framework to jointly train segmentation network along with policy and value function. The first part of the method takes two consecutive frames as input and predicts a number of segmentation masks, corresponding object translations, and camera translations. They used a modified version of actor-critic [262][329][371] for the network of first step. Following the unsupervised fashion, the authors used the approach similar to [375] and trained the network to interpolate between consecutive frames and used the masks and translations to estimate the optical flow using the method that was proposed in Spatial Transformer Networks [159]. They also used structural dissimilarity (DSSIM) [388] to calculate reconstruction loss and actor-critic [262] algorithm to learn policy in the second step.

A DRL method for dynamic semantic face video segmentation was proposed by [387], where Deep Feature Flow [431] was utilized as the feature propagation framework and RL was used for an efficient and effective scheduling policy. The method involved dividing frames into key (I_k) and non-key (I_i), and using the last key frame features for performing segmentation of non-key frame. The actions made by the policy network corresponded to categorizing a frame as I_k or I_i and the state consisted of deviation information and expert information, where the deviation information described the difference between current I_i and last I_k and expert information encapsulated the key decision history. The authors utilized FlowNet2-s model [156] as an optical flow estimation function, and divided the network into feature extraction module and task-specific module. After policy network which consisted of one convolution layer, 4 fully connected layers and 2 concatenated channels consisting of KAR (Key all ratio: Ratio between key frame and every other frame in decision history) and LKD (Last key distance: Distance between current and last key frame) predicted the action, If the current frame is categorized as key frame the feature extraction module produced the frame features and task-specific module predicted the segmentation, However if the frame is categorized as a non-key frame the features from the last key frame along with the optical flow was used by the task-specific module to predict the segmentation. The authors proposed two types of reward functions, The first reward function was calculated by considering the difference between the IOU for key and non-key actions. The second reward function was proposed for a situation when ground truth was not available and was calculated by considering the accuracy score between segmentation for key and non-key actions.

Table 8: Comparing various methods associated with video. First group for video object segmentation, second group for action recognition and third group for video summarisation

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets and Source code
Object segmentation in videos[323]	2016	Deep Belief Network [47]	Actions changed local values in sub-images	States: sub-images. Reward: quality of segmentation.	Not specified	Not specified	Not specified
Surgical gesture segmentation and classification [228]	2018	Trust Region Policy Optimization (TRPO) [326]	2 types: optimal step size and gesture class	States: TCN [[195], [199]] and future frames. Reward: encourage larger steps and minimize action errors. Statistical language model [311] for gesture probability.	Spacial CNN [196]	Comparable accuracy, and higher edit and F1 scores as compared to methods like SD-SDL [331], Bidir LSTM [76], LC-SC-CRF [197], Seg-ST-CNN [196], TCN [198], etc	JIGSAWS [[6], [100]] benchmark dataset Code

Cutting agent for video object segmentation [124]	2018	DQN	8 actions: 4 translation actions (Up, Down, Left, Right), 4 scaling action (Horizontal shrink, Vertical shrink, Horizontal zoom, Vertical zoom) and 1 terminal action (Stop)	States: input frame, action history and segmentation mask. Reward: change in IOU. cutting-policy network for box-context pair and cutting-execution network for mask generation	DenseNet [166]	Higher mean region similarity, counter accuracy and temporal stability [293] as compared to methods like MSK [292], ARP [173], CTN [165], VPN [164], etc.	DAVIS dataset [293] and the YouTube Objects dataset [162], [300]
Unsupervised video object segmentation (MOREL) [111]	2018	Actor-critic (a2c) [262]	Not specified	States: consecutive frames. Two step process with optical flow using Spatial Transformer Networks [159] and reconstruction loss using structural dissimilarity [388].	Multi-layer CNN	Higher total episodic reward as compared to methods that used actor-critic without MOREL	59 Atari games. Code

Face video segmentation [387]	2020	Not specified	2 actions: categorising a frame as a key or a non-key	States: deviation information which described the difference between current non-key and last key decision, and expert information which encapsulated the key decision history. Reward: improvement in mean IOU/accuracy score between segmentation of key and non-key frames	Multi-layer CNN	Higher mean IOU then other methods like DVSNet [410], DFF [431].	300VW dataset [336] and Cityscape dataset [61]
Multi-agent Video Object Segmentation [373]	2020	DQN	Actions of 2 types: movement actions (up, down, left and right) and set action (action to place location prior at a random location on the patch)	States: input frame, optical flow [156] from previous frame and action history. Reward: clicks generated by gamification. Down-sampling and up-sampling similar to U-Net [313]	DenseNet [147]	Higher mean region similarity and contour accuracy [293] as compared to semi-supervised methods such as SeamSeg [14], BSVS [248], VSOF [363], OSVOS [41] and weakly-supervised methods such as GVOS [346], Spftn [419]	DAVIS-17 dataset [293]

Skeleton-based Action Recognition [357]	2018	DQN	3 actions: shifting to left, staying the same and shifting to right	States: Global video information and selected frames. Reward: change in categorical probability. 2 step network (FDNet) to filter frames and GCNN for action labels	Multi-layer CNN	Higher cross subject and cross view metrics for NTU+RGBD dataset [333], and higher accuracy for SYSU-3D [145] and UT-Kinect Dataset [399] when compared with other methods like Dynamic Skeletons [145], HBRNN-L [81], Part-aware LSTM [333], LieNet-3Blocks [151], Two-Stream CNN [211], etc.	NTU+RGBD [333], SYSU-3D [145] and UT-Kinect Dataset [399]
Video summarisation [429]	2018	DQN	2 actions: selecting and rejecting the frame	tates: bidirectional LSTM [150] produced states by input frame features. Reward: Diversity-Representativeness Reward Function.	GoogLeNet [355]	Higher F-score [421] as compared to methods like Uniform sampling, K-medoids, Dictionary selection [88], Video-MMR [218], Vsumm [69], etc.	TVSum [344] and SumMe [122]. Code

Video summarization [430]	2018	Duel DQN Double DQN	2 actions: selecting and rejecting the frame	States: sequence of frames Reward: Diversity-Representativeness Reward Function 2 stage implementation: classification and summarisation network using bidirectional GRU network and LSTM [150]	GoogLeNet [355]	Higher F-score [421] as compared to methods like Dictionary selection [88], GAN [245], DR-DSN [429], Backprop-Grad [287], etc in most cases.	TVSum [344] and CoSum [57] datasets. Code
Video summarization in Ultrasound [233]	2020	Not specified	2 actions: selecting and rejecting the frame	States: frame latent scores Reward: R_{det}, R_{rep} and R_{div} bidirectional LSTM [150] and Kernel temporal segmentation [298]	Not specified	Higher F1-scores in supervised and unsupervised fashion as compared to methods like FCSN [312] and DR-DSN [429].	Fetal Ultrasound [179]

Video object segmentation using human-provided location priors have been capable of producing promising results. An RL method for this task was proposed by [373], in which the authors proposed MASK-RL, a multiagent RL framework for object segmentation in videos. They proposed a weakly supervised method where the location priors were provided by the user in form of clicks using gamification (Web game to collect location priors by different users) to support the segmentation and used a Gaussian filter to emphasize the areas. The segmentation network is fed a 12 channel input tensor that contained a sequence of video frames and their corresponding location priors (3×3 color channels + three gray-scale images). The authors used a fully convoluted DenseNet [147] with down-sampling and up-sampling similar to U-Net [313] and an LSTM [139] for the segmentation network. For the RL method, the actor takes a series of steps over a frame divided into a grid of equal size patches and makes the decision whether there is an object in the patch or not. In their MDP formulation the states consisted of the input frame, optical flow (computed by [156]) from the previous frame, patch from the previous iteration, and the episode location history, the actions consisted of movement actions (up, down, left and right) and set action (action to place location prior at a random location on the patch), and two types of rewards one for set actions and one for movement actions were used. The reward was calculated using the

clicks generated by the game player.

Action recognition is an important task in the computer vision field which focuses on categorizing the action that is being performed in the video frame. To address the problem a deep progressive RL (DPRL) method for action recognition in skeleton-based videos was proposed by [357]. The authors proposed a method that distills the most informative frames and discards ambiguous frames by considering the quality of the frame and the relationship of the frame with the complete video along with a graph-based structure to map the human body in form of joints and vertices. DPRL was utilized to filter out informative frames in a video and graph-based CNNs were used to learn the spatial dependency between the joints. The approach consisted of two sub-networks, a frame distillation network (FDNet) to filter a fixed number of frames from input sequence using DPRL and GCNN to recognize the action labels using output in form of a graphical structure by the FDNet. The authors modeled the problem as an MDP where the state consisted of the concatenation of two tensors F and M, where F consisted of global information about the video and M consisted of the frames that were filtered, The actions which correspond to the output of FDNet were divided into three types: shifting to left, staying the same and shifting to the right, and the reward function corresponded to the change in probability of categorizing the video equal to the ground truth clipped it between [-1 and 1] and is provided by GCNN to FDNet.

Video summarization is a useful yet difficult task in the computer vision field that involves predicting the object or the task that is being performed in a video. A DRL method for unsupervised video summarisation was proposed by [429], in which the authors proposed a Diversity-Representativeness reward system and a deep summarisation network (DSN) which was capable of predicting a probability for each video frame that specified the likeliness of selecting the frame and then take actions to form video summaries. They used an encode-decoder framework for the DSN where GoogLeNet [355] pre-trained on ImageNet [320] [72] was used as an encoder and a bidirectional RNNs (BiRNNs) topped with a fully connected (FC) layer was used as a decoder. The authors modeled the problem as an MDP where the action corresponded to the task of selecting or rejecting a frame. They proposed a novel Diversity-Representativeness Reward Function in their implementation, where diversity reward corresponded to the degree of dissimilarity among the selected frames in feature space, and representativeness reward measured how well the generated summary can represent the original video. For the RNN unit they used an LSTM [139] to capture long-term video dependencies and used REINFORCE algorithm for training the policy function.

An improvement to [429] was proposed by [430], where the summarisation network was implemented using Deep Q-learning (DQSN), and a trained classification network was used to provide a reward for training the DQSN. The approach included using (Bi-GRU) bidirectional recurrent networks with a gated recurrent unit (GRU) [50] for both classification and summarisation network. The authors first trained the classification network using a supervised classification loss and then used the classification network with fixed weights for the classification of summaries generated by the summarisation network. The summarisation network included an MDP-based framework in which states consisted of a sequence of video frames and actions reflected the task of either keeping the frame or discarding it. They used

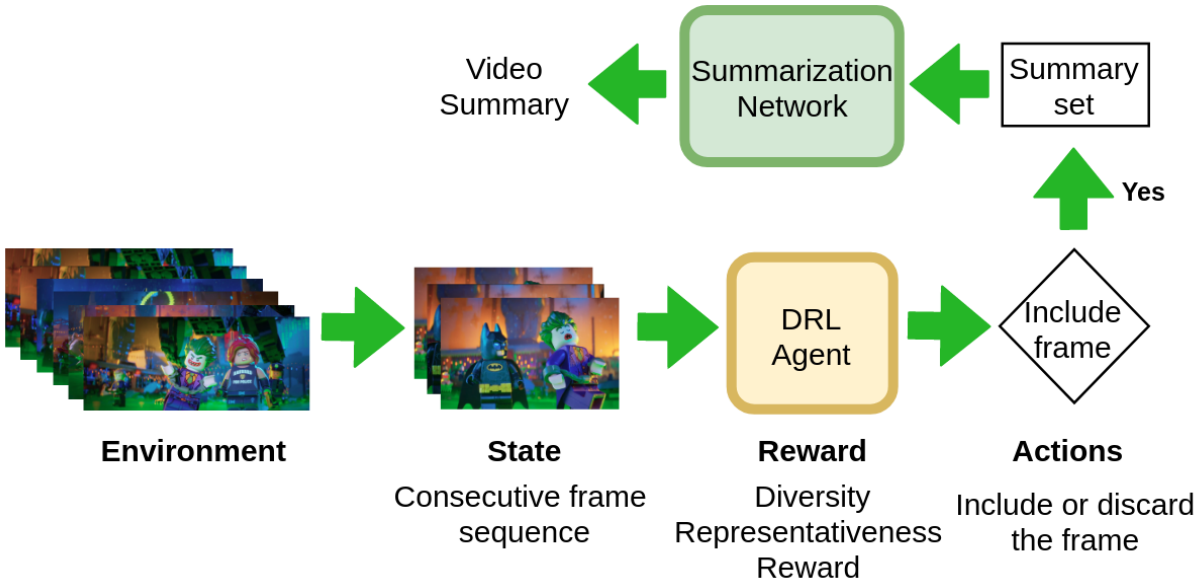

Figure 16: DRL implementation for video summarization. For state a sequence of consecutive frames are used and the DRL agent decided whether to include the frame in the summary set that is used to predict video summary.

a structure similar to Duel-DQN where value function and advantage function are trained together. In their implementation, the authors considered 3 different rewards: Global Recognisability reward using the classification network with +1 as reward and -5 as punishment, Local Relative Importance Reward for rewarding the action of accepting or rejecting a frame by summarisation network, and an Unsupervised Reward that is computed globally using the unsupervised diversity-representativeness (DR) reward proposed in [429]. The authors trained both the networks using the features generated by GoogLeNet [355] pre-trained on ImageNet [72].

A method for video summarization in Ultrasound using DRL was proposed by [233], in which the authors proposed a deep summarisation network in an encoder-decoder fashion and used a bidirectional LSTM (Bi-LSTM) [150] for sequential modeling. In their implementation, the encoder-decoder convolution network extracted features from video frames and fed them into the Bi-LSTM. The RL network accepted states in form of latent scores from Bi-LSTM and produced actions, where the actions consist of the task of including or discarding the video frame inside the summary set that is used to produce video summaries. The authors used three different rewards R_{det}, R_{rep} and R_{div} where R_{det} evaluated the likelihood of a frame being a standard diagnostic plane, R_{rep} defined the representativeness reward and R_{div} was the diversity reward that evaluated the quality of the selected summary. They used Kernel temporal segmentation (KTS) [298] for video summary generalization.

Various works associated with video analysis have been summarised and compared in Table 8 and a basic implementation of video summarization using DRL has been shown in Fig. 16, where the states consist of a sequence of video frames. The DRL agent performs

actions to include or discard a frame from the summary set that is later used by the summarization network to predict video summary. Each research paper propose their own reward function for this application, for example [429] and [430] used diversity representativeness reward function and [233] used a combination of various reward functions.

11 Others Applications

Object manipulation refers to the task of handling and manipulating an object using a robot. A method for deformable object manipulation using RL was proposed by [250], where the authors used a modified version of Deep Deterministic Policy Gradients (DDPG) [224]. They used the simulator Pybullet [63] for the environment where the observation consisted of a $84 \times 84 \times 3$ image, the state consists of joint angles and gripper positions and action of four dimensions: first three for velocity and lasts for gripper velocity was used. The authors used sparse reward for the task that returns the reward at the completion of the task. They used the algorithm to perform tasks such as folding and hanging cloth and got a success rate of up to 90%.

Visual perception-based control refers to the task of controlling robotic systems using a visual input. A virtual to real method for control using semantic segmentation was proposed by [142], in which the authors combined various modules such as, Perception module, control policy module, and a visual guidance module to perform the task. For the perception module, the authors directly used models such as DeepLab [46] and ICNet [424], pre-trained on ADE20K [428] and Cityscape [61], and used the output of these model as the state for the control policy module. They implemented the control policy module using the actor-critic [262] framework, where the action consisted of forward, turn right, and turn left. In their implementation, a reward of 0.001 is given at each time step. They used the Unity3D engine for the environment and got higher success and lower collision rate than other implementations such as ResNet-A3C and Depth-A3C.

Automatic tracing of structures such as axons and blood vessels is an important yet challenging task in the field of biomedical imaging. A DRL method for sub-pixel neural tracking was proposed by [65], where the authors used 2D grey-scale images as the environment. They considered a full resolution 11px $\times$ 11px window and a 21px $\times$ 21px window down-scaled to 11px $\times$ 11px as state and the actions were responsible for moving the position of agent in 2D space using continuous control for sub-pixel tracking because axons can be smaller then a pixel. The authors used a reward that was calculated using the average integral of intensity between the agent's current and next location, and the agent was penalized if it does not move or changes directions more than once. They used an Actor-critic [262] framework to estimate value and policy functions.

An RL method for automatic diagnosis of acute appendicitis in abdominal CT images was proposed by [8], in which the authors used RL to find the location of the appendix and then used a CNN classifier to find the likelihood of Acute Appendicitis, finally they defined a region of low-entropy (RLE) using the spatial representation of output scores to obtain optimal diagnosis scores. The authors considered the problem of appendix localization as

an MDP, where the state consisted of a $50 \times 50 \times 50$ volume around the predicted appendix location, 6 actions (2 per axis) were used and the reward consisted of the change in distance between the predicted appendix location and actual appendix location across an action. They utilized an Actor-critic [262] framework to estimate policy and value functions.

Table 9: Comparing various other methods besides landmark detection, object detection, object tracking, image registration, image segmentation, video analysis, that is associated with DRL

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets Source code
Object manipulation [250]	2018	Rainbow DDPG	4 actions: 3 for velocity 1 for gripper velocity	State: joint angle and gripper position. Reward: at the end of task.	Multi layer CNN	Success rate up to 90%	Pybullet [63]. Code
Visual based control [142]	2018	Actor-critic (a3c) [262]	3 actions: forward, turn right and turn left	State: output by backbones. Reward: 0.001 at each time-step.	DeepLab [46] and ICNet [424]	Higher success and lower collision rate then ResNet-A3C and Depth-A3C	Unity3D engine
Automatic tracing [65]	2019	Actor-critic [262]	4 actions	State: 11px × 11px window. Reward: average integral of intensity between the agent's current and next location.	Multi layer CNN	Comparable convergence % and average error as compared to other methods like Vaa3D software [291] and APP2 neuron tracer [403]	Synthetic and microscopy dataset [24]
Automatic diagnosis (RLE) [8]	2019	Actor-critic [262]	6 actions: 2 per axis	State: $50 \times 50 \times 50$ volume. Reward: change in distance error.	Fully connected CNN	Higher sensitivity and specificity as compared to only CNN classifier and CNN classifier with RL without RLE.	Abdominal CT Scans

Learning to paint [149]	2019	Actor-critic with DDPG	Actions control the stoke parameter: location, shape, color and transparency	State: Reference image, Drawing canvas and time step. Reward: change in discriminator score (calculated by WGAN-GP [117] across an action. GANs [113] to improve image quality	ResNet18 [133]	Able to replicate the original images to a large extent, and better resemblance to the original image as compared to SPIRAL [98] with same number of brush strokes.	MNIST [202], SVHN [276], CelebA [235] and ImageNet [320]. Code
Guiding medical robots [129]	2020	Double-DQN, Duel-DQN	5 actions: up, down, left, right and stop	State: probe position. Reward: Move closer: 0.05, Move away: -0.1, Correct stop: 1.0, Incorrect stop: -0.25.	ResNet18 [133]	Higher % of policy correctness and reachability as compared to CNN Classifier, where MS-DQN showed the best results	Ultrasound Images Dataset. Code
Crowd counting [230]	2020	DQN	9 actions: -10, -5, -2, -1, +1, +2, +5, +10 and end	State: weight vector W_t and image feature vector FV_I. Reward: Intermediate reward and ending reward	VGG16 [340]	Lower/comparable mean squared error (MSE) and mean absolute error (MAE) as compared to other methods like DRSAN [232], PGCNet [412], MBTTBF [341], S-DCNet [405], CAN [234], etc.	The ShanghaiTech (SHT) Dataset [423], The UCFCC50 Dataset [154] and The UCF-QNRF Dataset [155]. Code

Automated Exposure bracketing [389]	2020	Not Specified	selecting optimal bracketing from candidates	State: quality of generated HDR image. Reward: improvement in peak signal to noise ratio	AlexNet [188]	Higher peak signal to noise ratio as compared to other methods like Barakat [22], Pourreza-Shahri [299], Beek [369], etc.	Proposed benchmark dataset. Code/data
Urban Autonomous driving [361]	2020	Rainbow-IQN	36 or 108 actions: (9×4) or (27×4), 9/27 steering and 4 throttle	State: environment variables like traffic light, pedestrians, position with respect to center lane. Reward: generated by CARLA waypoint API	Resnet18 [133]	Won the 2019 camera only CARLA challenge [314].	CARLA urban driving simulator [314] Code
Mitigating bias in Facial Recognition [382]	2020	DQN	3 actions:(Margin adjustment) staying the same, shifting to a larger value and shifting to a smaller value	State: the race group, current adaptive margin and bias between the race group and Caucasians. Reward: change in the sum of inter-class and intra-class bias	Multi-layer CNN	Proposed algorithm had higher verification accuracy as compared to other methods such as CosFace [379] and ArcFace [73].	RFW [383] and proposed novel datasets: BUPT-Globalface and BUPT-Balancedface Data
Attention mechanism to improve CNN performance [212]	2020	DQN [264]	Actions are weights for every location or channel in the feature map.	State: Feature map at each intermediate layer of model. Reward: predicted by a LSTM model.	ResNet-101 [133]	Improves the performances of [144], [205] and [396], which attend on feature channel, spatial-channel and style, respectively	ImageNet [72]

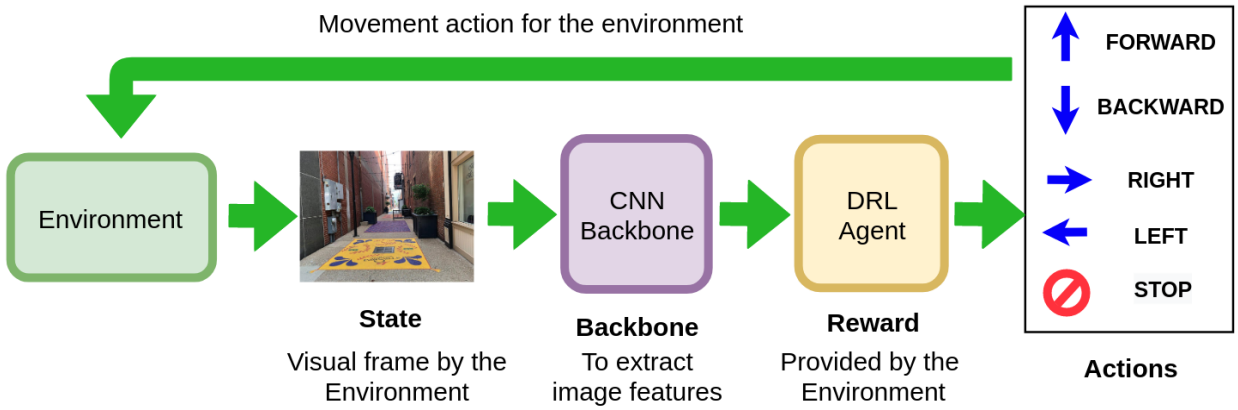

Figure 17: A general DRL implementation for agent movement with visual inputs. The state is provided by the environment based on which the agent performs movement actions to get a new state and a reward from the environment.

Painting using an algorithm is a fantastic yet challenging task in the computer vision field. An automated painting method was proposed by [149], where the authors introduced a model-based DRL technique for this task. The specified work involved using a neural renderer in DRL, where the agent was responsible for making a decision about the position and color of each stroke, and making long-term decisions to organize those strokes into a visual masterpiece. In this work, GANs [113] were employed to improve image quality at pixel-level and DDPG [224] was utilized for determining the policy. The authors formulated the problem as an MDP, where the state consisted of three parts: the target image I, the canvas on which actions (paint strokes) are performed C_t, and the time step. The actions corresponding to a set of parameters that controlled the position, shape, color, and transparency of strokes, and for reward the WGAN with gradient penalty (WGAN-GP) [117] was used to calculate the discriminator score between the target image I and the canvas C_t, and the change in discriminator score across an action (time-step) was used as the reward. The agent that predicted the stroke parameters was trained in actor-critic [262] fashion with backbone similar to Resnet18 [133], and the stroke parameters by the actor were used by the neural renderer network to predict paint strokes. The network structure of the neural renderer and discriminator consisted of multiple convolutions and fully connected blocks.

A method for guiding medical robots using Ultrasound images with the help of DRL was proposed by [129]. The authors treated the problem as an MDP where the agent takes the Ultrasound images as input and estimates the state hence the problem became Partially observable MDP (POMDP). They used Double-DQN and Duel-DQN for estimating Q-Values and ResNet18 [133] backbone for extracting feature to be used by the algorithm along with Prioritized Replay Memory. In their implementation the action space consisted of 8 actions (up, down, left, right, and stop), probe position as compared to the sacrum was used as the state and the reward was calculated by considering the agent position as compared to the target (Move closer: 0.05, Move away: -0.1, Correct stop: 1.0, Incorrect stop: -0.25). In

their implementation, the authors proposed various architectures such as V-DQN, M-DQN, and MS-DQN for the task and performed experimentation on Ultrasound images.

Crowd counting is considered a tricky task in computer vision and is even trickier for humans. A DRL method for crowd counting was proposed by [230], where the authors used sequential decision making to approach the task through RL. In the specified work, the authors proposed a DQN agent (LibraNet) based on the motivation of a weighing scale. In their implementation crowd counting was modeled using a weighing scale where the agent was responsible for adding weights on one side of the scale sequentially to balance the crowded image on the other side. The problem of adding weights on one side of the pan for balancing was formulated as an MDP, where state consisted weight vector W_t and image feature vector FV_I, and the actions space was defined similar to scale weighing and money system [372] containing values $(-10, -5, -2, -1, +1, +2, +5, +10, end)$. For reinforcing the agent two different rewards: ending reward and intermediate reward were utilized, where ending reward (following [43]) was calculated by comparing the absolute value error between the ground-truth count and the accumulated value with the error tolerance, and three counting specific rewards: force ending reward, guiding reward and squeezing reward were calculated for the intermediate rewards.

Exposure bracketing is a method used in digital photography, where one scene is captured using multiple exposures for getting a high dynamic range (HDR) image. An RL method for automated bracketing selection was proposed by [389]. For flexible automated bracketing selection, an exposure bracketing selection network (EBSNet) was proposed for selecting optimal exposure bracketing and a multi-exposure fusion network (MEFNet) for generating an HDR image from selected exposure bracketing which consisted of 3 images. Since there is no ground truth for the exposure bracketing selection procedure, an RL scheme was utilized to train the agent (EBSNet). The authors also introduced a novel dataset consisting of a single auto-exposure image that was used as input to the EBSNet, 10 images with varying exposures from which EBSNet generated probability distribution for 120 possible candidate exposure bracketing (C^3_{10}) and a reference HDR image. The reward for EBSNet was defined as the difference between peak signal-to-noise ratio between generated and reference HDR for the current and previous iteration, and the MEFNet was trained by minimizing the Charbonnier loss [23]. For performing the action of bracketing selection ESBNet consisted of a semantic branch using AlexNet [188] for feature extraction, an illumination branch to understand the global and local illuminations by calculating a histogram of input and feeding it to CNN layers, and a policy module to generate a probability distribution for the candidate exposure bracketing from semantic and illumination branches. The neural network for MEFNet was derived from HDRNet [103].

Autonomous driving in an urban environment is a challenging task, because of a large number of environmental variables and constraints. A DRL approach to this problem was proposed by [361]. In their implementation, the authors proposed an end-to-end model-free RL method, where they introduced a novel technique called Implicit Affordances. For the environment, the CARLA Simulator [80] was utilized, which provided the observations and the training reward was obtained by using the CARLA waypoint API. In the novel implicit

affordances technique the training was broken into two phases, The first phase included using a Resnet18 [133] encoder to predict the state of various environment variables such as traffic light, pedestrians, position with respect to the center lane, etc., and the output features were used as a state for the RL agent, For which a modified version of Rainbow-IQN Ape-X [136] was used. CARLA simulator accepts actions in form of continuous steering and throttle values, so to make it work with Rainbow-IQN which supports discrete actions, the authors sampled steering values into 9 or 27 discrete values and throttle into 4 discrete values (including braking), making a total of $36(9 \times 4)$ or $108(27 \times 4)$ actions.

Racial discrimination has been one of the hottest topics of the 21st century. To mitigate racial discrimination in facial recognition, [382] proposed a facial recognition method using skewness-aware RL. According to the authors, the reason for racial bias in facial recognition algorithms can be either due to the data or due to the algorithm, so the authors provided two ethnicity-aware datasets, BUPT-Globalface and BUPT-Balancedface along with an RL based race balanced network (RL-RBN). In their implementation, the authors formulated an MDP for adaptive margin policy learning where the state consisted of three parts: the race group (0: Indian, 1: Asian, 2: African), current adaptive margin, and bias or the skewness between the race group and Caucasians. A DQN was used as a policy network that performed three actions (staying the same, shifting to a larger value, and shifting to a smaller value) to change the adaptive margin, and accepted reward in form of change in the sum of inter-class and intra-class bias.

Attention mechanisms are currently gaining popularity because of their powerful ability in eliminating uninformative parts of the input to leverage the other parts having a more useful information. Recently, attention mechanism has been integrated into typical CNN models at every individual layer to strengthen the intermediate outputs of each layer, in turn improving the final predictions for recognition in images. This model is usually trained with a weakly supervised method, however, this optimization method may lead to sub-optimal weights in the attention module. Hence, [212] proposed to train attention module by deep Q-learning with an LSTM model is trained to predict the reward, the whole process is called Deep REinforced Attention Learning (DREAL).

Various works specified here have been summarised and compared in Table 9 and general implementation of a DRL method to control an agents movement in an environment has been shown in fig 17 where state consists of an image frame provided by the environment, the DRL agent predicts actions to move the agent in the environment providing next state and the reward is provided by the environment, for example, [142].

12 Future Perspectives

12.1 Challenge Discussion

DRL is a powerful framework, which has been successfully applied to various computer vision applications including landmark detection, object detection, object tracking, image registration, image segmentation, video analysis, and other computer vision applications.

DRL has also demonstrated to be an effective alternative for solving difficult optimization problems, including tuning parameters, selecting augmentation strategies, and neural architecture search (NAS). However, most approaches, that we have reviewed, assume a stationary environment, from which observations are made. Take landmark detection as an instance, the environment takes into account the image itself, and each state is defined as an image patch consisting of the landmark location. In such a case, the environment is known while the RL/DRL framework naturally accommodates a dynamic environment, that is the environment itself evolves with the state and action. Realizing the full potential of DRL for computer vision requires solving several challenges. In this section, we would like to discuss the challenges of DRL in computer vision for real-world systems.

- **Reward function:** In most real-world applications, it is hard to define a specified reward function because it requires the knowledge from different domains that may not always be available. Thus, the intermediate rewards at each time step are not always easily computed. Furthermore, a reward function with too long delay will make training difficult. In contrast, assigning a reward for each action requires careful and manual human design.

- **Continuous state and action space:** Training an RL system on a continuous state and action space is challenging because most RL algorithms, i.e. Q learning, can only deal with discrete states and discrete action space. To address this limitation, most existing works discretize the continuous state and action space.

- **High-dimensional state and action space**: Training Q-function on a high-dimensional action space is challenging. For this reason, existing works use low-dimensional parameterization, whose dimensions are typically less than 10 with an exception [184] that uses 15-D and 25-D to model 2D and 3D registration, respectively.

- **Environment is complicated:** Almost all real-world systems, where we would want to deploy DRL/RL, are partially observable and non-stationary. Currently, the approaches we have reviewed assume a stationary environment, from which observations are made. However, the DRL/RL framework naturally accommodates dynamic environment, that is the environment itself evolves with the state and action. Furthermore, those systems are often stochastic and noisy (action delay, sensor and action noise) as compared to most simulated environments.

- **Training data requirement:** RL/DRL requires a large amount of training data or expert demonstrations. Large-scale datasets with annotations are expensive and hard to come by.

More details of challenges that embody difficulties to deploy RL/DRL in the real world are discussed in [82]. In this work, they designed a set of experiments and analyzed their effects on common RL agents. Open-sourcing an environmental suite, realworldrl-suite [83] is provided in this work as well.

12.2 DRL Recent Advances

Some advanced DRL approaches such as Inverse DRL, Multi-agent DRL, Meta DRL, and imitation learning are worth the attention and may promote new insights for many machine learning and computer vision tasks.

- **Inverse DRL:** DRL has been successfully applied into domains where the reward function is clearly defined. However, this is limited in real-world applications because it requires knowledge from different domains that may not always be available. Inverse DRL is one of the special cases of imitation learning. An example is autonomous driving, the reward function should be based on all factors such as driver's behavior, gas consumption, time, speed, safety, driving quality, etc. In real-world scenario, it is exhausting and hard to control all these factors. Different from DRL, inverse DRL [278], [4], [413], [86] a specific form of imitation learning [286], infers the reward function of an agent, given its policy or observed behavior, thereby avoiding a manual specification of its reward function. In the same problem of autonomous driving, inverse RL first uses a dataset collected from the human-generated driving and then approximates the reward function. Inverse RL has been successfully applied to many domains [4]. Recently, to analyze complex human movement and control high-dimensional robot systems, [215] proposed an online inverse RL algorithm. [2] combined both RL and Inverse RL to address planning problems in autonomous driving.

- **Multi-Agent DRL:** Most of the successful DRL applications such as game[38], [376], robotics[181], and autonomous driving [335], stock trading [206], social science [207], etc., involve multiple players that requires a model with multi-agent. Take autonomous driving as an instance, multi-agent DRL addresses the sequential decision-making problem which involves many autonomous agents, each of which aims to optimize its own utility return by interacting with the environment and other agents [40]. Learning in a multi-agent scenario is more difficult than a single-agent scenario because non-stationarity [135], multi-dimensionality [40], credit assignment [5], etc., depend on the multi-agent DRL approach of either fully cooperative or fully competitive. The agents can either collaborate to optimize a long-term utility or compete so that the utility is summed to zero. Recent work on Multi-Agent RL pays attention to learning new criteria or new setup [348].

- **Meta DRL:** As aforementioned, DRL algorithms consume large amounts of experience in order to learn an individual task and are unable to generalize the learned policy to newer problems. To alleviate the data challenge, Meta-RL algorithms [330], [380] are studied to enable agents to learn new skills from small amounts of experience. Recently, there is a research interest in meta RL [271], [119], [322], [303], [229], each using a different approach. For benchmarking and evaluation of meta RL algorithms, [415] presented Meta-world, which is an open-source simulator consisting of 50 distinct robotic manipulation tasks.

- **Imitation Learning:** Imitation learning is close to learning from demonstrations which aims at training a policy to mimic an expert's behavior given the samples collected from that expert. Imitation learning is also considered as an alternative to RL/DRL to solve sequential decision-making problems. Besides inverse DRL, an imitation learning approach as aforementioned, behavior cloning is another imitation learning approach to train policy under supervised learning manner. Bradly et al. [347] presented a method for unsupervised third-person imitation learning to observe how other humans perform and infer the task. Building on top of Deep Deterministic Policy Gradients and Hindsight Experience Replay, Nair et al. [272] proposed behavior cloning Loss to increase imitating the demonstrations. Besides Q-learning, Generative Adversarial Imitation Learning [364] proposes P-GAIL that integrates imitation learning into the policy gradient framework. P-GAIL considers both smoothness and causal entropy in policy update by utilizing Deep P-Network [365].

Conclusion

Deep Reinforcement Learning (DRL) is nowadays the most popular technique for an artificial agent to learn closely optimal strategies by experiences. This paper aims to provide a state-of-the-art comprehensive survey of DRL applications to a variety of decision-making problems in the area of computer vision. In this work, we firstly provided a structured summarization of the theoretical foundations in Deep Learning (DL) including AutoEncoder (AE), Multi-Layer Perceptron (MLP), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN). We then continued to introduce key techniques in RL research including model-based methods (value functions, transaction models, policy search, return functions) and model-free methods (value-based, policy-based, and actor-critic). Main techniques in DRL were thirdly presented under two categories of model-based and model-free approaches. We fourthly surveyed the broad-ranging applications of DRL methods in solving problems affecting areas of computer vision, from landmark detection, object detection, object tracking, image registration, image segmentation, video analysis, and many other applications in the computer vision area. We finally discussed several challenges ahead of us in order to realize the full potential of DRL for computer vision. Some latest advanced DRL techniques were included in the last discussion.

References

[1] Model-based contextual policy search for data-efficient generalization of robot skills. *Artificial Intelligence*, 247:415 – 439, 2017.

[2] Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning. *Robotics and Autonomous Systems*, 114:1 – 18, 2019.

[3] Pieter Abbeel, Adam Coates, and Andrew Y. Ng. Autonomous helicopter aerobatics through apprenticeship learning. *The International Journal of Robotics Research*, 29(13):1608–1639, 2010.

[4] Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-First International Conference on Machine Learning*, pages 1–8. Association for Computing Machinery, 2004.

[5] Adrian K. Agogino and Kagan Tumer. Unifying temporal and structural credit assignment problems. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2*, page 980–987. IEEE Computer Society, 2004.

[6] Narges Ahmidi, Lingling Tao, Shahin Sefati, Yixin Gao, Colin Lea, Benjamin Bejar Haro, Luca Zappella, Sanjeev Khudanpur, René Vidal, and Gregory D Hager. A dataset and benchmarks for segmentation and recognition of gestures in robotic surgery. *IEEE Transactions on Biomedical Engineering*, 64(9):2025–2041, 2017.

[7] Walid Abdullah Al and Il Dong Yun. Partial policy-based reinforcement learning for anatomical landmark localization in 3d medical images. *IEEE transactions on medical imaging*, 2019.

[8] Walid Abdullah Al, Il Dong Yun, and Kyong Joon Lee. Reinforcement learning-based automatic diagnosis of acute appendicitis in abdominal ct. *arXiv preprint arXiv:1909.00617*, 2019.

[9] Stephan Alaniz. Deep reinforcement learning with model learning and monte carlo tree search in minecraft. In *Conference on Reinforcement Learning and Decision Making*, 2018.

[10] Amir Alansary, Ozan Oktay, Yuanwei Li, Loic Le Folgoc, Benjamin Hou, Ghislain Vaillant, Konstantinos Kamnitsas, Athanasios Vlontzos, Ben Glocker, Bernhard Kainz, et al. Evaluating reinforcement learning agents for anatomical landmark detection. *Medical image analysis*, 53:156–164, 2019.

[11] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1):1–18, 2015.

[12] O. Andersson, F. Heintz, and P. Doherty. Model-based reinforcement learning in continuous environments using real-time constrained optimization. In *AAAI*, 2015.

[13] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. A brief survey of deep reinforcement learning. *arXiv preprint arXiv:1708.05866*, 2017.

[14] S Avinash Ramakanth and R Venkatesh Babu. Seamseg: Video object segmentation using patch seams. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 376–383, 2014.

[15] Morgane Ayle, Jimmy Tekli, Julia El-Zini, Boulos El-Asmar, and Mariette Awad. Bar-a reinforcement learning agent for bounding-box automated refinement.

[16] Mohammad Babaeizadeh, Iuri Frosio, Stephen Tyree, Jason Clemons, and Jan Kautz. GA3C: gpu-based A3C for deep reinforcement learning. *CoRR*, abs/1611.06256, 2016.

[17] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Visual tracking with online multiple instance learning. In *2009 IEEE conference on computer vision and pattern recognition*, pages 983–990. IEEE, 2009.

[18] Seung-Hwan Bae and Kuk-Jin Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1218–1225, 2014.

[19] Seung-Hwan Bae and Kuk-Jin Yoon. Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):595–610, 2017.

[20] J. Bagnell. Learning decision: Robustness, uncertainty, and approximation. 04 2012.

[21] J. A. Bagnell and J. G. Schneider. Autonomous helicopter control using reinforcement learning policy search methods. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, volume 2, pages 1615–1620, 2001.

[22] Neil Barakat, A Nicholas Hone, and Thomas E Darcie. Minimal-bracketing sets for high-dynamic-range image capture. *IEEE Transactions on Image Processing*, 17(10):1864–1875, 2008.

[23] Jonathan T Barron. A general and adaptive robust loss function. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4331–4339, 2019.

[24] Cher Bass, Pyry Helkkula, Vincenzo De Paola, Claudia Clopath, and Anil Anthony Bharath. Detection of axonal synapses in 3d two-photon images. *PloS one*, 12(9):e0183309, 2017.

[25] Miriam Bellver, Xavier Giró-i Nieto, Ferran Marqués, and Jordi Torres. Hierarchical object detection with deep reinforcement learning. *arXiv preprint arXiv:1611.03718*, 2016.

[26] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Networks*, 5(2):157–166, 1994.

[27] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger. Mvtec ad — a comprehensive real-world dataset for unsupervised anomaly detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9584–9592, 2019.

[28] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008:1–10, 2008.

[29] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016.

[30] Shalabh Bhatnagar. An actor–critic algorithm with function approximation for discounted cost constrained markov decision processes. *Systems & Control Letters*, 59(12):760–766, 2010.

[31] Shalabh Bhatnagar, Richard S. Sutton, Mohammad Ghavamzadeh, and Mark Lee. Natural actorâ-critic algorithms. *Automatica*, 45(11):2471 – 2482, 2009.

[32] Michael J Black and Yaser Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. In *Proceedings of IEEE international conference on computer vision*, pages 374–381. IEEE, 1995.

[33] N Bloch, A Madabhushi, H Huisman, J Freymann, J Kirby, M Grauer, A Enquobahrie, C Jaffe, L Clarke, and K Farahani. Nci-isbi 2013 challenge: automated segmentation of prostate structures. *The Cancer Imaging Archive*, 370, 2015.

[34] J. Boedecker, J. T. Springenberg, J. Wülfing, and M. Riedmiller. Approximate real-time optimal control based on sparse gaussian process models. In *2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, pages 1–8, 2014.

[35] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, Seoul, South Korea, 2019.

[36] Gustav Bredell, Christine Tanner, and Ender Konukoglu. Iterative interaction training for segmentation editing networks. In *International Workshop on Machine Learning in Medical Imaging*, pages 363–370. Springer, 2018.

[37] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[38] Noam Brown and Tuomas Sandholm. Superhuman ai for multiplayer poker. *Science*, 365(6456):885–890, 2019.

[39] Antoine Buetti-Dinh, Vanni Galli, SÃ¶ren Bellenberg, Olga Ilie, Malte Herold, Stephan Christel, Mariia Boretska, Igor V. Pivkin, Paul Wilmes, Wolfgang Sand, Mario Vera, and Mark Dopson. Deep neural networks outperform human expert's capacity in characterizing bioleaching bacterial biofilm composition. *Biotechnology Reports*, 22:e00321, 2019.

[40] L. Busoniu, R. Babuska, and B. De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

[41] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 221–230, 2017.

[42] Yunliang Cai, Said Osman, Manas Sharma, Mark Landis, and Shuo Li. Multi-modality vertebra recognition in arbitrary views using 3d deformable hierarchical model. *IEEE transactions on medical imaging*, 34(8):1676–1693, 2015.

[43] Juan C Caicedo and Svetlana Lazebnik. Active object localization with deep reinforcement learning. In *Proceedings of the IEEE international conference on computer vision*, pages 2488–2496, 2015.

[44] D. Carrera, F. Manganini, G. Boracchi, and E. Lanzarone. Defect detection in sem images of nanofibrous materials. *IEEE Transactions on Industrial Informatics*, 13(2):551–561, 2017.

[45] Boyu Chen, Dong Wang, Peixia Li, Shuang Wang, and Huchuan Lu. Real-time'actor-critic'tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 318–334, 2018.

[46] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[47] Yushi Chen, Xing Zhao, and Xiuping Jia. Spectral–spatial classification of hyperspectral data based on deep belief network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(6):2381–2392, 2015.

[48] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. Segflow: Joint learning for video object segmentation and optical flow. In *Proceedings of the IEEE international conference on computer vision*, pages 686–695, 2017.

[49] Ming-Ming Cheng, Niloy J Mitra, Xiaolei Huang, Philip HS Torr, and Shi-Min Hu. Global contrast based salient region detection. *IEEE transactions on pattern analysis and machine intelligence*, 37(3):569–582, 2014.

[50] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[51] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.

[52] Jongwon Choi, Hyung Jin Chang, Sangdoo Yun, Tobias Fischer, Yiannis Demiris, and Jin Young Choi. Attentional correlation filter network for adaptive visual tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4807–4816, 2017.

[53] Wongun Choi. Near-online multi-target tracking with aggregated local flow descriptor. In *Proceedings of the IEEE international conference on computer vision*, pages 3029–3037, 2015.

[54] Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, KyungHyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. *CoRR*, abs/1506.07503, 2015.

[55] Qi Chu, Wanli Ouyang, Hongsheng Li, Xiaogang Wang, Bin Liu, and Nenghai Yu. Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4836–4845, 2017.

[56] Wen-Hsuan Chu and Kris M. Kitani. Neural batch sampling with reinforcement learning for semi-supervised anomaly detection. In *European Conference on Computer Vision*, pages 751–766, 2020.

[57] Wen-Sheng Chu, Yale Song, and Alejandro Jaimes. Video co-summarization: Video summarization by visual co-occurrence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3584–3592, 2015.

[58] Ignasi Clavera, Jonas Rothfuss, John Schulman, Yasuhiro Fujita, Tamim Asfour, and Pieter Abbeel. Model-based reinforcement learning via meta-policy optimization. *CoRR*, abs/1809.05214, 2018.

[59] Adam Coates, Pieter Abbeel, and Andrew Y. Ng. Apprenticeship learning for helicopter control. *Commun. ACM*, 52(7):97–105, July 2009.

[60] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-time tracking of non-rigid objects using mean shift. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 2, pages 142–149. IEEE, 2000.

[61] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[62] Rémi Coulom. Efficient selectivity and backup operators in monte-carlo tree search. In *Proceedings of the 5th International Conference on Computers and Games*, page 72–83, 2006.

[63] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. 2016.

[64] Antonio Criminisi, Jamie Shotton, Duncan Robertson, and Ender Konukoglu. Regression forests for efficient anatomy detection and localization in ct studies. In *International MICCAI Workshop on Medical Computer Vision*, pages 106–117. Springer, 2010.

[65] Tianhong Dai, Magda Dubois, Kai Arulkumaran, Jonathan Campbell, Cher Bass, Benjamin Billot, Fatmatulzehra Uslu, Vincenzo De Paola, Claudia Clopath, and Anil Anthony Bharath. Deep reinforcement learning for subpixel neural tracking. In *International Conference on Medical Imaging with Deep Learning*, pages 130–150, 2019.

[66] Martin Danelljan, Goutam Bhat, Fahad Shahbaz Khan, and Michael Felsberg. Eco: Efficient convolution operators for tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6638–6646, 2017.

[67] Martin Danelljan, Gustav Hager, Fahad Shahbaz Khan, and Michael Felsberg. Learning spatially regularized correlation filters for visual tracking. In *Proceedings of the IEEE international conference on computer vision*, pages 4310–4318, 2015.

[68] Kristopher De Asis, J Fernando Hernandez-Garcia, G Zacharias Holland, and Richard S Sutton. Multi-step reinforcement learning: A unifying algorithm. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[69] Sandra Eliza Fontes De Avila, Ana Paula Brandão Lopes, Antonio da Luz Jr, and Arnaldo de Albuquerque Araújo. Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters*, 32(1):56–68, 2011.

[70] Antonio de Marvao, Timothy JW Dawes, Wenzhe Shi, Christopher Minas, Niall G Keenan, Tamara Diamond, Giuliana Durighel, Giovanni Montana, Daniel Rueckert, Stuart A Cook, et al. Population-based studies of myocardial hypertrophy: high resolution cardiovascular magnetic resonance atlases improve statistical power. *Journal of cardiovascular magnetic resonance*, 16(1):16, 2014.

[71] M. P. Deisenroth, P. Englert, J. Peters, and D. Fox. Multi-task policy search for robotics. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3876–3881, 2014.

[72] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[73] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[74] Joachim Denzler and Dietrich WR Paulus. Active motion detection and object tracking. In *Proceedings of 1st International Conference on Image Processing*, volume 3, pages 635–639. IEEE, 1994.

[75] B. Depraetere, M. Liu, G. Pinte, I. Grondman, and R. BabuÅ¡ka. Comparison of model-free and model-based methods for time optimal hit control of a badminton robot. *Mechatronics*, 24(8):1021 – 1030, 2014.

[76] Robert DiPietro, Colin Lea, Anand Malpani, Narges Ahmidi, S Swaroop Vedula, Gyusung I Lee, Mija R Lee, and Gregory D Hager. Recognizing surgical activities with recurrent neural networks. In *International conference on medical image computing and computer-assisted intervention*, pages 551–558. Springer, 2016.

[77] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: A benchmark. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 304–311. IEEE, 2009.

[78] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. *CoRR*, abs/1411.4389, 2014.

[79] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015.

[80] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. *arXiv preprint arXiv:1711.03938*, 2017.

[81] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1110–1118, 2015.

[82] Gabriel Dulac-Arnold, Nir Levine, Daniel J. Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. An empirical investigation of the challenges of real-world reinforcement learning, 2020.

[83] Gabriel Dulac-Arnold, Nir Levine, Daniel J. Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. An empirical investigation of the challenges of real-world reinforcement learning. 2020.

[84] Matteo Dunnhofer, Niki Martinel, Gian Luca Foresti, and Christian Micheloni. Visual tracking by means of deep reinforcement learning and an expert demonstrator. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[85] Chi Nhan Duong, Kha Gia Quach, Ibsa Jalata, Ngan Le, and Khoa Luu. Mobiface: A lightweight deep learning face recognition on mobile devices. In *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–6. IEEE, 2019.

[86] Chi Nhan Duong, Kha Gia Quach, Khoa Luu, T. Hoang Le, Marios Savvides, and Tien D. Bui. Learning from longitudinal face demonstration–where tractable deep modeling meets inverse reinforcement learning. 127(6–7), 2019.

[87] A. El-Fakdi and M. Carreras. Policy gradient based reinforcement learning for real autonomous underwater cable tracking. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3635–3640, 2008.

[88] Ehsan Elhamifar, Guillermo Sapiro, and Rene Vidal. See all by looking at a few: Sparse modeling for finding representative objects. In *2012 IEEE conference on computer vision and pattern recognition*, pages 1600–1607. IEEE, 2012.

[89] Dumitru Erhan, Christian Szegedy, Alexander Toshev, and Dragomir Anguelov. Scalable object detection using deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2147–2154, 2014.

[90] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge 2007 (voc2007) results. 2007.

[91] Mark Everingham and John Winn. The pascal visual object classes challenge 2012 (voc2012) development kit. *Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep*, 8, 2011.

[92] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5374–5383, 2019.

[93] Heng Fan and Haibin Ling. Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5486–5494, 2017.

[94] Jialue Fan, Wei Xu, Ying Wu, and Yihong Gong. Human tracking using convolutional neural networks. *IEEE Transactions on Neural Networks*, 21(10):1610–1623, 2010.

[95] Chelsea Finn, Xin Yu Tan, Yan Duan, Trevor Darrell, Sergey Levine, and Pieter Abbeel. Deep spatial autoencoders for visuomotor learning. In Danica Kragic, Antonio Bicchi, and Alessandro De Luca, editors, *2016 IEEE International Conference on Robotics and Automation, ICRA 2016, Stockholm, Sweden, May 16-21, 2016*, pages 512–519.

[96] J Michael Fitzpatrick and Jay B West. The distribution of target registration error in rigid-body point-based registration. *IEEE transactions on medical imaging*, 20(9):917–927, 2001.

[97] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G Bellemare, and Joelle Pineau. An introduction to deep reinforcement learning. *arXiv preprint arXiv:1811.12560*, 2018.

[98] Yaroslav Ganin, Tejas Kulkarni, Igor Babuschkin, SM Eslami, and Oriol Vinyals. Synthesizing programs for images using reinforced adversarial learning. *arXiv preprint arXiv:1804.01118*, 2018.

[99] Mingfei Gao, Ruichi Yu, Ang Li, Vlad I Morariu, and Larry S Davis. Dynamic zoom-in network for fast object detection in large images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6926–6935, 2018.

[100] Yixin Gao, S Swaroop Vedula, Carol E Reiley, Narges Ahmidi, Balakrishnan Varadarajan, Henry C Lin, Lingling Tao, Luca Zappella, Benjamın Béjar, David D Yuh, et al. Jhu-isi gesture and skill assessment working set (jigsaws): A surgical activity dataset for human motion modeling. In *Miccai workshop: M2cai*, volume 3, page 3, 2014.

[101] Romane Gauriau, Rémi Cuingnet, David Lesage, and Isabelle Bloch. Multi-organ localization combining global-to-local regression and confidence maps. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 337–344. Springer, 2014.

[102] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[103] Michaël Gharbi, Jiawen Chen, Jonathan T Barron, Samuel W Hasinoff, and Frédo Durand. Deep bilateral learning for real-time image enhancement. *ACM Transactions on Graphics (TOG)*, 36(4):1–12, 2017.

[104] Florin C Ghesu, Edward Krubasik, Bogdan Georgescu, Vivek Singh, Yefeng Zheng, Joachim Hornegger, and Dorin Comaniciu. Marginal space deep learning: efficient architecture for volumetric image parsing. *IEEE transactions on medical imaging*, 35(5):1217–1228, 2016.

[105] Florin-Cristian Ghesu, Bogdan Georgescu, Yefeng Zheng, Sasa Grbic, Andreas Maier, Joachim Hornegger, and Dorin Comaniciu. Multi-scale deep reinforcement learning for real-time 3d-landmark detection in ct scans. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):176–189, 2017.

[106] M Giles. Mit technology review. *Google researchers have reportedly achieved" quantum supremacy" URL: https:/www.technologyreview. com/f*, 614416, 2017.

[107] Justin Girard and M Reza Emami. Concurrent markov decision processes for robot team learning. *Engineering applications of artificial intelligence*, 39:223–234, 2015.

[108] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[109] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[110] Georgia Gkioxari, Ross Girshick, and Jitendra Malik. Contextual action recognition with r* cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1080–1088, 2015.

[111] Vikash Goel, Jameson Weng, and Pascal Poupart. Unsupervised video object segmentation for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 5683–5694, 2018.

[112] Abel Gonzalez-Garcia, Alexander Vezhnevets, and Vittorio Ferrari. An active search strategy for efficient object class detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3022–3031, 2015.

[113] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[114] Leo Grady. Random walks for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 28(11):1768–1783, 2006.

[115] Alex Graves, Abdel-rahman Mohamed, and Geoffrey E. Hinton. Speech recognition with deep recurrent neural networks. *CoRR*, abs/1303.5778, 2013.

[116] Albert Gubern-Mérida, Robert Martí, Jaime Melendez, Jakob L Hauth, Ritse M Mann, Nico Karssemeijer, and Bram Platel. Automated localization of breast cancer in dce-mri. *Medical image analysis*, 20(1):265–274, 2015.

[117] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[118] Minghao Guo, Jiwen Lu, and Jie Zhou. Dual-agent deep reinforcement learning for deformable face tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 768–783, 2018.

[119] Abhishek Gupta, Russell Mendonca, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Meta-reinforcement learning of structured exploration strategies. In *Advances in Neural Information Processing Systems*, pages 5302–5311, 2018.

[120] Saurabh Gupta, Pablo Arbelaez, and Jitendra Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 564–571, 2013.

[121] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *European conference on computer vision*, pages 345–360. Springer, 2014.

[122] Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. Creating summaries from user videos. In *European conference on computer vision*, pages 505–520. Springer, 2014.

[123] Seyed Hamid Rezatofighi, Anton Milan, Zhen Zhang, Qinfeng Shi, Anthony Dick, and Ian Reid. Joint probabilistic data association revisited. In *Proceedings of the IEEE international conference on computer vision*, pages 3047–3055, 2015.

[124] Junwei Han, Le Yang, Dingwen Zhang, Xiaojun Chang, and Xiaodan Liang. Reinforcement cutting-agent learning for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9080–9089, 2018.

[125] Robert M Haralick and Linda G Shapiro. Image segmentation techniques. *Computer vision, graphics, and image processing*, 29(1):100–132, 1985.

[126] Sam Hare, Stuart Golodetz, Amir Saffari, Vibhav Vineet, Ming-Ming Cheng, Stephen L Hicks, and Philip HS Torr. Struck: Structured output tracking with kernels. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2096–2109, 2015.

[127] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision*, pages 297–312. Springer, 2014.

[128] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 447–456, 2015.

[129] Hannes Hase, Mohammad Farid Azampour, Maria Tirindelli, Magdalini Paschali, Walter Simson, Emad Fatemizadeh, and Nassir Navab. Ultrasound-guided robotic navigation with deep reinforcement learning. *arXiv preprint arXiv:2003.13321*, 2020.

[130] Hado V Hasselt. Double q-learning. In *Advances in neural information processing systems*, pages 2613–2621, 2010.

[131] Matthew J. Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. *CoRR*, abs/1507.06527, 2015.

[132] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[133] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[134] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE transactions on pattern analysis and machine intelligence*, 37(3):583–596, 2014.

[135] Pablo Hernandez-Leal, Michael Kaisers, Tim Baarslag, and Enrique Munoz de Cote. A survey of learning in multiagent environments: Dealing with non-stationarity. *CoRR*, abs/1707.09183, 2017.

[136] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. *arXiv preprint arXiv:1710.02298*, 2017.

[137] Todd Hester, Michael Quinlan, and Peter Stone. A real-time model-based reinforcement learning architecture for robot control. *CoRR*, abs/1105.1749, 2011.

[138] Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The goldilocks principle: Reading children's books with explicit memory representations. *CoRR*, abs/1511.02301, 2015.

[139] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[140] Derek Hoiem, Alexei A Efros, and Martial Hebert. Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1):151–172, 2007.

[141] James B. Holliday and Ngan T.H. Le. Follow then forage exploration: Improving asynchronous advantage actor critic. *International Conference on Soft Computing, Artificial Intelligence and Applications (SAI 2020)*, pages 107–118, 2020.

[142] Zhang-Wei Hong, Chen Yu-Ming, Shih-Yang Su, Tzu-Yun Shann, Yi-Hsiang Chang, Hsuan-Kung Yang, Brian Hsi-Lin Ho, Chih-Chieh Tu, Yueh-Chuan Chang, Tsu-Ching Hsiao, et al. Virtual-to-real: Learning to control in visual semantic segmentation. *arXiv preprint arXiv:1802.00285*, 2018.

[143] Ju Hong Yoon, Chang-Ryeol Lee, Ming-Hsuan Yang, and Kuk-Jin Yoon. Online multi-object tracking via structural constraint event aggregation. In *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pages 1392–1400, 2016.

[144] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.

[145] Jian-Fang Hu, Wei-Shi Zheng, Jianhuang Lai, and Jianguo Zhang. Jointly learning heterogeneous features for rgb-d activity recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5344–5352, 2015.

[146] Weiming Hu, Xi Li, Wenhan Luo, Xiaoqin Zhang, Stephen Maybank, and Zhongfei Zhang. Single and multiple object tracking using log-euclidean riemannian subspace and block-division appearance model. *IEEE transactions on pattern analysis and machine intelligence*, 34(12):2420–2440, 2012.

[147] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[148] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[149] Zhewei Huang, Wen Heng, and Shuchang Zhou. Learning to paint with model-based deep reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8709–8718, 2019.

[150] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[151] Zhiwu Huang, Chengde Wan, Thomas Probst, and Luc Van Gool. Deep learning on lie groups for skeleton-based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6099–6108, 2017.

[152] Gabriel Efrain Humpire-Mamani, Arnaud Arindra Adiyoso Setio, Bram van Ginneken, and Colin Jacobs. Efficient organ localization using multi-label convolutional neural networks in thorax-abdomen ct scans. *Physics in Medicine & Biology*, 63(8):085003, 2018.

[153] Luis Ibanez, Will Schroeder, Lydia Ng, and Josh Cates. The itk software guide: updated for itk version 2.4, 2005.

[154] Haroon Idrees, Imran Saleemi, Cody Seibert, and Mubarak Shah. Multi-source multi-scale counting in extremely dense crowd images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2547–2554, 2013.

[155] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Maadeed, Nasir Rajpoot, and Mubarak Shah. Composition loss for counting, density map estimation and localization in dense crowds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–546, 2018.

[156] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017.

[157] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[158] Clifford R Jack Jr, Matt A Bernstein, Nick C Fox, Paul Thompson, Gene Alexander, Danielle Harvey, Bret Borowski, Paula J Britson, Jennifer L. Whitwell, Chadwick Ward, et al. The alzheimer's disease neuroimaging initiative (adni): Mri methods. *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 27(4):685–691, 2008.

[159] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[160] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Deep features for text spotting. In *European conference on computer vision*, pages 512–528. Springer, 2014.

[161] Arjit Jain, Alexander Powers, and Hans J Johnson. Robust automatic multiple landmark detection. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, pages 1178–1182. IEEE, 2020.

[162] Suyog Dutt Jain and Kristen Grauman. Supervoxel-consistent foreground propagation in video. In *European conference on computer vision*, pages 656–671. Springer, 2014.

[163] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pages 2117–2126. IEEE, 2017.

[164] Varun Jampani, Raghudeep Gadde, and Peter V Gehler. Video propagation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 451–461, 2017.

[165] Won-Dong Jang and Chang-Su Kim. Online video object segmentation via convolutional trident network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5849–5858, 2017.

[166] Simon Jégou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 11–19, 2017.

[167] Zeyu Jia, Lin Yang, Csaba Szepesvari, and Mengdi Wang. Model-based reinforcement learning with value-targeted regression. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, volume 120 of *Proceedings of Machine Learning Research*, pages 666–686, The Cloud, 10–11 Jun 2020.

[168] Ming-xin Jiang, Chao Deng, Zhi-geng Pan, Lan-fang Wang, and Xing Sun. Multiobject tracking in videos based on lstm and deep reinforcement learning. *Complexity*, 2018, 2018.

[169] Mingxin Jiang, Tao Hai, Zhigeng Pan, Haiyan Wang, Yinjie Jia, and Chao Deng. Multi-agent deep reinforcement learning for multi-object tracker. *IEEE Access*, 7:32400–32407, 2019.

[170] Zequn Jie, Xiaodan Liang, Jiashi Feng, Xiaojie Jin, Wen Lu, and Shuicheng Yan. Tree-structured reinforcement learning for sequential object localization. In *Advances in Neural Information Processing Systems*, pages 127–135, 2016.

[171] Oscar Jimenez-del Toro, Henning Müller, Markus Krenn, Katharina Gruenberg, Abdel Aziz Taha, Marianne Winterstein, Ivan Eggel, Antonio Foncubierta-Rodríguez, Orcun Goksel, András Jakab, et al. Cloud-based evaluation of anatomical structure segmentation and landmark detection algorithms: Visceral anatomy benchmarks. *IEEE transactions on medical imaging*, 35(11):2459–2475, 2016.

[172] V Craig Jordan. Long-term adjuvant tamoxifen therapy for breast cancer. *Breast cancer research and treatment*, 15(3):125–136, 1990.

[173] Yeong Jun Koh and Chang-Su Kim. Primary object segmentation in videos based on region augmentation and reduction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3442–3450, 2017.

[174] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *IEEE transactions on pattern analysis and machine intelligence*, 34(7):1409–1422, 2011.

[175] Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. Association for Computational Linguistics, October 2013.

[176] Michał Kempka, Marek Wydmuch, Grzegorz Runc, Jakub Toczek, and Wojciech Jaśkowski. Vizdoom: A doom-based ai research platform for visual reinforcement learning. In *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8. IEEE, 2016.

[177] Du Yong Kim and Moongu Jeon. Data fusion of radar and image measurements for multi-object tracking via kalman filtering. *Information Sciences*, 278:641–652, 2014.

[178] Kye Kyung Kim, Soo Hyun Cho, Hae Jin Kim, and Jae Yeon Lee. Detecting and tracking moving object using an active camera. In *The 7th International Conference on Advanced Communication Technology, 2005, ICACT 2005.*, volume 2, pages 817–820. IEEE, 2005.

[179] Donna Kirwan. Nhs fetal anomaly screening programme. *National Standards and Guidance for England*, 18(0), 2010.

[180] Stefan Klein, Marius Staring, Keelin Murphy, Max A Viergever, and Josien PW Pluim. Elastix: a toolbox for intensity-based medical image registration. *IEEE transactions on medical imaging*, 29(1):196–205, 2009.

[181] Jens Kober, J. Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[182] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.

[183] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011.

[184] Julian Krebs, Tommaso Mansi, Hervé Delingette, Li Zhang, Florin C Ghesu, Shun Miao, Andreas K Maier, Nicholas Ayache, Rui Liao, and Ali Kamen. Robust non-rigid registration through agent-based action learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 344–352. Springer, 2017.

[185] Matej Kristan, Ales Leonardis, Jiri Matas, Michael Felsberg, Roman Pflugfelder, Luka Cehovin Zajc, Tomas Vojir, Goutam Bhat, Alan Lukezic, Abdelrahman Eldesokey, et al. The sixth visual object tracking vot2018 challenge results. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.

[186] Matej Kristan, Jiri Matas, Ales Leonardis, Michael Felsberg, Luka Cehovin, Gustavo Fernandez, Tomas Vojir, Gustav Hager, Georg Nebehay, and Roman Pflugfelder. The visual object tracking vot2015 challenge results. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 1–23, 2015.

[187] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[188] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[189] A. Kupcsik, M. Deisenroth, Jan Peters, and G. Neumann. Data-efficient generalization of robot skills with contextual policy search. In *AAAI*, 2013.

[190] Thanard Kurutach, Ignasi Clavera, Yan Duan, Aviv Tamar, and Pieter Abbeel. Model-ensemble trust-region policy optimization. 02 2018.

[191] Ngan Le, Trung Le, Kashu Yamazaki, Toan Duc Bui, Khoa Luu, and Marios Savides. Offset curves loss for imbalanced problem in medical segmentation. *arXiv preprint arXiv:2012.02463*, 2020.

[192] Ngan Le, Kashu Yamazaki, Dat Truong, Kha Gia Quach, and Marios Savvides. A multi-task contextual atrous residual network for brain tumor detection & segmentation. *arXiv preprint arXiv:2012.02073*, 2020.

[193] T Hoang Ngan Le, Chi Nhan Duong, Ligong Han, Khoa Luu, Kha Gia Quach, and Marios Savvides. Deep contextual recurrent residual networks for scene labeling. *Pattern Recognition*, 80:32–41, 2018.

[194] T Hoang Ngan Le, Kha Gia Quach, Khoa Luu, Chi Nhan Duong, and Marios Savvides. Reformulating level sets as deep recurrent neural network approach to semantic segmentation. *IEEE Transactions on Image Processing*, 27(5):2393–2407, 2018.

[195] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 156–165, 2017.

[196] Colin Lea, Austin Reiter, René Vidal, and Gregory D Hager. Segmental spatiotemporal cnns for fine-grained action segmentation. In *European Conference on Computer Vision*, pages 36–52. Springer, 2016.

[197] Colin Lea, René Vidal, and Gregory D Hager. Learning convolutional action primitives for fine-grained action recognition. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 1642–1649. IEEE, 2016.

[198] Colin Lea, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks: A unified approach to action segmentation. In *European Conference on Computer Vision*, pages 47–54. Springer, 2016.

[199] Laura Leal-Taixé, Cristian Canton-Ferrer, and Konrad Schindler. Learning by tracking: Siamese cnn for robust target association. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 33–40, 2016.

[200] Laura Leal-Taixé, Michele Fenzi, Alina Kuznetsova, Bodo Rosenhahn, and Silvio Savarese. Learning an image-based motion context for multiple people tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3542–3549, 2014.

[201] Laura Leal-Taixé, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler. Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv preprint arXiv:1504.01942*, 2015.

[202] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[203] Yann LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–50. Springer, 1998.

[204] Yann LeCun, D Touresky, G Hinton, and T Sejnowski. A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school*, pages 21–28. CMU, Pittsburgh, Pa: Morgan Kaufmann, 1988.

[205] Hyunjae Lee, Hyo-Eun Kim, and Hyeonseob Nam. Srm: A style-based recalibration module for convolutional neural networks. pages 1854–1862, 10 2019.

[206] Jae Won Lee, Jonghun Park, Jangmin O, Jongwoo Lee, and Euyseok Hong. A multi-agent approach to q-learning for daily stock trading. *Trans. Sys. Man Cyber. Part A*, 37(6):864–877, November 2007.

[207] Joel Z. Leibo, Vinícius Flores Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. Multi-agent reinforcement learning in sequential social dilemmas. *CoRR*, abs/1702.03037, 2017.

[208] Sergey Levine and Vladlen Koltun. Learning complex neural network policies with trajectory optimization. In *Proceedings of the 31st International Conference on Machine Learning*, pages 829–837, 2014.

[209] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8971–8980, 2018.

[210] Buyu Li, Wanli Ouyang, Lu Sheng, Xingyu Zeng, and Xiaogang Wang. GS3D: an efficient 3d object detection framework for autonomous driving. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 1019–1028. Computer Vision Foundation / IEEE, 2019.

[211] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. *arXiv preprint arXiv:1804.06055*, 2018.

[212] Duo Li and Qifeng Chen. Deep reinforced attention learning for quality-aware visual recognition. In *European Conference on Computer Vision*, pages 493–509, 2020.

[213] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. *arXiv preprint arXiv:1503.08663*, 2015.

[214] Jiwei Li, Minh-Thang Luong, and Dan Jurafsky. A hierarchical neural autoencoder for paragraphs and documents. *CoRR*, abs/1506.01057, 2015.

[215] K. Li, M. Rath, and J. W. Burdick. Inverse reinforcement learning via function approximation for clinical motion analysis. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 610–617, 2018.

[216] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

[217] Peixia Li, Dong Wang, Lijun Wang, and Huchuan Lu. Deep visual tracking: Review and experimental comparison. *Pattern Recognition*, 76:323–338, 2018.

[218] Yingbo Li and Bernard Merialdo. Multi-video summarization based on video-mmr. In *11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10*, pages 1–4. IEEE, 2010.

[219] Yuanwei Li, Amir Alansary, Juan J Cerrolaza, Bishesh Khanal, Matthew Sinclair, Jacqueline Matthew, Chandni Gupta, Caroline Knight, Bernhard Kainz, and Daniel Rueckert. Fast multiple landmark localisation using a patch-based iterative network. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 563–571. Springer, 2018.

[220] Pengpeng Liang, Erik Blasch, and Haibin Ling. Encoding color information for visual tracking: Algorithms and benchmark. *IEEE Transactions on Image Processing*, 24(12):5630–5644, 2015.

[221] Rui Liao, Shun Miao, Pierre de Tournemire, Sasa Grbic, Ali Kamen, Tommaso Mansi, and Dorin Comaniciu. An artificial agent for robust image registration. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[222] Xuan Liao, Wenhao Li, Qisen Xu, Xiangfeng Wang, Bo Jin, Xiaoyun Zhang, Yanfeng Wang, and Ya Zhang. Iteratively-refined interactive 3d medical image segmentation with multi-agent reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9394–9402, 2020.

[223] Timothy P. Lillicrap, Jonathan J. Hunt, Alexand er Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv e-prints*, page arXiv:1509.02971, September 2015.

[224] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[225] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[226] Tony Lindeberg. *Scale-space theory in computer vision*, volume 256. Springer Science & Business Media, 2013.

[227] Geert Litjens, Robert Toth, Wendy van de Ven, Caroline Hoeks, Sjoerd Kerkstra, Bram van Ginneken, Graham Vincent, Gwenael Guillard, Neil Birbeck, Jindang Zhang, et al. Evaluation of prostate segmentation algorithms for mri: the promise12 challenge. *Medical image analysis*, 18(2):359–373, 2014.

[228] Daochang Liu and Tingting Jiang. Deep reinforcement learning for surgical gesture segmentation and classification. In *International conference on medical image computing and computer-assisted intervention*, pages 247–255. Springer, 2018.

[229] Hao Liu, Richard Socher, and Caiming Xiong. Taming maml: Efficient unbiased meta-reinforcement learning. In *International Conference on Machine Learning*, pages 4061–4071, 2019.

[230] Liang Liu, Hao Lu, Hongwei Zou, Haipeng Xiong, Zhiguo Cao, and Chunhua Shen. Weighing counts: Sequential crowd counting by reinforcement learning. 2020.

[231] Lijie Liu, Chufan Wu, Jiwen Lu, Lingxi Xie, Jie Zhou, and Qi Tian. Reinforced axial refinement network for monocular 3d object detection. In *European Conference on Computer Vision ECCV*, pages 540–556, 2020.

[232] Lingbo Liu, Hongjun Wang, Guanbin Li, Wanli Ouyang, and Liang Lin. Crowd counting using deep recurrent spatial-aware network. *arXiv preprint arXiv:1807.00601*, 2018.

[233] Tianrui Liu, Qingjie Meng, Athanasios Vlontzos, Jeremy Tan, Daniel Rueckert, and Bernhard Kainz. Ultrasound video summarization using deep reinforcement learning. *arXiv preprint arXiv:2005.09531*, 2020.

[234] Weizhe Liu, Mathieu Salzmann, and Pascal Fua. Context-aware crowd counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5099–5108, 2019.

[235] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[236] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[237] Marco Lorenzi, Nicholas Ayache, Giovanni B Frisoni, Xavier Pennec, Alzheimer's Disease Neuroimaging Initiative (ADNI, et al. Lcc-demons: a robust and accurate symmetric diffeomorphic registration algorithm. *NeuroImage*, 81:470–483, 2013.

[238] Tayebeh Lotfi, Lisa Tang, Shawn Andrews, and Ghassan Hamarneh. Improving probabilistic image registration via reinforcement learning and uncertainty evaluation. In *International Workshop on Machine Learning in Medical Imaging*, pages 187–194. Springer, 2013.

[239] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[240] Wenhan Luo, Peng Sun, Fangwei Zhong, Wei Liu, Tong Zhang, and Yizhou Wang. End-to-end active object tracking via reinforcement learning. *arXiv preprint arXiv:1705.10561*, 2017.

[241] Thang Luong, Ilya Sutskever, Quoc V. Le, Oriol Vinyals, and Wojciech Zaremba. Addressing the rare word problem in neural machine translation. *CoRR*, abs/1410.8206, 2014.

[242] Khoa Luu, Chenchen Zhu, Chandrasekhar Bhagavatula, T Hoang Ngan Le, and Marios Savvides. A deep learning approach to joint face detection and segmentation. In *Advances in Face Detection and Facial Image Analysis*, pages 1–12. Springer, 2016.

[243] Chao Ma, Jia-Bin Huang, Xiaokang Yang, and Ming-Hsuan Yang. Hierarchical convolutional features for visual tracking. In *Proceedings of the IEEE international conference on computer vision*, pages 3074–3082, 2015.

[244] Kai Ma, Jiangping Wang, Vivek Singh, Birgi Tamersoy, Yao-Jen Chang, Andreas Wimmer, and Terrence Chen. Multimodal image registration with deep context reinforcement learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 240–248. Springer, 2017.

[245] Behrooz Mahasseni, Michael Lam, and Sinisa Todorovic. Unsupervised video summarization with adversarial lstm networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 202–211, 2017.

[246] Gabriel Maicas, Gustavo Carneiro, Andrew P Bradley, Jacinto C Nascimento, and Ian Reid. Deep reinforcement learning for active breast lesion detection from dce-mri. In *International conference on medical image computing and computer-assisted intervention*, pages 665–673. Springer, 2017.

[247] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, and Alan L. Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *CoRR*, abs/1412.6632, 2014.

[248] Nicolas Märki, Federico Perazzi, Oliver Wang, and Alexander Sorkine-Hornung. Bilateral space video segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 743–751, 2016.

[249] T. Martinez-Marin and T. Duckett. Fast reinforcement learning for vision-guided mobile robots. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 4170–4175, 2005.

[250] Jan Matas, Stephen James, and Andrew J Davison. Sim-to-real reinforcement learning for deformable object manipulation. *arXiv preprint arXiv:1806.07851*, 2018.

[251] Stefan Mathe, Aleksis Pirinen, and Cristian Sminchisescu. Reinforcement learning for visual object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2894–2902, 2016.

[252] George K Matsopoulos, Nicolaos A Mouravliansky, Konstantinos K Delibasis, and Konstantina S Nikita. Automatic retinal image registration scheme using global optimization techniques. *IEEE Transactions on Information Technology in Biomedicine*, 3(1):47–60, 1999.

[253] Darryl McClymont, Andrew Mehnert, Adnan Trakic, Dominic Kennedy, and Stuart Crozier. Fully automatic lesion segmentation in breast mri using mean-shift and graph-cuts on a region adjacency graph. *Journal of Magnetic Resonance Imaging*, 39(4):795–804, 2014.

[254] Bjoern H Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE transactions on medical imaging*, 34(10):1993–2024, 2014.

[255] Shun Miao, Rui Liao, Marcus Pfister, Li Zhang, and Vincent Ordy. System and method for 3-d/3-d registration between non-contrast-enhanced cbct and contrast-enhanced ct for abdominal aortic aneurysm stenting. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 380–387. Springer, 2013.

[256] Shun Miao, Z Jane Wang, and Rui Liao. A cnn regression approach for real-time 2d/3d registration. *IEEE transactions on medical imaging*, 35(5):1352–1363, 2016.

[257] Tomas Mikolov, Stefan Kombrink, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Extensions of recurrent neural network language model. In *ICASSP*, pages 5528–5531, 2011.

[258] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

[259] Anton Milan, Laura Leal-Taixé, Konrad Schindler, and Ian Reid. Joint tracking and segmentation of multiple targets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5397–5406, 2015.

[260] Anton Milan, S Hamid Rezatofighi, Anthony Dick, Ian Reid, and Konrad Schindler. Online multi-target tracking using recurrent neural networks. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[261] Shervin Minaee, AmirAli Abdolrashidi, Hang Su, Mohammed Bennamoun, and David Zhang. Biometric recognition using deep learning: A survey. *CoRR*, abs/1912.00271, 2019.

[262] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

[263] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1928–1937, 20–22 Jun 2016.

[264] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[265] I. Mordatch, N. Mishra, C. Eppner, and P. Abbeel. Combining model-based policy search with online model learning for control of physical humanoids. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 242–248, 2016.

[266] J. Morimoto, G. Zeglin, and C. G. Atkeson. Minimax differential dynamic programming: application to a biped walking robot. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, volume 2, pages 1927–1932, 2003.

[267] Jun Morimoto and Christopher G. Atkeson. Nonparametric representation of an approximated poincaré map for learning biped locomotion. In *Autonomous Robots*, page 131–144, 2009.

[268] A. Mousavian, D. Anguelov, J. Flynn, and J. Košecká. 3d bounding box estimation using deep learning and geometry. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5632–5640, 2017.

[269] Matthias Mueller, Neil Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. In *European conference on computer vision*, pages 445–461. Springer, 2016.

[270] Don Murray and Anup Basu. Motion tracking with an active camera. *IEEE transactions on pattern analysis and machine intelligence*, 16(5):449–459, 1994.

[271] Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. *arXiv preprint arXiv:1803.11347*, 2018.

[272] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6292–6299, 2018.

[273] Hyeonseob Nam and Bohyung Han. Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4293–4302, 2016.

[274] Ali Bou Nassif, Ismail Shahin, Imtinan Attili, Mohammad Azzeh, and Khaled Shaalan. Speech recognition using deep neural networks: A systematic review. *IEEE Access*, 7:19143–19165, 2019.

[275] Fernando Navarro, Anjany Sekuboyina, Diana Waldmannstetter, Jan C Peeken, Stephanie E Combs, and Bjoern H Menze. Deep reinforcement learning for organ localization in ct. *arXiv preprint arXiv:2005.04974*, 2020.

[276] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[277] Dominik Neumann, Saša Grbić, Matthias John, Nassir Navab, Joachim Hornegger, and Razvan Ionasec. Probabilistic sparse matching for robust 3d/3d fusion in minimally invasive surgery. *IEEE transactions on medical imaging*, 34(1):49–60, 2014.

[278] Andrew Y. Ng and Stuart J. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, page 663–670, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[279] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000.

[280] Trung Thanh Nguyen, Zhuoru Li, Tomi Silander, and Tze-Yun Leong. Online feature selection for model-based reinforcement learning. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, page I–498–I–506, 2013.

[281] Chi Nhan Duong, Kha Gia Quach, Khoa Luu, Ngan Le, and Marios Savvides. Temporal non-volume preserving approach to facial age-progression and age-invariant face recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3735–3743, 2017.

[282] C. Niedzwiedz, I. Elhanany, Zhenzhen Liu, and S. Livingston. A consolidated actor-critic model with function approximation for high-dimensional pomdps. In *AAAI 2008Workshop for Advancement in POMDP Solvers*, 2008.

[283] Yishuang Ning, Sheng He, Zhiyong Wu, Chunxiao Xing, and Liang-Jie Zhang. A review of deep learning based speech synthesis. *Applied Sciences*, 9(19), 2019.

[284] Kenji Okuma, Ali Taleghani, Nando De Freitas, James J Little, and David G Lowe. A boosted particle filter: Multitarget detection and tracking. In *European conference on computer vision*, pages 28–39. Springer, 2004.

[285] José Ignacio Orlando, Huazhu Fu, João Barbosa Breda, Karel van Keer, Deepti R Bathula, Andrés Diaz-Pinto, Ruogu Fang, Pheng-Ann Heng, Jeyoung Kim, JoonHo Lee, et al. Refuge challenge: A unified framework for evaluating automated methods for glaucoma assessment from fundus photographs. *Medical image analysis*, 59:101570, 2020.

[286] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters. 2018.

[287] Rameswar Panda, Abir Das, Ziyan Wu, Jan Ernst, and Amit K Roy-Chowdhury. Weakly supervised summarization of web videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3657–3666, 2017.

[288] Anestis Papazoglou and Vittorio Ferrari. Fast object segmentation in unconstrained video. In *Proceedings of the IEEE international conference on computer vision*, pages 1777–1784, 2013.

[289] I. C. Paschalidis, K. Li, and R. Moazzez Estanjini. An actor-critic method using least squares temporal difference learning. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 2564–2569, 2009.

[290] Massimiliano Patacchiola and Angelo Cangelosi. Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods. *Pattern Recognition*, 71:132–143, 2017.

[291] Hanchuan Peng, Zongcai Ruan, Fuhui Long, Julie H Simpson, and Eugene W Myers. V3d enables real-time 3d visualization and quantitative analysis of large-scale biological image data sets. *Nature biotechnology*, 28(4):348–353, 2010.

[292] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2663–2672, 2017.

[293] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 724–732, 2016.

[294] Jan Peters and Stefan Schaal. Reinforcement learning of motor skills with policy gradients. *Neural Networks*, 21(4):682 – 697, 2008.

[295] Aleksis Pirinen and Cristian Sminchisescu. Deep reinforcement learning of region proposal networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6945–6954, 2018.

[296] Hamed Pirsiavash, Deva Ramanan, and Charless C Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR 2011*, pages 1201–1208. IEEE, 2011.

[297] Aske Plaat, Walter Kosters, and Mike Preuss. Deep model-based reinforcement learning for high-dimensional problems, a survey, 2020.

[298] Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid. Category-specific video summarization. In *European conference on computer vision*, pages 540–555. Springer, 2014.

[299] Reza Pourreza-Shahri and Nasser Kehtarnavaz. Exposure bracketing via automatic exposure selection. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 320–323. IEEE, 2015.

[300] Alessandro Prest, Christian Leistner, Javier Civera, Cordelia Schmid, and Vittorio Ferrari. Learning object class detectors from weakly annotated video. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3282–3289. IEEE, 2012.

[301] Yuankai Qi, Shengping Zhang, Lei Qin, Hongxun Yao, Qingming Huang, Jongwoo Lim, and Ming-Hsuan Yang. Hedged deep tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4303–4311, 2016.

[302] Zengyi Qin, Jinglu Wang, and Yan Lu. Monogrnet: A geometric reasoning network for monocular 3d object localization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):8851–8858, Jul. 2019.

[303] Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International conference on machine learning*, pages 5331–5340, 2019.

[304] Vidhiwar Singh Rathour, Kashu Yamakazi, and T Le. Roughness index and roughness distance for benchmarking medical segmentation. *arXiv preprint arXiv:2103.12350*, 2021.

[305] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[306] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[307] Liangliang Ren, Jiwen Lu, Zifeng Wang, Qi Tian, and Jie Zhou. Collaborative deep reinforcement learning for multi-object tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 586–602, 2018.

[308] Liangliang Ren, Xin Yuan, Jiwen Lu, Ming Yang, and Jie Zhou. Deep reinforcement learning with iterative shift for visual tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 684–700, 2018.

[309] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[310] Md Reza, Jana Kosecka, et al. Reinforcement learning for semantic segmentation in indoor scenes. *arXiv preprint arXiv:1606.01178*, 2016.

[311] Alexander Richard and Juergen Gall. Temporal action detection using a statistical language model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3131–3140, 2016.

[312] Mrigank Rochan, Linwei Ye, and Yang Wang. Video summarization using fully convolutional sequence networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 347–363, 2018.

[313] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[314] German Ros, Vladfen Koltun, Felipe Codevilla, and Antonio Lopez. The carla autonomous driving challenge, 2019.

[315] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. " grabcut" interactive foreground extraction using iterated graph cuts. *ACM transactions on graphics (TOG)*, 23(3):309–314, 2004.

[316] David Rotman. Mit technology review. *Retrieved from Meet the Man with a Cheap and Easy Plan to Stop Global Warming: http://www. technologyreview. com/featuredstor y/511016/a-cheap-and-easy-plan-to-stop-globalwarming*, 2013.

[317] J-M Rouet, J-J Jacq, and Christian Roux. Genetic algorithms for a robust 3-d mr-ct registration. *IEEE transactions on information technology in biomedicine*, 4(2):126–136, 2000.

[318] David E Rumelhart. The architecture of mind: A connectionist approach. *Mind readings*, pages 207–238, 1998.

[319] T. P. Runarsson and S. M. Lucas. Imitating play from game trajectories: Temporal difference learning versus preference learning. In *2012 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 79–82, 2012.

[320] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[321] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 300–311, 2017.

[322] Steindór Sæmundsson, Katja Hofmann, and Marc Peter Deisenroth. Meta reinforcement learning with latent variable gaussian processes. *arXiv preprint arXiv:1803.07551*, 2018.

[323] Farhang Sahba. Deep reinforcement learning for object segmentation in video sequences. In *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 857–860. IEEE, 2016.

[324] Farhang Sahba, Hamid R Tizhoosh, and Magdy MA Salama. A reinforcement learning framework for medical image segmentation. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pages 511–517. IEEE, 2006.

[325] Farhang Sahba, Hamid R Tizhoosh, and Magdy MMA Salama. Application of opposition-based reinforcement learning in image segmentation. In *2007 IEEE Symposium on Computational Intelligence in Image and Signal Processing*, pages 246–251. IEEE, 2007.

[326] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897, 2015.

[327] John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust Region Policy Optimization. *arXiv e-prints*, February 2015.

[328] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *arXiv e-prints*, July 2017.

[329] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[330] Nicolas Schweighofer and Kenji Doya. Meta-learning in reinforcement learning. *Neural Networks*, 16(1):5–9, 2003.

[331] Shahin Sefati, Noah J Cowan, and René Vidal. Learning shared, discriminative dictionaries for surgical gesture segmentation and classification. In *MICCAI Workshop: M2CAI*, volume 4, 2015.

[332] Mohammad Javad Shafiee, Brendan Chywl, Francis Li, and Alexander Wong. Fast yolo: A fast you only look once system for real-time embedded object detection in video. *arXiv preprint arXiv:1709.05943*, 2017.

[333] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1010–1019, 2016.

[334] M. R. Shaker, Shigang Yue, and T. Duckett. Vision-based reinforcement learning using approximate policy iteration. In *2009 International Conference on Advanced Robotics*, pages 1–6, 2009.

[335] Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving. *CoRR*, abs/1610.03295, 2016.

[336] Jie Shen, Stefanos Zafeiriou, Grigoris G Chrysos, Jean Kossaifi, Georgios Tzimiropoulos, and Maja Pantic. The first facial landmark tracking in-the-wild challenge: Benchmark and results. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 50–58, 2015.

[337] Y. Shi, L. Cui, Z. Qi, F. Meng, and Z. Chen. Automatic road crack detection using random structured forests. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3434–3445, 2016.

[338] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012.

[339] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. 2014.

[340] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[341] Vishwanath A Sindagi and Vishal M Patel. Multi-level bottom-top and top-bottom feature fusion for crowd counting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1002–1012, 2019.

[342] Satya P. Singh, Lipo Wang, Sukrit Gupta, Haveesh Goli, Parasuraman Padmanabhan, and Balázs Gulyás. 3d deep learning on medical images: A review, 2020.

[343] Gwangmo Song, Heesoo Myeong, and Kyoung Mu Lee. Seednet: Automatic seed generation with deep reinforcement learning for robust interactive segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1760–1768, 2018.

[344] Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5179–5187, 2015.

[345] Yibing Song, Chao Ma, Lijun Gong, Jiawei Zhang, Rynson WH Lau, and Ming-Hsuan Yang. Crest: Convolutional residual learning for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2555–2564, 2017.

[346] Concetto Spampinato, Simone Palazzo, and Daniela Giordano. Gamifying video object segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(10):1942–1958, 2016.

[347] Bradly C. Stadie, Pieter Abbeel, and Ilya Sutskever. Third-person imitation learning. *CoRR*, abs/1703.01703, 2017.

[348] Jayakumar Subramanian and Aditya Mahajan. Reinforcement learning in stationary mean-field games. page 251–259. International Foundation for Autonomous Agents and Multiagent Systems, 2019.

[349] Shanhui Sun, Jing Hu, Mingqing Yao, Jinrong Hu, Xiaodong Yang, Qi Song, and Xi Wu. Robust multimodal image registration using deep recurrent reinforcement learning. In *Asian Conference on Computer Vision*, pages 511–526. Springer, 2018.

[350] Kalaivani Sundararajan and Damon L. Woodard. Deep learning for biometrics: A survey. 51(3), 2018.

[351] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[352] Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, page 1057–1063, 1999.

[353] Richard S Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12*, pages 1057–1063. 2000.

[354] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[355] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[356] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep neural networks for object detection. In *Advances in neural information processing systems*, pages 2553–2561, 2013.

[357] Yansong Tang, Yi Tian, Jiwen Lu, Peiyang Li, and Jie Zhou. Deep progressive reinforcement learning for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5323–5332, 2018.

[358] Ran Tao, Efstratios Gavves, and Arnold WM Smeulders. Siamese instance search for tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1420–1429, 2016.

[359] Philippe Thévenaz and Michael Unser. Optimization of mutual information for multiresolution image registration. *IEEE transactions on image processing*, 9(12):2083–2099, 2000.

[360] Zhiqiang Tian, Xiangyu Si, Yaoyue Zheng, Zhang Chen, and Xiaojian Li. Multi-step medical image segmentation based on reinforcement learning. *JOURNAL OF AMBIENT INTELLIGENCE AND HUMANIZED COMPUTING*, 2020.

[361] Marin Toromanoff, Emilie Wirbel, and Fabien Moutarde. End-to-end model-free reinforcement learning for urban driving using implicit affordances. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7153–7162, 2020.

[362] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.

[363] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J Black. Video segmentation via object flow. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3899–3908, 2016.

[364] Y. Tsurumine, Y. Cui, K. Yamazaki, and T. Matsubara. Generative adversarial imitation learning with deep p-network for robotic cloth manipulation. In *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, pages 274–280, 2019.

[365] Yoshihisa Tsurumine, Yunduan Cui, Eiji Uchibe, and Takamitsu Matsubara. Deep reinforcement learning with smooth policy update: Application to robotic cloth manipulation. *Robotics and Autonomous Systems*, 112:72 – 83, 2019.

[366] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[367] Burak Uzkent, Christopher Yeh, and Stefano Ermon. Efficient object detection in large images using deep reinforcement learning. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1824–1833, 2020.

[368] Jack Valmadre, Luca Bertinetto, Joao Henriques, Andrea Vedaldi, and Philip HS Torr. End-to-end representation learning for correlation filter based tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2805–2813, 2017.

[369] Peter van Beek. Improved image selection for stack-based hdr imaging. *arXiv preprint arXiv:1806.07420*, 2018.

[370] Hado van Hasselt, Arthur Guez, and David Silver. Deep Reinforcement Learning with Double Q-learning. *arXiv e-prints*, page arXiv:1509.06461, September 2015.

[371] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI conference on artificial intelligence*, 2016.

[372] Leo Van Hove. Optimal denominations for coins and bank notes: in defense of the principle of least effort. *Journal of Money, Credit and Banking*, pages 1015–1021, 2001.

[373] Giuseppe Vecchio, Simone Palazzo, Daniela Giordano, Francesco Rundo, and Concetto Spampinato. Mask-rl: Multiagent video object segmentation framework through reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[374] Kashu Yamakazi Akihiro Sugimoto Viet-Khoa Vo-Ho, Ngan T.H. Le and Triet Tran. Agent-environment network for temporal action proposal generation. In *International Conference on Acoustics, Speech and Signal Processing*. 2021.

[375] Sudheendra Vijayanarasimhan, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, and Katerina Fragkiadaki. Sfm-net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*, 2017.

[376] Oriol Vinyals, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, Wojtek Czarnecki, Andrew Dudzik, Aja Huang, Petko Georgiev, Richard Powell, Timo Ewalds, Dan Horgan, Manuel Kroiss, Ivo Danihelka, John Agapiou, Junhyuk Oh, Valentin Dalibard, David Choi, Laurent Sifre, Yury Sulsky, Sasha Vezhnevets, James Molloy, Trevor Cai, David Budden, Tom Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Toby Pohlen, Dani Yogatama, Julia Cohen, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Chris Apps, Koray Kavukcuoglu, Demis Hassabis, and David Silver. AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. `https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/`, 2019.

[377] Athanasios Vlontzos, Amir Alansary, Konstantinos Kamnitsas, Daniel Rueckert, and Bernhard Kainz. Multiple landmark detection using multi-agent reinforcement learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 262–270. Springer, 2019.

[378] Guotai Wang, Maria A Zuluaga, Wenqi Li, Rosalind Pratt, Premal A Patel, Michael Aertsen, Tom Doel, Anna L David, Jan Deprest, Sébastien Ourselin, et al. Deepigeos: a deep interactive geodesic framework for medical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1559–1572, 2018.

[379] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018.

[380] Jane X. Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z. Leibo, Rémi Munos, Charles Blundell, Dharshan Kumaran, and Matthew Botvinick. Learning to reinforcement learn. *CoRR*, abs/1611.05763, 2016.

[381] Lijun Wang, Huchuan Lu, Xiang Ruan, and Ming-Hsuan Yang. Deep networks for saliency detection via local estimation and global search. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 3183–3192. IEEE, 2015.

[382] Mei Wang and Weihong Deng. Mitigating bias in face recognition using skewness-aware reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9322–9331, 2020.

[383] Mei Wang, Weihong Deng, Jiani Hu, Xunqiang Tao, and Yaohai Huang. Racial faces in the wild: Reducing racial bias by information maximization adaptation network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 692–702, 2019.

[384] Naiyan Wang and Dit-Yan Yeung. Learning a deep compact image representation for visual tracking. In *Advances in neural information processing systems*, pages 809–817, 2013.

[385] Tingwu Wang, Xuchan Bao, Ignasi Clavera, Jerrick Hoang, Yeming Wen, Eric Langlois, Shunshi Zhang, Guodong Zhang, Pieter Abbeel, and Jimmy Ba. Benchmarking model-based reinforcement learning. *CoRR*, abs/1907.02057, 2019.

[386] Yan Wang, Lei Zhang, Lituan Wang, and Zizhou Wang. Multitask learning for object localization with deep reinforcement learning. *IEEE Transactions on Cognitive and Developmental Systems*, 11(4):573–580, 2018.

[387] Yujiang Wang, Mingzhi Dong, Jie Shen, Yang Wu, Shiyang Cheng, and Maja Pantic. Dynamic face video segmentation via reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6959–6969, 2020.

[388] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[389] Zhouxia Wang, Jiawei Zhang, Mude Lin, Jiong Wang, Ping Luo, and Jimmy Ren. Learning a reinforced agent for flexible exposure bracketing selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1820–1828, 2020.

[390] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando De Freitas. Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581*, 2015.

[391] Wayne A Wickelgren. The long and the short of memory. *Psychological Bulletin*, 80(6):425, 1973.

[392] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[393] Aaron Wilson, Alan Fern, and Prasad Tadepalli. Using trajectory data to improve bayesian optimization for reinforcement learning. *Journal of Machine Learning Research*, 15(8):253–282, 2014.

[394] C. Wirth and J. Fürnkranz. On learning from game annotations. *IEEE Transactions on Computational Intelligence and AI in Games*, 7(3):304–316, 2015.

[395] Paul Wohlhart and Vincent Lepetit. Learning descriptors for object recognition and 3d pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3109–3118, 2015.

[396] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *European Conference on Computer Vision*, pages 3–19, 2018.

[397] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2411–2418, 2013.

[398] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848, 2015.

[399] Lu Xia, Chia-Chih Chen, and Jake K Aggarwal. View invariant human action recognition using histograms of 3d joints. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 20–27. IEEE, 2012.

[400] Sitao Xiang and Hao Li. On the effects of batch and weight normalization in generative adversarial networks. *arXiv preprint arXiv:1704.03971*, 2017.

[401] Yu Xiang, Alexandre Alahi, and Silvio Savarese. Learning to track: Online multi-object tracking by decision making. In *Proceedings of the IEEE international conference on computer vision*, pages 4705–4713, 2015.

[402] Fanyi Xiao and Yong Jae Lee. Track and segment: An iterative unsupervised approach for video object proposals. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 933–942, 2016.

[403] Hang Xiao and Hanchuan Peng. App2: automatic tracing of 3d neuron morphology based on hierarchical pruning of a gray-weighted image distance-tree. *Bioinformatics*, 29(11):1448–1454, 2013.

[404] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698, 2020.

[405] Haipeng Xiong, Hao Lu, Chengxin Liu, Liang Liu, Zhiguo Cao, and Chunhua Shen. From open set to closed set: Counting objects by spatial divide-and-conquer. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8362–8371, 2019.

[406] Hailiang Xu and Feng Su. Robust seed localization and growing with deep convolutional features for scene text detection. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 387–394. ACM, 2015.

[407] Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas S Huang. Deep interactive object selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 373–381, 2016.

[408] Tianyang Xu, Zhen-Hua Feng, Xiao-Jun Wu, and Josef Kittler. Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual object tracking. *IEEE Transactions on Image Processing*, 28(11):5596–5609, 2019.

[409] Xuanang Xu, Fugen Zhou, Bo Liu, Dongshan Fu, and Xiangzhi Bai. Efficient multiple organ localization in ct image using 3d region proposal network. *IEEE transactions on medical imaging*, 38(8):1885–1898, 2019.

[410] Yu-Syuan Xu, Tsu-Jui Fu, Hsuan-Kung Yang, and Chun-Yi Lee. Dynamic video segmentation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6556–6565, 2018.

[411] Kashu Yamazaki, Vidhiwar Singh Rathour, and T Le. Invertible residual network with regularization for effective medical image segmentation. *arXiv preprint arXiv:2103.09042*, 2021.

[412] Zhaoyi Yan, Yuchen Yuan, Wangmeng Zuo, Xiao Tan, Yezhen Wang, Shilei Wen, and Errui Ding. Perspective-guided convolution networks for crowd counting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 952–961, 2019.

[413] Zhibo Yang, Lihan Huang, Yupei Chen, Zijun Wei, Seoyoung Ahn, Gregory Zelinsky, Dimitris Samaras, and Minh Hoai. Predicting goal-directed human attention using inverse reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[414] Chao Yu, Jiming Liu, and Shamim Nemati. Reinforcement learning in healthcare: a survey. *arXiv preprint arXiv:1908.08796*, 2019.

[415] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, pages 1094–1100, 2020.

[416] Sangdoo Yun, Jongwon Choi, Youngjoon Yoo, Kimin Yun, and Jin Young Choi. Action-decision networks for visual tracking with deep reinforcement learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2711–2720, 2017.

[417] Daochen Zha, Kwei-Herng Lai, Kaixiong Zhou, and Xia Hu. Experience replay optimization. *arXiv preprint arXiv:1906.08387*, 2019.

[418] Da Zhang, Hamid Maei, Xin Wang, and Yuan-Fang Wang. Deep reinforcement learning for visual object tracking in videos. *arXiv preprint arXiv:1701.08936*, 2017.

[419] Dingwen Zhang, Le Yang, Deyu Meng, Dong Xu, and Junwei Han. Spftn: A self-paced fine-tuning network for segmenting objects in weakly labelled videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4429–4437, 2017.

[420] Jing Zhang, Wanqing Li, Philip O Ogunbona, Pichao Wang, and Chang Tang. Rgb-d-based action recognition datasets: A survey. *Pattern Recognition*, 60:86–105, 2016.

[421] Ke Zhang, Wei-Lun Chao, Fei Sha, and Kristen Grauman. Video summarization with long short-term memory. In *European conference on computer vision*, pages 766–782. Springer, 2016.

[422] Pengyu Zhang, Dong Wang, and Huchuan Lu. Multi-modal visual tracking: Review and experimental comparison, 2020.

[423] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 589–597, 2016.

[424] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 405–420, 2018.

[425] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[426] Zhong-Qiu Zhao, Shou-Tao Xu, Dian Liu, Wei-Dong Tian, and Zhi-Da Jiang. A review of image set classification. *Neurocomputing*, 335:251–260, 2019.

[427] Yefeng Zheng, David Liu, Bogdan Georgescu, Hien Nguyen, and Dorin Comaniciu. 3d deep learning for efficient and robust landmark detection in volumetric data. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 565–572. Springer, 2015.

[428] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

[429] Kaiyang Zhou, Yu Qiao, and Tao Xiang. Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[430] Kaiyang Zhou, Tao Xiang, and Andrea Cavallaro. Video summarisation by classification with deep reinforcement learning. *arXiv preprint arXiv:1807.03089*, 2018.

[431] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2349–2358, 2017.

[432] Xiahai Zhuang and Juan Shen. Multi-scale patch and multi-modality atlases for whole heart segmentation of mri. *Medical image analysis*, 31:77–87, 2016.

[433] Will Y Zou, Xiaoyu Wang, Miao Sun, and Yuanqing Lin. Generic object detection with dense neural patterns and regionlets. *arXiv preprint arXiv:1404.4316*, 2014.