



Deep reinforcement learning in computer vision: a comprehensive survey

Ngan Le^{1,2} · Vidhiwar Singh Rathour^{1,2} · Kashu Yamazaki^{1,2} · Khoa Luu^{1,2} · Marios Savvides^{1,2}

Published online: 29 September 2021

© The Author(s), under exclusive licence to Springer Nature B.V. 2021

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.

Keywords Deep learning · Reinforcement learning · Deep reinforcement learning · Computer vision · Autonomous landmark detection · Object detection · Object tracking · Image registration · Image segmentation · Video analysis

Vidhiwar Singh Rathour and Kashu Yamazaki have equal contribution.

✉ Ngan Le
thile@uark.edu

Extended author information available on the last page of the article

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 (Sutton and Barto 2018). 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 Rotman (2013), Giles (2017) 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 Arulkumaran et al. (2017), François-Lavet et al. (2018), Yu et al. (2019) have been introduced recently. While Arulkumaran et al. (2017) covers central algorithms in DRL, François-Lavet et al. (2018) 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, Yu et al. (2019) 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 Sect. 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 Sect. 3, which starts with the Markov Decision Process (MDP) and continues with value function and Q-function. In the end of Sect. 3, we introduce

various techniques in RL under two categories of model-based and model-free RL. Next, we introduce DRL in Sect. 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 Sects. 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 Sect. 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. 1a. A computational model of a single neuron is called a perceptron which consists of one or more inputs, a processor, and a single output.

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 $f: \mathbb{R}^N \rightarrow \mathbb{R}^M$ 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 weights and are known as weights $\theta = \{w_i\}$. Mathematically, the output \mathbf{o} of a one neuron is defined as:

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

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 perceptron (MLP) neural network is

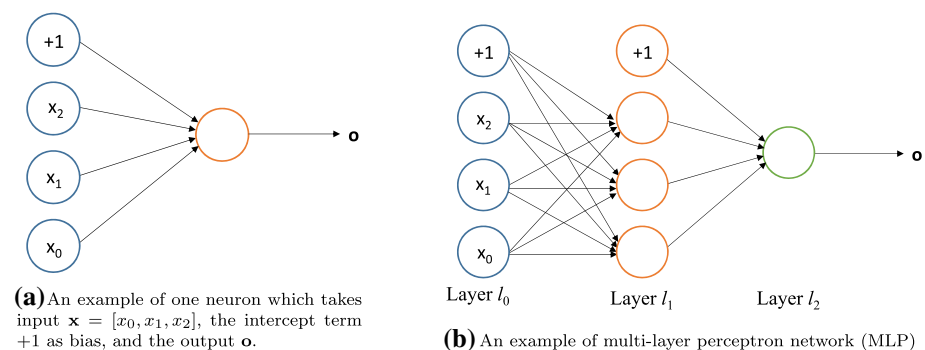


Fig. 1 An example of one neuron and multi-layer perceptron network (MLP)

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. 1b, 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 An and Cho (2015) and VAE framework is illustrated in Fig. 2a. 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. 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. 2b.

2.4 Convolutional Neural Networks (CNN)

Convolutional Neural Network (CNN) LeCun et al. (1988) LeCun et al. (1998) 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

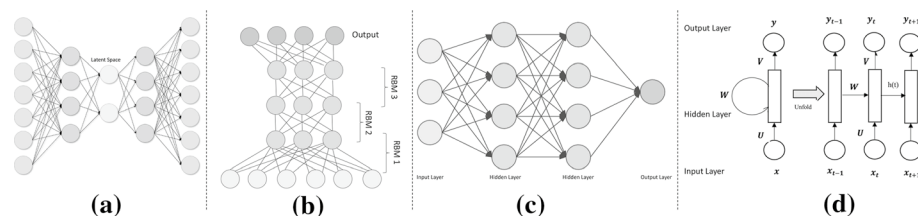


Fig. 2 An illustration of various DL architectures. **a** Autoencoder (AE), **b** Deep Belief Network, **c** Convolutional Neural Network (CNN), **d** Recurrent Neural Network (RNN)

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 (Deng et al. 2009; Krizhevsky et al. 2012) (Fig. 3).

A typical CNN is composed of multiple stages, as shown in Fig. 2c. 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

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

where $\mathbf{w}^l, \mathbf{b}^l$ are trainable parameters as in MLPs at layer l th. $\mathbf{x} \in \mathbb{R}^{c \times h \times w}$ is vectorized from an input image with c being the color channels, h the image height and w the image width. $\mathbf{o} \in \mathbb{R}^{n \times h' \times w'}$ 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* (Xie et al. 2020; Zhao et al. 2019), CNN outperforms human capacities (Buetti-Dinh et al. 2019). *Video analysis* Zhang and Wang (2020), Li et al. (2018d) is another application that turns the CNN model from a detector (Vo-Ho et al. 2021) into a tracker (Fan et al. 2010). As a special case of *image segmentation* (Le et al. 2018b, a), *saliency detection* is another computer vision application that uses CNN (Wang et al. 2015a; Li and Yu 2015). In addition to the previous applications, *pose estimation* (Patacchiola and Cangelosi 2017), Toshev and Szegedy (2014) 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. Gki-oxari et al. (2015) 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

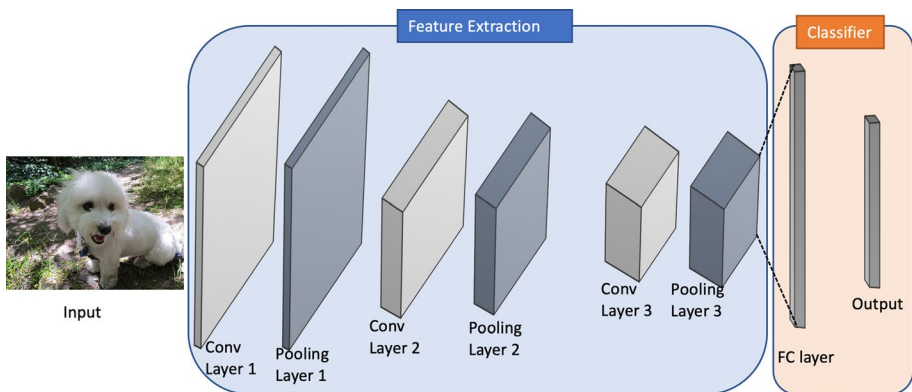


Fig. 3 Architecture of a typical convolutional network for image classification containing three basic layers: convolution layer, pooling layer and fully connected layer

action recognition in video sequences is reviewed in Zhang et al. (2016a). *Text detection and recognition* using CNN is the next step of optical character recognition (OCR) (Xu and Su 2015) and word spotting (Jaderberg et al. 2014). Not only in computer vision, CNN has been successfully applied into other domains such as *speech recognition and speech synthesis* (Nassif et al. 2019; Ning et al. 2019), biometrics (Luu et al. 2016; Duong et al. 2019; Nhan Duong et al. 2017; Sundararajan and Woodard 2018; Minaee et al. 2019), biomedical (Le et al. 2020a; Singh et al. 2020; Le et al. 2020b; Yamazaki et al. 2021).

2.5 Recurrent Neural Networks (RNN)

RNN is an extremely powerful sequence model and was introduced in the early 1990s (Jordan 1990). 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. 2d.

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

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

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) Rumelhart (1998) 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

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

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 (Fig. 4).

The difficulty of training RNN to capture long-term dependencies has been studied in Bengio et al. (1994). To address the issue of learning long-term dependencies, Hochreiter and Schmidhuber (1997) 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 Cho et al. (2014b) 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 Mikolov et al. (2011), Li et al. (2015), speech recognition Graves et al. (2013), Chorowski et al. (2015), machine translation

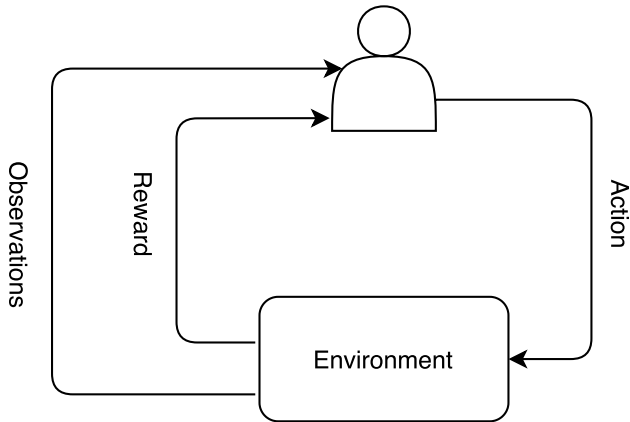


Fig. 4 An illustration of agent-environment interaction in RL

Kalchbrenner and Blunsom (2013), Luong et al. (2014), question answering Hill et al. (2015), image captioning (Mao et al. 2014; Donahue et al. 2014), 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.

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 $\text{MDP}(S, \mathcal{A}, \gamma, T, R)$, 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 $(s_0, a_0, r_1, s_1, a_1, r_2, \dots)$ 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 $(s_0, a_0, r_1, s_1, a_1, r_2, \dots)$ is called *trajectory distribution* and is denoted as:

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

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:

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

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

$$\mathbb{E}_{\mathcal{T}_\pi} \sum_{t=0}^{\tau-1} r_{t+1} \rightarrow \max_{\pi} \quad (7)$$

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

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

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

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:

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

$V^{\pi}(s_t)$ 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:

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

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:

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

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:

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

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 (Levine and Koltun 2014; Morimoto et al. 2003), Temporal Difference Learning Martinez-Marin and Duckett (2005), Policy Iteration Shaker et al. (2009) and Monte Carlo Hester et al. (2011).
- **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 Nguyen et al. (2013) and linear regression Mordatch et al. (2016). Some common methods for stochastic models are Gaussian processes Deisenroth et al. (2014), Kupcsik et al. (2017), Andersson et al. (2015), Expectation-Maximization Coates et al. (July 2009) and dynamic Bayesian networks Nguyen et al. (2013).
- **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 El-Fakdi and Carreras (2008), Morimoto and Atkeson (2009), information theory Kupcsik et al. 2017 (2017), Kupcsik et al. (2013) and sampling based Bagnell and Schneider (2001).
- **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 (Bagnell and Schneider 2001; Depraetere et al. 2014; Wilson et al. 2014) and averaged returns functions Boedecker et al. (2014), Abbeel et al. (2010). 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 Wang et al. (2019b) with over 18 benchmarking environments including noisy environments. The Fig. 5 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

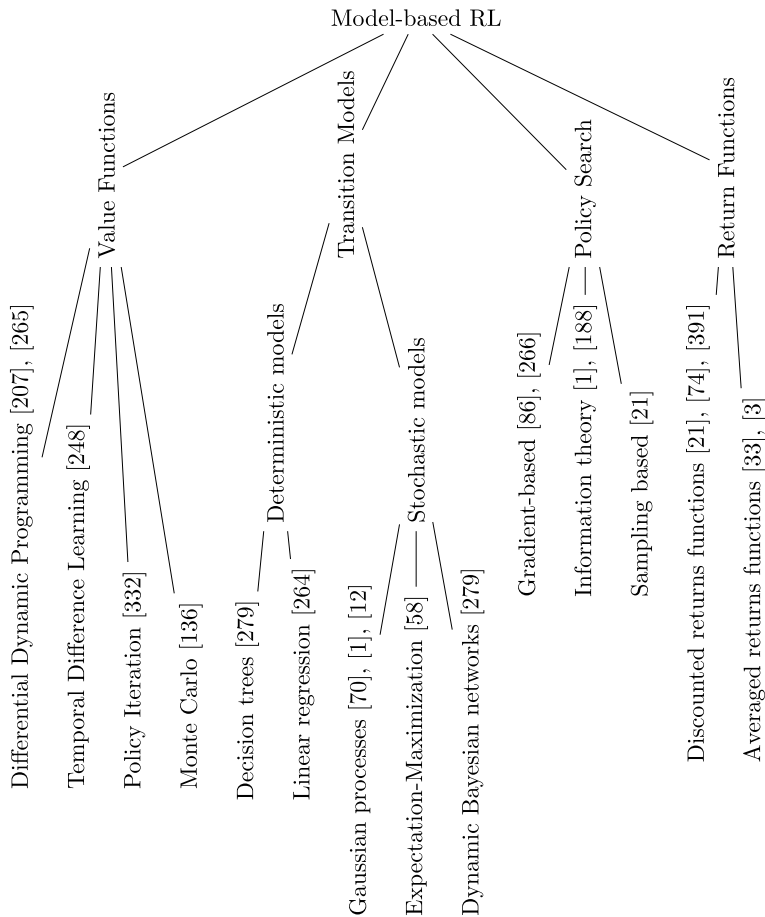


Fig. 5 Summarization of model-based RL approaches

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 obtain 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 Williams (1992) as an example, policy gradient parameterizes the policy and updates parameters θ ,

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

where \mathcal{R} is the total accumulated return and defined in Eq. 6. Common used policies are Gibbs policies Bagnell (2012), Sutton et al. (1999) and Gaussian policies Peters and

Schaal (2008). 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:

$$Q(s_t, a_t) = Q(s_t, a_t) + \lambda \delta_t \quad (14)$$

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

Target at self-play Chess, Wirth and Fürnkranz (2015) investigates inasmuch it is possible to leverage the qualitative feedback for learning an evaluation function for the game. Runarsson and Lucas (2012) 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:

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

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 Niedzwiedz et al. (2008), [?] and average return Paschalidis et al. (2009), Bhatnagar et al. (2009). 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

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

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.

4.1.1 Value-based DRL methods

Deep Q-Learning Network (DQN): Deep Q-learning (Mnih et al. 2015) (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. 6. 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:

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

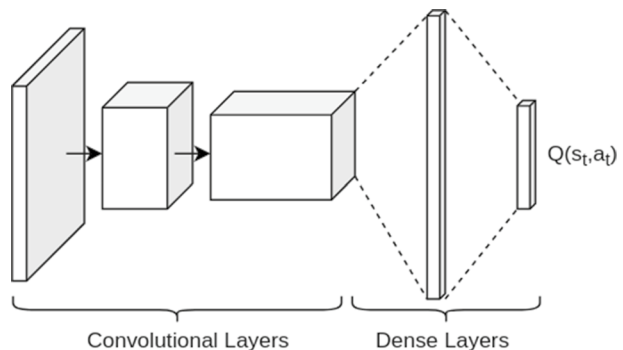
Where θ is vector of parameters, $\theta \in \mathbb{R}^{|S||A|}$ and s_{t+1} is a sample from $T(s_{t+1}|s_t, a_t)$ with input of (s_t, a_t) .

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

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

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 van Hasselt et al. (2015) is an improvement of DQN that combines double Q-learning (Hasselt 2010) with DQN and it aims at reducing observed overestimation

Fig. 6 Network structure of Deep Q-Network (DQN), where Q-values $Q(s, a)$ are generated for all actions for a given state



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:

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

Thus

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

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

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

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 (Wang et al. 2015b) incorporates approximation of V^* explicitly in a computational graph by introducing an advantage function as follows:

$$A^\pi(s_t, a_t) = Q^\pi(s_t, a_t) - V^\pi(s_t) \quad (21)$$

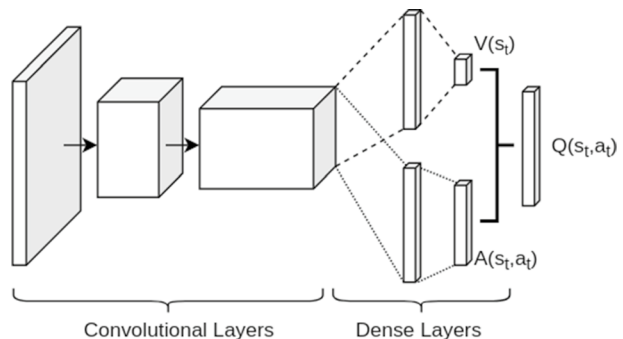
Therefore, we can reformulate Q-value: $Q^*(s, a) = A^*(s, a) + V^*(s)$. This implies that after DL the feature map is decomposed into two parts corresponding to $V^*(s)$ and $A^*(s, a)$ as illustrated in Fig. 7. This can be implemented by splitting the fully connected layers in the DQN architecture 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:

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

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

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

Fig. 7 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



Furthermore, to address the limitation of memory and imperfect information at each decision point, Deep Recurrent Q-Network (DRQN) Hausknecht and Stone (2015) employed RNNs into DQN by replacing the first fully-connected layer with an RNN. Multi-step DQN De Asis et al. (2018) is one of the most popular improvements of DQN by substituting one-step approximation with N-steps.

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 .

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

For any MDP and differentiable policy π_θ , the gradient of objective Eq. 23 is defined by policy gradient theorem Sutton et al. (2000) as follows:

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

REINFORCE: REINFORCE was introduced by Williams (1992) to approximately calculate the gradient in Eq. 24 by using Monte-Carlo estimation. In REINFORCE approximate estimator, Eq. 24 is reformulated as:

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

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 optimum 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 Konda and Tsitsiklis (2000) Mnih et al. (2016a) 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. 8, 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.

Advantage Actor-Critic (A2C) Advantage Actor-Critic (A2C) Mnih et al. (2016b) consist of two neural networks i.e. actor network $\pi_{\theta}(a_t|s_t)$ representing for policy and critic network V_{ω}^{π} 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 defined as:

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

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:

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

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

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

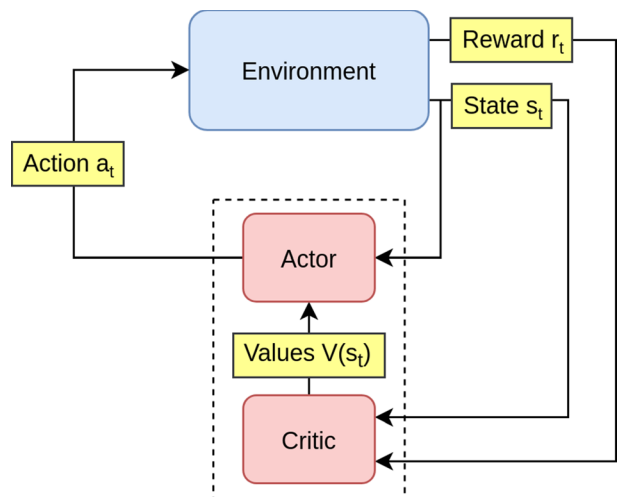
Similar to DQN, on each update a target is computed using current approximation:

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

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: $\nabla^{critic} = \frac{\partial \mathcal{L}}{\partial \omega}$.

Fig. 8 Flowchart showing the structure of actor critic algorithm



- Step 5: Compute actor gradient: $\nabla^{actor} = \frac{1}{B} \sum_T \nabla_{\theta} \log \pi(a_t | s_t) A^{\pi}(s_t, a_t)$

Asynchronous Advantage Actor Critic (A3C) Besides A2C, there is another strategy to implement an Actor-Critic agent. Asynchronous Advantage Actor-Critic (A3C) Mnih et al. (2016b) 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. 9.

In order to overcome the limitation of speed, GA3C Babaeizadeh et al. (2016) was proposed and it achieved a significant speedup compared to the original CPU implementation. To more effectively train A3C, Holliday and Le (2020) proposed FFE which forces random exploration at the right time during a training episode, that can lead to improved training performance.

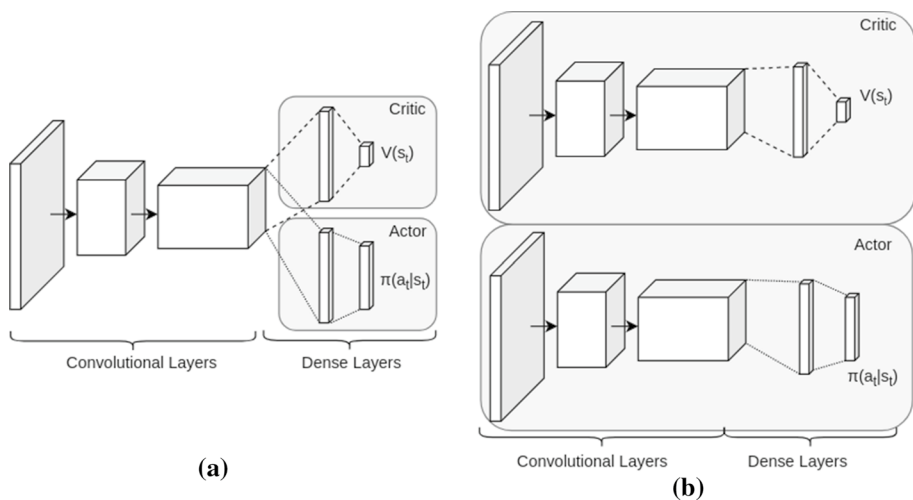


Fig. 9 An illustration of Actor-Critic algorithm in two cases: sharing parameters (a) and not sharing parameters (b)

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 Finn et al. (2016). 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 Plaet et al. (2020) 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 Mnih et al. (2015) which has been successfully applied to classic Atari and illustrated in Fig. 6. DQN uses CNNs to deal with high dimensional state space like pixels, to approximate the Q-value function.

Monte Carlo tree search (MCTS) MCTS Coulom (2006) 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 Alaniz (2018). 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 Jia et al. (2020). 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 Kurutach et al. (2018) is mainly based on Trust Region Policy Optimization (TRPO)

Schulman et al. (2015) which imposes a trust region constraint on the policy to further stabilize learning.

Model-Based Meta-Policy-Optimization (MB-MPO) MB-MPO Clavera et al. (2018) 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 $\pi^0, \pi^0, \dots, \pi^k$. Each policy has its own data collection, then the data collected from $\pi^0, \pi^1, \dots, \pi^k$ is used to train π^{k+1} .

4.3 Good practices

Inspired by Deep Q-learning Mnih et al. (2015), we discuss some useful techniques that are used during training an agent in DRL framework in practices.

Experience replay Experience replay Zha et al. (2019) 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 Mnih et al. (2015), 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 Lillicrap et al. (2015b).

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

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

DRL Algorithms	Description	Category
DQN Mnih et al. (2015)	Deep Q Network	Value-based, Off-policy
Double DQN van Hasselt et al. (2015)	Double Deep Q Network	Value-based, Off-policy
Dueling DQN Wang et al. (2015b)	Dueling Deep Q Network	Value-based, Off-policy
MCTS Alaniz (2018)	Monte Carlo tree search	Value-based, On-Policy
UCRL-VTRJia et al. (2020)	Optimistic planning problem	Value-based, On-Policy
DDPG Lillicrap et al. (2015a)	DQN with Deterministic Policy Gradient	Policy gradient, Off-policy
TRPO Schulman et al. (2015)	Trust Region Policy Optimization	Policy gradient, On-policy
PPO Schulman et al. (2017a)	Proximal Policy Optimization	Policy gradient, On-policy
ME-TRPO Kurutach et al. (2018)	Model-Ensemble Trust-Region Policy Optimization	Policy gradient, On-policy
MB-MPO Clavera et al. (2018)	Model-Based Meta- Policy-Optimization	Policy gradient, On-policy
A3C Mnih et al. (2016b)	Asynchronous Advantage Actor Critic	Actor Critic, On-Policy
A2C Mnih et al. (2016b)	Advantage Actor Critic	Actor Critic, On-Policy

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. Criminisi et al. (2010) 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. Gauriau et al. (2014) extended the work of Criminisi et al. (2010) 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, Ghesu et al. (2017) 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 Lindeberg (2013), 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 , $s_t = I(\mathbf{p}_t)$ is defined as an axis-aligned box of image intensities extracted from the image I and centered at the voxel-position \mathbf{p}_t in image space. An action a_t allows the agent to move from any voxel position \mathbf{p}_t to an adjacent voxel position \mathbf{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 $M1$. 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 Ghesu et al. (2016) and 3D-DL Zheng et al. (2015)

Focus on anatomical landmark localization in 3D fetal US images, Alansary et al. (2019) 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, Al and Yun (2019) adopts an actor-critic Mnih et al. (2016a) 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, Vlontzos et al. (2019) 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) Girard and Emami (2015) 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.

Different from the previous works on RL-based landmark detection, which detect a single landmark, Jain et al. (2020) 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. 10. 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 Ghesu et al. (2017), Al and Yun (2019), Alansary et al. (2019), Vlontzos et al. (2019), Jain et al. (2020).

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 Ghesu et al. (2017)	2017	DQN	6 action: 2 per axis	State: an axis-aligned box centered at the voxel-position. Action: move from \mathbf{p}_i to \mathbf{p}_{i+1} . Reward: distance-based feedback	Average accuracy increase 20–30%. Lower distance error than other techniques such as SADNN Ghesu et al. (2016) and 3D-DL Zheng et al. (2015)	3D CT Scan
SLD Alansary et al. (2019)	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 Li et al. (2018e). Available code
SLD Al and Yun (2019)	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 Cai et al. (2015).
MLD Vrontzos et al. (2019)	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	Collab DQN got better results than supervised CNN and DQN	Brain MRI landmark Jack et al. (2008), Cardiac MRI landmark de Marvao et al. (2014), Fetal brain landmark Alansary et al. (2019). Available code

Table 3 (continued)

Approaches	Year	Training Technique	Actions	Remarks	Performance	Datasets and source code
MLD Jain et al. (2020)	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

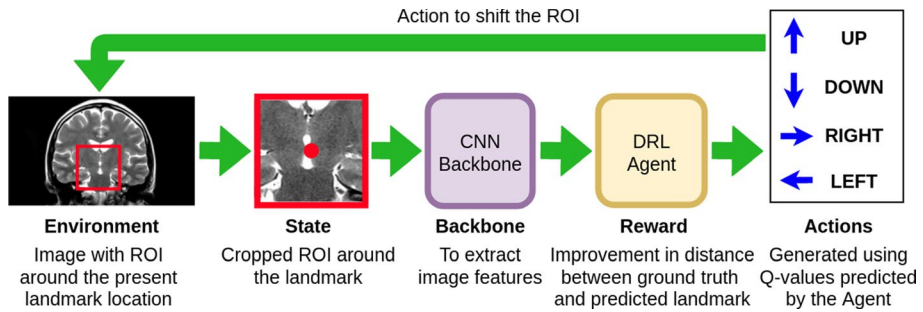


Fig. 10 DRL implementation for landmark detection, The red point corresponds to the current landmark location and Red box is the Region of Interest (ROI) centered around the landmark, the actions of DRL agent shift the ROI across the image to maximize the reward corresponding to the improvement in distance 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 Girshick et al. (2014), 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 Girshick (2015), 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 Ren et al. (2015), 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 Lin et al. (2017) and Fast YOLO Shafiee et al. (2017) have been proposed to address the imbalanced data problem in object detection with focal loss Lin et al. (2017), and perform object detection in video on embedded devices in a real-time manner Shafiee et al. (2017).

Considering MDP as the framework for solving the problem, Caicedo and Lazebnik (2015) 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 Caicedo and Lazebnik (2015) was proposed by Bellver et al. (2016), 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 Simonyan and Zisserman (2014) 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, Mathe et al. (2016) 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, Jie et al. (2016) 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 Deng et al. (2009) Russakovsky et al. (2015) trained VGG-16 Simonyan and Zisserman (2014) model and for reward the change in IOU across an action was used. Tree-RL utilized a top-down tree 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 Maicas et al. (2017), 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 He et al. (2016) was used as the backbone and DQN as the agent.

Different from the previous methods, Wang et al. (2018) 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 Pirinen and Sminchisescu (2018), 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, Ayle et al. (2020) 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 He et al. (2016) Szegedy et al. (2017) pretrained on ImageNet Deng et al. (2009) Russakovsky et al. (2015) 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 Li et al. (2010) 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 Mathe et al. (2016) was proposed by Uz kent et al. (2020), 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.

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 Navarro et al. (2020), 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, Liu et al. (2020b) introduced Reinforced Axial Refinement Network (RARNet) 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 Geiger et al. (2012) 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. 11. 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

Table 4 Comparing various DRL-based object detection methods

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets and source code
Active Object Localization Caicedo and Lazebnik (2015)	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 pretrained CNN	Higher mAP as compared to methods that did not use region proposals like MultiBox Erhan et al. (2014), RegionLets Zou et al. (2014), DetNet Szegedy et al. (2013), and second best mAP as compared to R-CNN Girshick et al. (2014)	Pascal VOC-2007 Everingham et al. (2007), 2012 Everingham and Winn (2011) Image Dataset
Hierarchical Object Detection Bellver et al. (2016)	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 Simonyan and Zisserman (2014)	Objects detected with very few region proposals per image	Pascal VOC-2007 Image Dataset Everingham et al. (2007). Available Code

Table 4 (continued)

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets and source code
Visual Object Detection Mathe et al. (2016)	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 Krizhevsky et al. (2012)	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) Gonzalez-Garcia et al. (2015) Uijlings et al. (2013)	Pascal VOC 2012 Object detection challenge Everingham and Winn (2011)
Tree-Structured Sequential Object Localization (Tree-RL) Jie et al. (2016)	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 Deng et al. (2009) Russakovsky et al. (2015)	Tree-RL with faster R-CNN outperformed RPN with fast R-CNN Girshick (2015) in terms of AP and comparable results to Faster R-CNN Ren et al. (2015)	Pascal VOC 2007 Everingham et al. (2007) and 2012 Everingham and Winn (2011)

Table 4 (continued)

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets and source code
Active Breast Lesion Detection Maicas et al. (2017)	2017	DQN	9 actions: 6 translation, 2 scaling, 1 trigger	States: feature vector of current region, Reward: improvement in localization	ResNet He et al. (2016)	Comparable true positive and false positive proportions as compared to SL McClymont et al. (2014) and Ms-C Gubern-Mérida et al. (2015), but with lesser mean inference time	DCE-MRI and T1-weighted anatomical dataset McClymont et al. (2014)
Multitask object localization Wang et al. (2018)	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 Simonyan and Zisserman (2014) with ImageNet Deng et al. (2009) Russakovsky et al. (2015)	Better mAP as compared to Multi-Box Erhan et al. (2014), Caicedo et al. Caicedo and Lazebnik (2015) and second best to R-CNN Girshick et al. (2014)	Pascal VOC-2007 Image Dataset Everingham et al. (2007)
Bounding-Box Automated Refinement Ayle et al. (2020)	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 He et al. (2016)	Better final IOU for boxes generated by methods such as RetinaNet Lin et al. (2017)	Pascal VOC-2007 Everingham et al. (2007), 2012 Everingham and Winn (2011) Image Dataset

Table 4 (continued)

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets and source code
Efficient Object Detection in Large Images Uzkent et al. (2020)	2020	DQN	binary action array: where 1 means that the agent would consider acquiring sub-patches for that particular patch	Course CPNet and fine FPNNet level search. States: selected region. Reward: detection recall image acquisition cost. Policy: REINFORCE Sutton and Barto (2018)	ResNet32 He et al. (2016) for policy network. and YOLOv3 Redmon and Farhadi (2018) with DarkNet-53 for Object detector	Higher mAP and lower run time as compared to other methods such as Gao et al. (2018)	Caltech Pedestrian dataset (CPD) Dollár et al. (2009) Available Code
Organ Localization in CT Navarro et al. (2020)	2020	DQN	11 actions: 6 translation, 2 scaling, 3 deformation	States: region inside the Bounding box. Reward: change in IOU	Architecture similar to Alansary et al. (2019)	Lower distance error for organ localization and run time as compared to other methods such as 3D-RCNN Xuanang et al. (2019) and CNNs Humpire-Mamani et al. (2018)	CT scans from the VISCERAL dataset Jimenez-del Toro et al. (2016)
Monocular 3D Object Detection Liu et al. (2020b)	2020	DQN Mnih et al. (2015)	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 He et al. (2016)	Higher average precision (AP) compared to Mousavian et al. (2017), Qin et al. (2019), Li et al. (2019) and Brazil and Liu (2019)	KITTI Geiger et al. (2012)

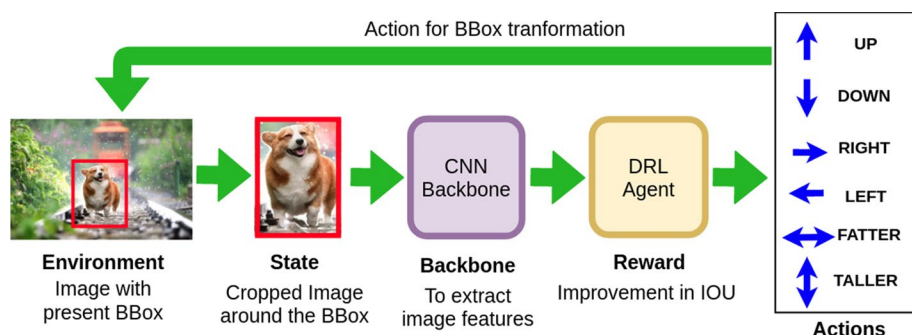


Fig. 11 DRL implementation for object detection. The red box corresponds to the initial bounding box which for $t=0$ is predicted by some other algorithm or the transformed bounding box by previous iterations of DRL using the actions to maximize the improvement in IOU

DRL, actions predicted by the DRL agent predict a series of bounding box transformation 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 Caicedo and Lazebnik (2015), Bellver et al. (2016), Ayle et al. (2020), Wang et al. (2018), Jie et al. (2016), Navarro et al. (2020).

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 Wu et al. (2013), Weiming et al. (2012), where Weiming et al. (2012) 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 Denzler and Paulus (1994) Murray and Basu (1994) Kim et al. (2005). 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 Luo et al. (2017), where the authors used CNNs along with an LSTM Hochreiter and Schmidhuber (1997) in their implementation. They used the actor-critic algorithm Mnih et al. (2016a) 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 Zhang et al. (2017a). The method included using an RNN along with REINFORCE Williams (1992) 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 Hochreiter and Schmidhuber (1997) 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 Guo et al. (2018). 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 Simonyan and Zisserman (2014) 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 Hochreiter and Schmidhuber (1997) 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 Ren et al. (2018b). 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, Chen et al. (2018) proposed an actor-critic Mnih et al. (2016a) 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 Δx for relative horizontal translation, Δy for relative vertical translation, and Δs 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 Simonyan and Zisserman (2014) was used as a backbone, and the actor-critic network was trained using the DDPG approach Lillicrap et al. (2015b).

An improvement to Chen et al. (2018) for SOT was proposed by Dunnhofer et al. (2019), 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: Δx , Δy for relative vertical translation, Δw for width scaling, and Δh 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 He et al. (2016) as backbone followed by an LSTM unit Wickelgren (1973) Hochreiter and Schmidhuber (1997) to encode past information, and performed training based on the on-policy A3C framework Mnih et al. (2016a).

In MOT the algorithm is responsible to track trajectories of multiple objects in the given video. Many attempts have been made with MOT including Choi (2015), Chu et al. (2017) and Yoon et al. (2016). 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 Choi (2015) and Online Chu et al. (2017) Yoon et al. (2016). 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 Kim and Jeon (2014), Particle Filtering Okuma et al. (2004) or Markov decision Xiang et al. (2015). These techniques however are prone to errors due to environmental constraints.

To overcome the constraints of MOT by previous methods, Xiang et al. (2015) 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 Ng et al. (2000).

Previous approaches for MOT follow a tracking by detection technique that is prone to errors. An improvement was proposed by Ren et al. (2018a), 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 Δx for relative horizontal translation, Δy for relative vertical translation, Δw for width scaling, and Δh for height scaling, and the reward consisted of values 1, 0, -1 based on the IoU.

Another method for MOT was proposed by Jiang et al. (2018), where the authors used LSTM Hochreiter and Schmidhuber (1997) and DRL to approach the problem of multi-object tracking. The method described in the paper used three basic components: a YOLO V2 Milan et al. (2017) 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 Simonyan and Zisserman (2014) for CNN backbone and performed experiments on MOT benchmark Leal-Taixé et al. (2015) for people tracking.

To address the problems in existing tracking methods such as varying numbers of targets, non-real-time tracking, etc, Jiang et al. (2019) proposed a multi-object tracking algorithm based on a multi-agent DRL tracker (MADRL). In their object tracking pipeline the authors used YOLO-V3 Redmon and Farhadi (2018) as object detector, where

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 Luo et al. (2017)	2017	Actor-Critic (a3c) Mnih et al. (2016a)	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 Hochreiter and Schmidhuber (1997)	ConvNet-LSTM	Higher accumulated reward and episode length as compared to methods like MIL Babenko et al. (2009), MeanShift Comaniciu (2000), KCF Henriques et al. (2014)	VizDoom Kempka et al. (2016), Unreal Engine
DRL for object tracking Zhang et al. (2017a)	2017	DRLT	None	State: feature vector, Reward: change in IOU use of LSTM Hochreiter and Schmidhuber (1997) and REINFORCE Williams (1992)	YOLO network Redmon et al. (2016)	Higher area under curve (success rate Vs overlap threshold), precision and speed (fps) as compared to STUCK Hare et al. (2015) and DLT Wang and Yeuug (2013)	Object tracking benchmark Wu et al. (2013). Available Code
Dual-agent deformable face tracker Guo et al. (2018)	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 Hochreiter and Schmidhuber (1997)	VGG-M Simonyan and Zisserman (2014)	Lower normalized point to point error for landmarks and higher success rate for facial tracking as compared to ICCR Krizhevsky et al. (2012), MDM Shen et al. (2015), Xiao et al Black and Yacoob (1995), etc.	Large-scale face tracking dataset, the 300-VW test set Shen et al. (2015)

Table 5 (continued)

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets and Source code
Tracking with iterative shift Ren et al. (2018b)	2018	Actor-critic Mnih et al. (2016a)	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 Song et al. (2017), ADNet Yun et al. (2017), MDNet Nam and Han (2016), HCFT Ma et al. (2015), SINT Tao et al. (2016), DeepSRDCF Danelljan et al. (2015), and HDT Qi et al. (2016)	OTB-2015 Yi et al. (2015), Temple-Color Liang et al. (2015), and VOT-2016 Dataset Kristan et al. (2015)
Tracking with actor-critic Chen et al. (2018)	2018	Actor-critic Mnih et al. (2016a)	3 actions: Δx , Δy , and Δs	States: image region using bounding box. Reward: IOU greater then threshold. Offline training, online tracking	VGG-M Simonyan and Zisserman (2014)	Higher average precision score then PTAV Fan and Ling (2017), CFNet Valmadre et al. (2017), ACFN Choi et al. (2017), SiameFC Bertinetto et al. (2016), ECO-HC Danelljan et al. (2015), etc.	OTB-2013 Wu et al. (2013), OTB-2015 Yi et al. (2015) and VOT-2016dataset Kristan et al. (2015) Available Code

Table 5 (continued)

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets and Source code
Visual tracking and expert demonstrator Dunnhofer et al. (2019)	2019	Actor-critic (a3c) Mnih et al. (2016a)	4 actions: Δx , Δy , Δw and Δh	States: image region using bounding box. Reward: change in IOU. SOT using LSTM Wickelgren (1973) Hochreiter and Schmidhuber (1997)	ResNet-18 He et al. (2016)	Comparable success and precision scores as compared to LADCF Tianyang et al. (2019), SiamRPN Li et al. (2013), (2018a) and ECO Danelljan et al. (2017)	GOT-10k Huang et al. (2019a), LaSOT Fan et al. (2019), UAV123 Mueller et al. (2016), OTB-100 Wu et al. (2013), VOT-2018 Kristan et al. (2018) and VOT-2019
Object tracking by decision making Xi-ang et al. (2015)	2015	TLD Tracker Kalal et al. (2011)	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 Ng et al. (2000)	None	Comparable multiple object tracking accuracy (MOTA) and multiple object tracking precision (MOTP) Bernardin and Stiefelhagen (2008) as compared to DPNMS Pirsivash et al. (2011), TCODAL Bae and Yoon (2014), SegTrack Milan et al. (2015), MotiCon Leal-Taixé et al. (2014), etc.	MOT15 dataset Leal-Taixé et al. (2015) Available Code

Table 5 (continued)

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets and Source code
Collaborative multi object trackerRen et al. (2018a)	2018	DQN	4 actions: Δx , Δy , Δw and Δh	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) Bernardino and Stiefelhagen (2008) as compared to SCEA Yoon et al. (2016), MDP Xiang et al. (2015), CDADDALpb Bae and Yoon (2017), AMIR15 Sadeghian et al. (2017)	MOT15 Leal-Taixé et al. (2015) and MOT16 Milan et al. (2016) datasets

Table 5 (continued)

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets and Source code
Multi object tracking in videoJiang et al. (2018)	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 Milan et al. (2017) for detector and LSTM Hochreiter and Schmidhuber (1997)	VGG-16 Simonyan and Zisserman (2014)	Comparable if not better multiple object tracking accuracy (MOTA) and multiple object tracking precision (MOTP) Bernardin and Stiefelhagen (2008) as compared to RNN-LSTM Leal-Taixé et al. (2015), LP-SSVM Xiang et al. (2015), MDPSubCNNLeal-Taixé et al. (2016), and SiameseCNN Hamid RezaTofighi et al. (2015)	MOT15 Dataset Leal-Taixé et al. (2015)

Table 5 (continued)

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets and Source code
Multi agent multi object trackerJiang et al. (2019)	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 than threshold. YOLO-V3 Redmon and Farhadi (2018) for detection and LSTM Hochreiter and Schmidhuber (1997).	VGG-16 Simonyan and Zisserman (2014)	Higher running time, not better multiple object tracking accuracy (MOTA) and multiple object tracking precision (MOTP) Bernardin and Stiefelhagen (2008) as compared to RNN-LSTM Leal-Taixé et al. (2015), LP-SSVM Xiang et al. (2015), MDPSubCNNLeal-Taixé et al. (2016), and SiameseCNN Hamid RezaTofighi et al. (2015)	MOT15 challenge benchmark Leal-Taixé et al. (2015)

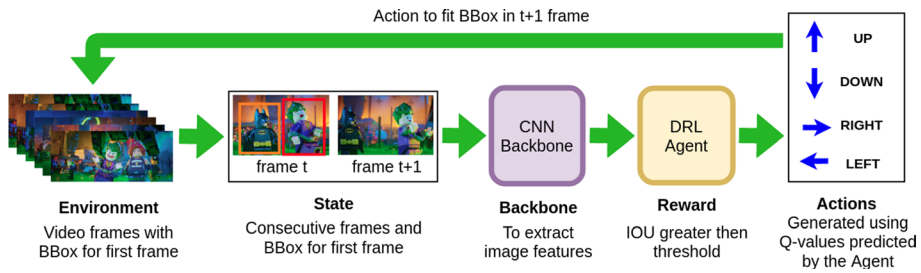


Fig. 12 DRL implementation for object tracking. Here the state consists of two consecutive frames with bounding box locations for the first frame that is predicted by some object detection algorithm or by the previous iteration of DRL, the actions move the bounding box present in the first frame to fit the object in the second frame to maximize the reward which is the whether the IOU is greater than a given threshold or not

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 Simonyan and Zisserman (2014) followed by an LSTM unit Hochreiter and Schmidhuber (1997) 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 Jiang et al. (2018) 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. 12. 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 Guo et al. (2018), Ren et al. (2018b), Chen et al. (2018), Dunnhofer et al. (2019), Ren et al. (2018a), Jiang et al. (2018), Jiang et al. (2019).

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

Table 6 Comparing various DRL-based image registration methods

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets
Image registration using uncertainty evaluation Lofth et al. (2013)	2013	DQN	Not specified	Probabilistic model using regression random forests (RRF) Breiman (1996)	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 Liao et al. (2017)	2017	DQN	12 actions: corresponding to different transformations	States: current transformation. Reward: distance error	5 Conv3D layers and 3 FC layers	Better success rate than TTK Ibanez et al. (2005), Quasi-global Miao et al. (2013) and Semantic registration Neumann et al. (2014)	Abdominal spine CT and CBCT dataset, Cardiac CT and CBCT
Multimodal image registration Ma et al. (2017)	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 Mnih et al. (2015), Dueling Wang et al. (2015b), etc.	Thorax and Abdomen (ABD) dataset
Robust non-rigid agent-based registration Krebs et al. (2017)	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 Lorenzi et al. (2013) and Elastix Klein et al. (2009)	MICCAI challenge PROMISE12 Litjens et al. (2014)

Table 6 (continued)

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets
Robust Multimodal registration Sun et al. (2018)	2018	Actor-Critic (a3c) Mnih et al. (2016a)	8 actions: for different transformations	States: fixed and moving image. Reward: Distance error. Monte-carlo method with LSTM Hochreiter and Schmidhuber (1997)	Multi layer CNN and FC layer	Comparable if not lower target registration error Fitzpatrick and West (2001) as compared to methods like SIFT Lowe (2004), Elastix Klein et al. (2009), Pure SL, RL-matrix, RL-LME, etc.	CT and MR images

by Thévenaz and Unser (2000). 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 Matsopoulos et al. (1999) through simulated annealing and through genetic algorithm Rouet et al. (2000). However, their cost of computation was very high. A CNN-based approach to this problem was suggested by Miao et al. (2016), and Dosovitskiy et al. (2015) proposed an optical flow method between 2D RGB images. A descriptor learned through a CNN was proposed by Wohlfart and Lepetit (2015), 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, Lotfi et al. (2013) 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) Breiman (1996) 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, Liao et al. (2017) 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.

A multi-modal method for image registration was proposed by Ma et al. (2017), 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 Krebs et al. (2017) 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 Sun et al. (2018), where the authors used a recurrent network with RL to solve the problem. Similar to Liao et al. (2017), 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 Hochreiter and Schmidhuber (1997) was used to encode past hidden states, Actor-critic Mnih et al. (2016a) for policy estimation,

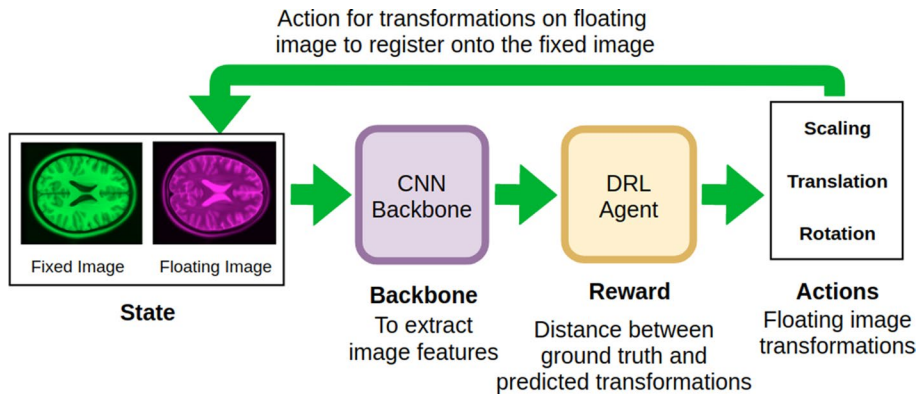


Fig. 13 DRL implementation for image registration. The state consists of fixed and floating image and the actions in form of transformations are performed on the floating image so as to maximize reward by minimizing distance between the ground truth and predicted transformations

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. 13. 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 Sun et al. (2018), Krebs et al. (2017), Liao et al. (2017).

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 Haralick and Shapiro (1985). With the improvement in detection techniques and introduction of CNN, new methods are introduced every year for image segmentation. Mask R-CNN He et al. (2017) extended the work by Faster R-CNN Ren et al. (2015) by adding a segmentation layer after the Bounding box has been predicted. Some earlier works include Girshick et al. (2014), Hariharan et al. (2014), Hariharan et al. (2015) 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 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 Sahba et al. (2006), 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. Sahba et al. (2007) also used a similar technique of Tabular method.

To overcome the constraints of the previous method for segmentation, Reza et al. (2016) 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 Hoiem et al. (2007) 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 Song et al. (2018), 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) Grady (2006) 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 Liao et al. (2020), where the authors proposed an iteratively-refined interactive multi-agent method for 3D medical image segmentation. They proposed a method to refine an initial coarse 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.

Further improvement in the results of medical image segmentation was proposed by Tian et al. (2020). 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 Mnih et al. (2016a) and similar to Deterministic policy gradient (DPG) Silver and Lever (2014). The authors used ResNet18 He et al. (2016) as backbone for actor and critic network along with batch normalisation Ioffe and Szegedy (2015) and weight normalization with Translated ReLU Xiang and Li (2017). 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

Table 7 Comparing various DRL-based image segmentation methods

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets
Semantic Segmentation for indoor scenes Reza et al. (2016)	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 Hoiem et al. (2007) for classification	Not Specified	Pixel-wise percentage jaccard index comparable to Gupta-L Gupta et al. (2014) and Gupta-P Gupta et al. (2013)	NYUD V2 dataset Silberman et al. (2012)
SeedNet Song et al. (2018)	2018	DQN, Double-DQN, Duel-DQN	800 actions: 2 per pixel	States: image features and segmentation mask. Reward: change in IOU. Random Walk (RW) Grady (2006) for segmentation algorithm	Multi layer CNN	Better IOU than methods like FCN Long et al. (2015) and iFCN Xu et al. (2016)	MSRA10K saliency dataset Cheng et al. (2014)
Iteratively refined multi agent segmentation Liao et al. (2020)	2020	Actor-critic (a3c) Mnih et al. (2016a)	1 action per voxel for adjusting segmentation probability	States: 3D image segmentation probability and hint map. Reward: cross entropy gain based framework	R-net Wang et al. (2018a)	Better performance than methods like MinCut Krähenbühl and Koltun (2011), DeepIGeoS (R-Net) Wang et al. (2018a) and InterCNN Breddel et al. (2018)	BraTS 2015 Menze et al. (2014), MM-WHS Zhuang and Shen (2016) and NCI-ISBI 2013 Challenge Bloch et al. (2013)

Table 7 (continued)

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets
Multi-step medical image segmentation Tian et al. (2020)	2020	Actor-critic (a3c) Mnih et al. (2016a)	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 Silver and Lever (2014)	ResNet18 He et al. (2016)	Higher Mean Dice score and lower Hausdorff distance than methods like Grab-Cut Rother et al. (2004), PSPNet Zhao et al. (2017), FCN Long et al. (2015), U-Net Ronneberger et al. (2015), etc.	Prostate MR image dataset (PROM-ISE12, ISBI2013) and retinal fundus image dataset (REFUGE challenge dataset Orlando et al. (2020))
Anomaly Detection in Images Chu et al. (2020)	2020	REINFORCE Williams (1992)	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 Bergmann et al. (2019) and Shi et al. (2016) on all metrics e.g. precision, recall and F1 when compared with U-Net Ronneberger et al. (2015) and baseline unsupervised method in Bergmann et al. (2019) but only wins on recall in Carrera et al. (2017)	MVTec AD Bergmann et al. (2019), NanoT-WICE Carrera et al. (2017), CrackForest Shi et al. (2016)

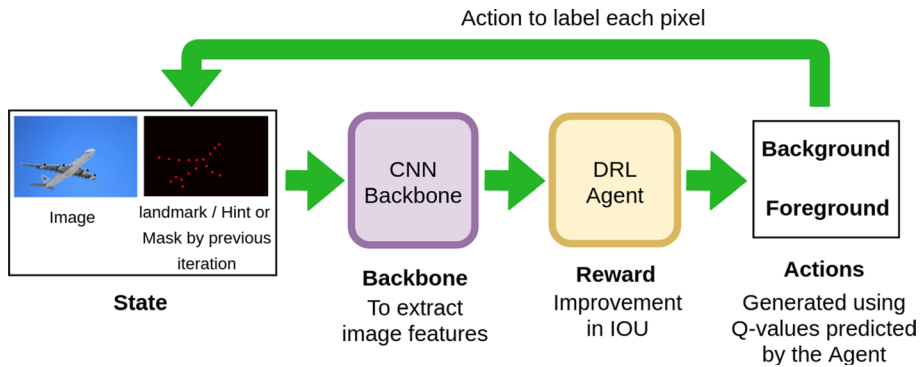


Fig. 14 DRL implementation for Image segmentation. The state consists of the image to be segmented along with a user hint for $t = 0$ or the segmentation mask by the previous iterations. The DRL agent performs actions by labeling each pixel as foreground and background to maximize the improvement in IOU over the iterations

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 Chu et al. (2020) 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). Chu et al. (2020) 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 Williams (1992) and the whole model is evaluated on multiple datasets e.g. MVTec AD Bergmann et al. (2019), NanoTWICE Carrera et al. (2017), CrackForest Shi et al. (2016).

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. 14. 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 Song et al. (2018), Liao et al. (2020).

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 (Papazoglou and Ferrari 2013; Xiao and Jae Lee 2016), weakly supervised (Cheng et al. 2017; Jain et al. 2017; Zhang et al. 2017b) and semi-supervised (Caelles et al. 2017; Jampani et al. 2017; Perazzi et al. 2017).

A DRL-based framework for video object segmentation was proposed by Sahba (2016), 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) Chen et al. (2015) was used for approximating the Q-values.

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 Liu and Jiang (2018). 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 Lea et al. (2017), Leal-Taixé et al. (2016) 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 Richard and Gall (2016) 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) Schulman et al. (2015) for training the policy and a spacial CNN Le et al. (2016a) to extract features.

Earlier approaches for video object segmentation required a large number of actions to complete the task. An Improvement was proposed by Han et al. (2018), 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 Jégou et al. (2017) 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 Goel et al. (2018), 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 Mnih et al. (2016a), Schulman et al. (2017b), Van Hasselt et al. (2016) for the network of first step. Following the unsupervised fashion, the authors used the approach similar to Vijayanarasimhan et al. (2017) 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 Jaderberg et al. (2015). They also used structural dissimilarity (DSSIM) Wang et al. (2004) to calculate reconstruction loss and actor-critic Mnih et al. (2016a) algorithm to learn policy in the second step.

A DRL method for dynamic semantic face video segmentation was proposed by Wang et al. (2020a), where Deep Feature Flow Zhu et al. (2017) 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 Ilg et al. (2017) 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.

Video object segmentation using human-provided location priors have been capable of producing promising results. An RL method for this task was proposed by Vecchio et al. (2020), 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 Huang et al. (2017) with down-sampling and up-sampling similar to U-Net Ronneberger et al. (2015) and an LSTM Hochreiter and Schmidhuber (1997) 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 Ilg et al. (2017)) 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 Tang et al. (2018). 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 Zhou et al. (2018a), 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 Szegedy et al. (2015) pre-trained on ImageNet Russakovsky et al. (2015), Deng et al. (2009) 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 Hochreiter and Schmidhuber (1997) to capture long-term video dependencies and used REINFORCE algorithm for training the policy function.

An improvement to Zhou et al. (2018a) was proposed by Zhou et al. (2018b), 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) Cho et al. (2014a) 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

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 Sahba (2016)	2016	Deep Belief Network Chen et al. (2015)	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 Liu and Jiang (2018)	2018	Trust Region Policy Optimization (TRPO) Schulman et al. (2015)	2 types: optimal step size and gesture class	States: TCN [Lea et al. (2017), Leal-Taixé et al. (2016)] and future frames. Reward: encourage larger steps and minimize action errors. Statistical language model Richard and Gall (2016) for gesture probability	Spatial CNN Le et al. (2016a)	Comparable accuracy, and higher edit and F1 scores as compared to methods like SD-SDL Sefati et al. (2015), Bidir LSTM DiPietro et al. (2016), LC-SC-CRF Le et al. (2016b), Seg-ST-CNN Le et al. (2016a), TCN Le et al. (2016c), etc.	JIGSAWS [Ahmidi et al. (2017), Gao et al. (2014)] benchmark dataset Available Code
Cutting agent for video object segmentation Han et al. (2018)	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 Jégou et al. (2017)	Higher mean region similarity, counter accuracy and temporal stability Perazzi et al. (2016) as compared to methods like MSK Perazzi et al. (2017), ARP Jun Koh and Kim (2017), CTN Jang and Kim (2017), VPN Jampant et al. (2017), etc.	DAVIS dataset Perazzi et al. (2016) and the YouTube Objects dataset Jain et al. (2014), Prest et al. (2012)

Table 8 (continued)

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets and Source code
Unsupervised video object segmentation (MOREL) Goel et al. (2018)	2018	Actor-critic (a2c) Mnih et al. (2016a)	Not specified	States: consecutive frames. Two step process with optical flow using Spatial Transformer Networks Jaderberg et al. (2015) and reconstruction loss using structural dissimilarity Wang et al. (2004)	Multi-layer CNN	Higher total episodic reward as compared to methods that used actor-critic without MOREL	59 Atari games. Available Code
Face video segmentation Wang et al. (2020a)	2020	Not specified	2 actions: categorising a frame as a key or anon-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 than other methods like DVSNet Xu et al. (2018), DFF Zhu et al. (2017)	300VW dataset Shen et al. (2015) and Cityscape dataset Cordts et al. (2016)

Table 8 (continued)

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets and Source code
Multi-agent Video Object Segmentation Vecchio et al. (2020)	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 Ilg et al. (2017) from previous frame and action history. Reward: clicks generated by gamification. Down-sampling and up-sampling similar to U-Net Ronneberger et al. (2015)	DenseNet Huang et al. (2017)	Higher mean region similarity and contour accuracy Perazzi et al. (2016) as compared to semi-supervised methods such as SeamSeg Avinash Ramakanth and Venkatesh Babu (2014), BSVS Märki et al. (2016), VSOF Tsai et al. (2016), OSVOS Caelles et al. (2017) and weakly-supervised methods such as GVOS Spampinato et al. (2016), Spftn Zhang et al. (2017b)	DAVIS-17 dataset Perazzi et al. (2016)

Table 8 (continued)

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets and Source code
Skeleton-based Action Recognition Tang et al. (2018)	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 Shahroudy et al. (2016), and higher accuracy for SYSU-3D Hu et al. (2015) and UT-Kinect Dataset Xia et al. (2012)	NTU+RGBD Shahroudy et al. (2016), SYSU-3D Hu et al. (2015) and UT-Kinect Dataset Xia et al. (2012)

Table 8 (continued)

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets and Source code
Video summarisation Zhou et al. (2018a)	2018	DQN	2 actions: selecting and rejecting the frame	tates: bidirectional LSTM Huang et al. (2015) produced states by input frame features. Reward: Diversity-Representativeness Reward Function	GoogLeNet Szegedy et al. (2015)	Higher F-score Zhang et al. (2016b) as compared to methods like Uniform sampling, K-medoids, Dictionary selection Elhamifar et al. (2012), Video-MMR Li and Merialdo (2010), Vsumm De Avila et al. (2011), etc.	TVSum Song et al. (2015) and SumMe Gygli et al. (2014). Available Code
Video summarization Zhou et al. (2018b)	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 Huang et al. (2015)	GoogLeNet Szegedy et al. (2015)	Higher F-score Zhang et al. (2016b) as compared to methods like Dictionary selection Elhamifar et al. (2012), GAN Mahasseni et al. (2017), DR-DSN Zhou et al. (2018a), Backprop-Grad Panda et al. (2017), etc. in most cases	TVSum Song et al. (2015) and CoSum Chu et al. (2015) datasets. Available Code

Table 8 (continued)

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets and Source code
Video summarization in Ultrasound Liu et al. (2020c)	2020	Not specified	2 actions: selecting and rejecting the frame	States: frame latent scores Reward: R_{der} , R_{rep} and R_{div} , bidirectional LSTM Huang et al. (2015) and Kernel temporal segmentation Potapov et al. (2014)	Not specified	Higher F1-scores in supervised and unsupervised fashion as compared to methods like FCNN Rochan et al. (2018) and DR-DSN Zhou et al. (2018a)	Fetal Ultrasound Kirwan (2010)

the task of either keeping the frame or discarding it. They used 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 Zhou et al. (2018a). The authors trained both the networks using the features generated by GoogLeNet Szegedy et al. (2015) pre-trained on ImageNet Deng et al. (2009).

A method for video summarization in Ultrasound using DRL was proposed by Liu et al. (2020c), in which the authors proposed a deep summarisation network in an encoder-decoder fashion and used a bidirectional LSTM (Bi-LSTM) Huang et al. (2015) 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) Potapov et al. (2014) 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. 15, 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 Zhou et al. (2018a) and Zhou et al. (2018b) used diversity representativeness reward function and Liu et al. (2020c) 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 Matas et al. (2018), where the authors used a modified version of Deep Deterministic Policy Gradients (DDPG) Lillicrap et al. (2015b). They used the simulator Pybullet Coumans and Bai (2016) 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 Hong et al. (2018), 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 Chen et al. (2017) and ICNet Zhao et al. (2018), pre-trained on ADE20K Zhou et al. (2017)

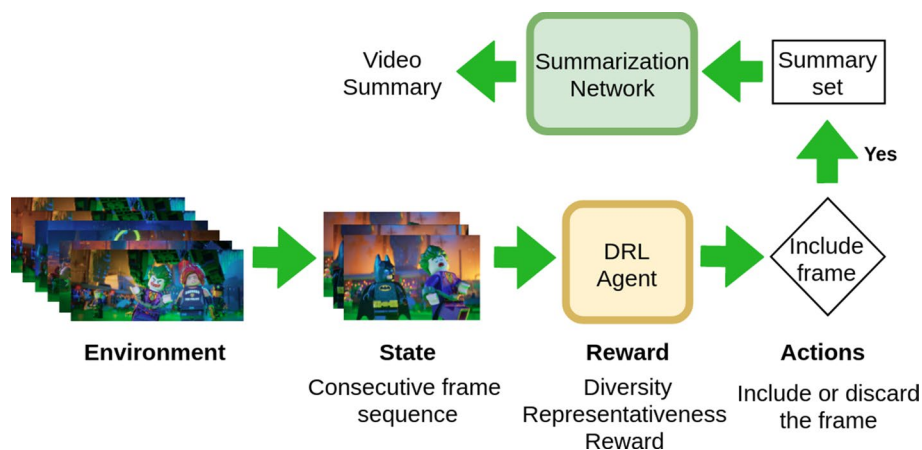


Fig. 15 DRL implementation for video summarization. For state a sequence of consecutive frames are used and the DRL agent decided whether to include the frame in the summary set that is used to predict video summary

and Cityscape Cordts et al. (2016), 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 Mnih et al. (2016a) 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 Dai et al. (2019), where the authors used 2D grey-scale images as the environment. They considered a full resolution $11\text{px} \times 11\text{px}$ window and a $21\text{px} \times 21\text{px}$ window down-scaled to $11\text{px} \times 11\text{px}$ 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 than 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 Mnih et al. (2016a) framework to estimate value and policy functions.

An RL method for automatic diagnosis of acute appendicitis in abdominal CT images was proposed by Al et al. (2019), 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 Mnih et al. (2016a) framework to estimate policy and value functions.

Painting using an algorithm is a fantastic yet challenging task in the computer vision field. An automated painting method was proposed by Huang et al. (2019b), 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 Goodfellow et al. (2014) were employed to improve image quality at pixel-level and DDPG Lillicrap et al. (2015b) 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) Gulrajani (2017) 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 Mnih et al. (2016a) fashion with backbone similar to Resnet18 He et al. (2016), 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 Hannes et al. (2020). 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 He et al. (2016) 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 Liu et al. (2020a), 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_t , and the actions space was defined similar to scale weighing and money system Van Hove (2001) 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 Caicedo and Lazebnik (2015)) 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 Wang et al. (2020b). 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_{10}^3) 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 Barron (2019). For performing the action of bracketing selection EBSNet consisted of a semantic branch using AlexNet Krizhevsky et al. (2017) 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 Gharbi et al. (2017).

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 Toromanoff et al. (2020). 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 Dosovitskiy et al. (2017) 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 He et al. (2016) 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 Hessel et al. (2017) 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, Wang and Deng (2020) 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, Li and Chen (2020) proposed to train

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 and Source code
Object manipulation Matas et al. (2018)	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 Coumans and Bai (2016). Available Code
Visual based control Hong et al. (2018)	2018	Actor-critic (a3c) Mnih et al. (2016a)	3 actions: forward, turn right and turn left	State: output by backbones. Reward: 0.001 at each time-step	DeepLab Chen et al. (2017) and ICNet Zhao et al. (2018)	Higher success and lower collision rate then ResNet-A3C and Depth-A3C	Unity3D engine
Automatic tracing Dai et al. (2019)	2019	Actor-critic Mnih et al. (2016a)	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 Peng et al. (2010) and APP2 neuron tracer Xiao and Peng (2013)	Synthetic and micro-copy dataset Bass et al. (2017)
Automatic diagnosis (RLE) Al et al. (2019)	2019	Actor-critic Mnih et al. (2016a)	6 actions: 2 per axis	State: 50 × 50 × 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

Table 9 (continued)

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets and Source code
Learning to paint Huang et al. (2019b)	2019	Actor-critic with DDPG	Actions control the stroke parameter: location, shape, color and transparency	State: Reference image, Drawing canvas and time step. Reward: change in discriminator score (calculated by WGAN-GP Gulrajani (2017) across an action. GANs Goodfellow et al. (2014) to improve image quality	ResNet18 He et al. (2016)	Able to replicate the original images to a large extent, and better resemblance to the original image as compared to SPIRAL Ganin et al. (2018) with same number of brush strokes	MNIST LeCun (1998), SVHN Netzer et al. (2011), CelebA Liu et al. (2015) and ImageNet Russakovsky et al. (2015). Available Code
Guiding medical robots Hannes et al. (2020)	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 He et al. (2016)	Higher % of policy correctness and reachability as compared to CNN Classifier, where MS-DQN showed the best results	Ultrasound Images Dataset. Available Code

Table 9 (continued)

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets and Source code
Crowd counting Liu et al. (2020a)	2020	DQN	9 actions: 10, 5, 2, 1, +1, +2, +5, +10 and end	State: weight vector W_t and image feature vector FV_t . Reward: Intermediate reward and endingreward	VGG16 Simonyan and Zisserman (2014)	Lower/comparable mean squared error (MSE) and mean absolute error (MAE) as compared to other methods like DRSAN Liu et al. (2018), PGCNet Yan et al. (2019), MBTTBF Sindagi and Patel (2019), S-DCNet Xiong et al. (2019), CAN Liu et al. (2019), etc.	The ShanghaiTech (SHT) Dataset Zhang et al. (2016c), The UCFC50 Dataset Idrees et al. (2013) and The UCF-QNRF Dataset Idrees et al. (2018), Available Code
Automated Exposure bracketing Wang et al. (2020b)	2020	Not Specified	selecting optimal bracketing from candidates	State: quality of generated HDR image. Reward: improvement in peak signal to noise ratio	AlexNet Krizhevsky et al. (2017)	Higher peak signal to noise ratio as compared to other methods like Barakat Barakat et al. (2008), Pourreza-Shahri Pourreza-Shahri and Kehtarnavaz (2015), Beek van Beek (2018), etc.	Proposed benchmark dataset. Available Code/data

Table 9 (continued)

Approaches	Year	Training Technique	Actions	Remarks	Backbone	Performance	Datasets and Source code
Urban Autonomous driving Toromanoff et al. (2020)	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 He et al. (2016)	Won the 2019 camera only CARLA challenge Ros et al. (2019)	CARLA urban driving simulator Ros et al. (2019) Available Code
Mitigating bias in Facial Recognition Wang and Deng (2020)	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 Wang et al. (2018b) and ArcFace Deng et al. (2019)	RFW Wang et al. (2019a) and proposed novel datasets; BUPT-Globalface and BUPT-Balanced-face Available Data
Attention mechanism to improve CNN performance Li and Chen (2020)	2020	DQN Mnih et al. (2015)	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 He et al. (2016)	Improves the performances of Hu et al. (2018), Lee et al. (2019) and Woo et al. (2018), which attend on feature channel, spatial-channel and style, respectively	ImageNet Deng et al. (2009)

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. 16 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, Hong et al. (2018).

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.

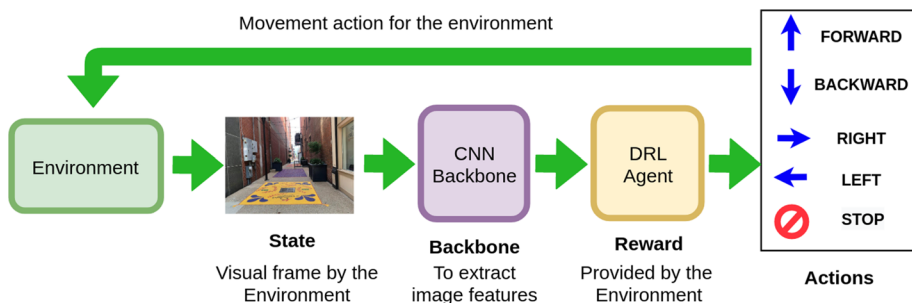


Fig. 16 A general DRL implementation for agent movement with visual inputs. The state is provided by the environment based on which the agent performs movement actions to get a new state and a reward from the environment

- **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 Krebs et al. (2017) 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 Dulac-Arnold et al. (2020). In this work, they designed a set of experiments and analyzed their effects on common RL agents. Open-sourcing an environmental suite, [realworldrl-suite](#) Dulac-Arnold et al. (2020) 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 Ng and Russell (2000), Abbeel and Ng (2004), Yang et al. (2020), Duong et al. (2019) a specific form of imitation learning Osa et al. (2018), 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 Abbeel and Ng (2004). Recently, to analyze complex human movement and control high-dimensional robot systems, Li et al. (2018c) proposed an online inverse RL algorithm. You et al. 2019 (2019) 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 (Brown and Sandholm 2019; Vinyals et al. 2019), robotics (Kober et al. 2013), and autonomous driving (Shalev-Shwartz et al. 2016), stock trading (Lee et al. 2007), social science (Leibo et al. 2017), 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 (Busoniu et al. 2008). Learning in a multi-agent scenario is more difficult than a single-agent scenario because non-stationarity (Hernandez-Leal et al. 2017), multi-dimensionality (Busoniu et al. 2008), credit assignment (Agogino and Tumer 2004), 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 (ubramanian and Mahajan 2019).
- **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 (Schweighofer and Doya 2003; Wang et al. 2016) are studied to enable agents to learn new skills from small amounts of experience. Recently, there is a research interest in meta RL (Nagabandi et al. 2018; Gupta et al. 2018; Sæmundsson et al. 2018; Rakelly et al. 2019; Liu et al. 2019), each using a different approach. For benchmarking and evaluation of meta RL algorithms, Yu et al. (2020) 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. Stadie et al. (2017) 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. (2018) proposed behavior cloning Loss to increase imitating the demonstrations. Besides Q-learning, Generative Adversarial Imitation Learning (Tsurumine et al. 2019) 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 (Tsurumine et al. 2019).

13 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.

Acknowledgements This material is based upon work supported by the National Science Foundation under Award No OIA-1946391.

References

- Aaron W, Alan F, Prasad T (2014) Using trajectory data to improve bayesian optimization for reinforcement learning. *J Mach Learn Res* 15(8):253–282
- Abeel P, Ng AY (2004) Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on machine learning*, pp 1–8. Association for Computing Machinery
- Adam C, Pieter A, Andrew YN (2009) Apprenticeship learning for helicopter control. *Commun ACM* 52(7):97–105
- Agogino AK, Tumer K (2004) Unifying temporal and structural credit assignment problems. In *Proceedings of the third international joint conference on autonomous agents and multiagent systems—vol 2*, pp 980–987. IEEE Computer Society
- Al WA, Yun ID (2019) Partial policy-based reinforcement learning for anatomical landmark localization in 3d medical images. *IEEE Trans Med Image*
- Al WA, Yun Io, Lee KJ (2019) Reinforcement learning-based automatic diagnosis of acute appendicitis in abdominal ct. *arXiv preprint arXiv:1909.00617*
- Alaniz S (2018) Deep reinforcement learning with model learning and monte carlo tree search in minecraft. In *Conference on reinforcement learning and decision making*
- Amir A, Ozan O, Yuanwei L, Loic LF, Benjamin H, Ghislain V, Konstantinos K, Athanasios V, Ben G, Bernhard K et al (2019) Evaluating reinforcement learning agents for anatomical landmark detection. *Med Image Anal* 53:156–164
- 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
- Andersson O, Heintz F, Doherty P (2015) Model-based reinforcement learning in continuous environments using real-time constrained optimization. In *AAAI*
- Arulkumaran K, Deisenroth MP, Brundage M, Bharath AA (2017) A brief survey of deep reinforcement learning. *arXiv preprint arXiv:1708.05866*
- Avinash Ramakanth S, Venkatesh Babu R (2014) Seamseg: Video object segmentation using patch seams. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 376–383
- Ayle M, Tekli J, El-Zini J, El-Asmar B, Awad M (2020) Bar-a reinforcement learning agent for bounding-box automated refinement
- Babaeizadeh M, Frosio I, Tyree S, Clemons J, Kautz J (2016) GA3C: gpu-based A3C for deep reinforcement learning. *arxiv:CoRR:abs/1611.06256*
- Babenko B, Yang M-H, Belongie S (2009) Visual tracking with online multiple instance learning. In *2009 IEEE conference on computer vision and pattern recognition*, pp 983–990. IEEE
- Bae S-H, Yoon K-J (2014) 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*, pp 1218–1225
- Bagnell J (2012) Learning decision: Robustness, uncertainty, and approximation. 04

- Bagnell JA, Schneider JG (2001) Autonomous helicopter control using reinforcement learning policy search methods. In Proceedings 2001 ICRA. IEEE international conference on robotics and automation (Cat. No.01CH37164), vol 2, pp 1615–1620
- Barron JT (2019) A general and adaptive robust loss function. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4331–4339
- Bellver M, Giró-i Nieto X, Marqués F, Torres J (2016) Hierarchical object detection with deep reinforcement learning. arXiv preprint [arXiv:1611.03718](https://arxiv.org/abs/1611.03718)
- Bergmann P, Fauser M, Sattlegger D, Steger C (2019) Mvtec ad a comprehensive real-world dataset for unsupervised anomaly detection. In 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 9584–9592
- Bertinetto L, Valmadre J, Henriques JF, Vedaldi A, Torr Philip HS (2016) Fully-convolutional siamese networks for object tracking. In European conference on computer vision, pp 850–865. Springer
- Black MJ, Yacoob Y (1995) 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, pp 374–381. IEEE
- Bloch N, Madabhushi A, Huisman H, Freymann J, Kirby J, Grauer M, Enquobahrie A, Jaffe C, Clarke L, Farahani K (2013) challenge: automated segmentation of prostate structures. *Cancer Imag Arch* 370:2015
- Boedeker J, Springenberg JT, Wlfling J, Riedmiller M (2014) Approximate real-time optimal control based on sparse gaussian process models. In 2014 IEEE symposium on adaptive dynamic programming and reinforcement learning (ADPRL), pp 1–8
- Brazil G, Liu X (2019) M3d-rpn: Monocular 3d region proposal network for object detection. In Proceedings of the IEEE international conference on computer vision, Seoul, South Korea,
- Bredell G, Tanner C, Konukoglu E (2018) Iterative interaction training for segmentation editing networks. In International workshop on machine learning in medical imaging, pp 363–370. Springer
- Buetti-Dinh A, Galli V, Bellenberg S, Ilie O, Herold M, Christel S, Boretska M, Pivkin Igor V, Wilmes P, Sand W, Vera M, Dopson M (2019) Deep neural networks outperform human experts capacity in characterizing bioleaching bacterial biofilm composition. *Biotechnol Rep* 22:e00321
- Busoniu L, Babuska R, De Schutter B (2008) A comprehensive survey of multiagent reinforcement learning. *IEEE Trans Syst Man Cybern C* 38(2):156–172
- Caelles S, Maninis K-K, Pont-Tuset J, Leal-Taixé L, Cremers D, Van Gool L (2017) One-shot video object segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 221–230
- Caicedo JC, Lazebnik S (2015) Active object localization with deep reinforcement learning. In Proceedings of the IEEE international conference on computer vision, pp 2488–2496
- Carrera D, Manganini F, Boracchi G, Lanzarone E (2017) Defect detection in sem images of nanofibrous materials. *IEEE Trans Ind Inf* 13(2):551–561
- Carsten R, Vladimir K, Andrew B (2004) ‘grabcut’ interactive foreground extraction using iterated graph cuts. *ACM Trans Graph* 23(3):309–314
- Chen B, Wang D, Li P, Wang S, Lu H (2018) Real-time ‘actor-critic’ tracking. In Proceedings of the European conference on computer vision (ECCV), pp 318–334
- Cheng J, Tsai Y-H, Wang S, Yang M-H (2017) Segflow: Joint learning for video object segmentation and optical flow. In Proceedings of the IEEE international conference on computer vision, pp 686–695
- Cher B, Pyry H, Vincenzo DP, Claudia C, Anthony BA (2017) Detection of axonal synapses in 3d two-photon images. *PLoS ONE* 12(9):e0183309
- Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint [arXiv:1406.1078](https://arxiv.org/abs/1406.1078)
- Cho K, van Merriënboer B, Gülçehre Ç, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. [arxiv:CoRR:abs/1406.1078](https://arxiv.org/abs/1406.1078)
- Choi J, Jin Chang H, Yun S, Fischer T, Demiris Y, Young Choi J (2017) Attentional correlation filter network for adaptive visual tracking. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4807–4816
- Choi W (2015) Near-online multi-target tracking with aggregated local flow descriptor. In Proceedings of the IEEE international conference on computer vision, pp 3029–3037
- Chorowski J, Bahdanau D, Serdyuk D, Cho KH, Bengio Y (2015) Attention-based models for speech recognition. [arxiv:CoRR:abs/1506.07503](https://arxiv.org/abs/1506.07503)
- Chu Q, Ouyang W, Li H, Wang X, Liu B, Yu N (2017) 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, pp 4836–4845

- Chu W-H, Kitani KM (2020) Neural batch sampling with reinforcement learning for semi-supervised anomaly detection. In European conference on computer vision, pp 751–766
- Chu W-S, Song Y, Jaimes A (2015) Video co-summarization: video summarization by visual co-occurrence. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3584–3592
- Clavera I, Rothfuss J, Schulman J, Fujita Y, Asfour T, Abbeel P (2018) Model-based reinforcement learning via meta-policy optimization. [arxiv:CoRR:abs/1809.05214](https://arxiv.org/abs/1809.05214)
- Comaniciu D, Ramesh V, Meer P (2000) 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), vol 2, pp 142–149. IEEE
- Concetto S, Simone P, Daniela G (2016) Gamifying video object segmentation. IEEE Trans Pattern Anal Mach Intell 39(10):1942–1958
- Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, Franke U, Roth S, Schiele B (2016) The cityscapes dataset for semantic urban scene understanding. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3213–3223
- Coulom R (2006) Efficient selectivity and backup operators in monte-carlo tree search. In Proceedings of the 5th international conference on computers and games, pp 72–83
- Coumans E, Bai Y (2016) Pybullet, a python module for physics simulation for games, robotics and machine learning
- Craig Jordan V (1990) Long-term adjuvant tamoxifen therapy for breast cancer. Breast Cancer Res Treat 15(3):125–136
- Criminisi A, Shotton J, Robertson D, Konukoglu E (2010) Regression forests for efficient anatomy detection and localization in ct studies. In International MICCAI workshop on medical computer vision, pp 106–117. Springer
- Dai T, Dubois M, Arulkumaran K, Campbell J, Bass C, Billot B, Uslu F, De Paola V, Clopath C, Bharath AA (2019) Deep reinforcement learning for subpixel neural tracking. In International conference on medical imaging with deep learning, pp 130–150
- Danelljan Martin, Bhat Goutam, Shahbaz Khan Fahad, Felsberg Michael (2017) Eco: efficient convolution operators for tracking. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6638–6646
- Danelljan M, Hager G, Shahbaz Khan F, Felsberg M (2015) Learning spatially regularized correlation filters for visual tracking. In Proceedings of the IEEE international conference on computer vision, pp 4310–4318
- Darryl MC, Andrew M, Adnan T, Dominic K, Stuart C (2014) Fully automatic lesion segmentation in breast mri using mean-shift and graph-cuts on a region adjacency graph. J Magn Reson Imaging 39(4):795–804
- David S, Guy L, Heess N, Wierstra D, Riedmiller M (2014) Deterministic policy gradient algorithms, Thomas Degris
- Deisenroth MP, Englert P, Peters J, Fox D (2014) Multi-task policy search for robotics. In 2014 IEEE international conference on robotics and automation (ICRA), pp 3876–3881
- Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) Imagenet: a large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pp 248–255. IEEE
- Deng J, Guo J, Xue N, Zafeiriou S (2019) Arcface: Additive angular margin loss for deep face recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4690–4699
- Denzler J, Paulus DWR (1994) Active motion detection and object tracking. In Proceedings of 1st international conference on image processing, vol 3, pp 635–639. IEEE
- Depraetere B, Liu M, Pinte G, Grondman I, Babuka R (2014) Comparison of model-free and model-based methods for time optimal hit control of a badminton robot. Mechatronics 24(8):1021–1030
- De Asis K, Hernandez-Garcia JF, Holland GZ, Sutton RS (2018) Multi-step reinforcement learning: a unifying algorithm. In Thirty-Second AAAI conference on artificial intelligence
- DiPietro R, Lea C, Malpani A, Ahmidi N, Vedula SS, Lee GI, Lee MR, Hager GD (2016) Recognizing surgical activities with recurrent neural networks. In International conference on medical image computing and computer-assisted intervention, pp 551–558. Springer
- Dollár P, Wojek C, Schiele B, Perona P (2009) Pedestrian detection: a benchmark. In 2009 IEEE conference on computer vision and pattern recognition, pp 304–311. IEEE
- Dominik N, Saša G, Matthias J, Nassir N, Joachim H, Razvan I (2014) Probabilistic sparse matching for robust 3d/3d fusion in minimally invasive surgery. IEEE Trans Med Imaging 34(1):49–60
- Don M, Anup B (1994) Motion tracking with an active camera. IEEE Trans Pattern Anal Mach Intell 16(5):449–459

- Donahue J, Hendricks LA, Guadarrama S, Rohrbach M, Venugopalan S, Saenko K, Darrell T (2014) Long-term recurrent convolutional networks for visual recognition and description. [arXiv:CoRR:abs/1411.4389](#)
- Dosovitskiy A, Fischer P, Ilg E, Hausser P, Hazirbas C, Golkov V, Van Der Smagt P, Cremers D, Brox T (2015) Flownet: Learning optical flow with convolutional networks. In Proceedings of the IEEE international conference on computer vision, pp 2758–2766
- Dosovitskiy A, Ros G, Codevilla F, Lopez A, Koltun V (2017) Carla: an open urban driving simulator. [arXiv preprint arXiv:1711.03938](#)
- Du Y, Wang W, Wang L (2015) Hierarchical recurrent neural network for skeleton based action recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1110–1118
- Dulac-Arnold G, Levine N, Mankowitz DJ, Li J, Paduraru C, Gowal S, Hester T (2020) An empirical investigation of the challenges of real-world reinforcement learning
- Dunnhofer M, Martinel N, Luca Foresti G, Micheloni C (2019) Visual tracking by means of deep reinforcement learning and an expert demonstrator. In Proceedings of the IEEE international conference on computer vision workshops
- Duong CN, Quach KG, Jalata I, Le N, Luu K (2019) Mobiface: a lightweight deep learning face recognition on mobile devices. In 2019 IEEE 10th international conference on biometrics theory, applications and systems (BTAS), pp 1–6. IEEE
- Duong CN, Quach KG, Luu K, Hoang LT, Savvides M, Bui TD (2019) Learning from longitudinal face demonstration—where tractable deep modeling meets inverse reinforcement learning. 127(6–7)
- Eddy I, Nikolaus M, Tonmoy S, Margret K, Alexey D, Thomas B (2017) Flownet 2.0: Evolution of optical flow estimation with deep networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2462–2470
- El-Fakdi A, Carreras M (2008) Policy gradient based reinforcement learning for real autonomous underwater cable tracking. In 2008 IEEE/RSJ international conference on intelligent robots and systems, pp 3635–3640
- Elhamifar E, Sapiro G, Vidal R (2012) See all by looking at a few: Sparse modeling for finding representative objects. In 2012 IEEE conference on computer vision and pattern recognition, pp 1600–1607. IEEE
- Erhan D, Szegedy C, Toshev A, Anguelov D (2014) Scalable object detection using deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2147–2154
- Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A (2007) The pascal visual object classes challenge 2007 (voc2007) results
- Everingham M, Winn J (2011) The pascal visual object classes challenge 2012 (voc2012) development kit. Pattern analysis, statistical modelling and computational learning, Tech. Rep, 8, 2011
- Fan H, Lin L, Yang F, Chu P, Deng G, Yu S, Bai H, Xu Y, Liao C, Ling H (2019) Lasot: a high-quality benchmark for large-scale single object tracking. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5374–5383
- Fan H, Ling H (2017) Parallel tracking and verifying: a framework for real-time and high accuracy visual tracking. In Proceedings of the IEEE international conference on computer vision, pp 5486–5494
- Felix H, Antoine B, Sumit C, Jason W (2015) The goldilocks principle: Reading children's books with explicit memory representations. [arXiv:CoRR:abs/1511.02301](#)
- Finn C, Tan XY, Duan Y, Darrell T, Levine S, Abbeel P (2016) Deep spatial autoencoders for visuomotor learning. In: Kragic D, Bicchi A, De Luca A (eds) 2016 IEEE international conference on robotics and automation, ICRA 2016. Stockholm, Sweden, pp 512–519
- Florin-Cristian G, Bogdan G, Yefeng Z, Sasa G, Andreas M, Joachim H, Dorin C (2017) Multi-scale deep reinforcement learning for real-time 3d-landmark detection in ct scans. IEEE Trans Pattern Anal Mach Intell 41(1):176–189
- FlorinC G, Edward K, Bogdan G, Vivek S, Yefeng Z, Joachim H, Dorin C (2016) Marginal space deep learning: efficient architecture for volumetric image parsing. IEEE Trans Med Imaging 35(5):1217–1228
- Fontes DASE, Brandão LAP, da Antonio L Jr, de Albuquerque Araújo A, (2011) Vsumm: a mechanism designed to produce static video summaries and a novel evaluation method. Pattern Ecogn Lett 32(1):56–68
- François-Lavet V, Henderson P, Islam R, Bellemare MG, Pineau J (2018) An introduction to deep reinforcement learning. [arXiv preprint arXiv:1811.12560](#)
- Ganin Y, Kulkarni T, Babuschkin I, Eslami SM, Vinyals O (2018) Synthesizing programs for images using reinforced adversarial learning. [arXiv preprint arXiv:1804.01118](#)

- Gao H, Zhuang L, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4700–4708
- Gao M, Yu R, Li A, Morariu VI, Davis LS (2018) Dynamic zoom-in network for fast object detection in large images. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6926–6935
- Gao Y, Vedula SS, Reiley CE, Ahmidi N, Varadarajan B, Lin HC, Tao L, Zappella L, Béjar B, Yuh DD, et al (2014) Jhu-isi gesture and skill assessment working set (jigsaws): a surgical activity dataset for human motion modeling. In Miccai workshop: M2cai, vol 3, pp 3, 2014
- Gauriau R, Cuingnet R, Lesage D, Bloch I (2014) Multi-organ localization combining global-to-local regression and confidence maps. In International conference on medical image computing and computer-assisted intervention, pp 337–344. Springer
- Geiger A, Lenz P, Urtasun R (2012) Are we ready for autonomous driving? the kitti vision benchmark suite. In 2012 IEEE conference on computer vision and pattern recognition, pp 3354–3361
- Giles M (2017) Mit technology review. Google researchers have reportedly achieved ‘quantum supremacy’. <http://www.technologyreview.com/f/614416>
- Girshick R (2015) Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, pp 1440–1448
- Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 580–587
- Gkioxari G, Girshick R, Malik J (2015) Contextual action recognition with r* cnn. In Proceedings of the IEEE international conference on computer vision, pp 1080–1088
- Gl M, Chen J, Barron JT, Hasinoff Samuel W, Durand F (2017) Deep bilateral learning for real-time image enhancement. ACM Trans Graph 36(4):1–12
- Goel V, Weng J, Poupart P (2018) Unsupervised video object segmentation for deep reinforcement learning. In Advances in neural information processing systems, pp 5683–5694
- Gonzalez-Garcia A, Vezhnevets A, Ferrari V (2015) An active search strategy for efficient object class detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3022–3031
- Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In Advances in neural information processing systems, pp 2672–2680
- Graves A, Mohamed A, Hinton GE (2013) Speech recognition with deep recurrent neural networks. [arxiv:CoRR:abs/1303.5778](https://arxiv.org/abs/1303.5778)
- Gubern-Mérida A, Martí R, Melendez J, Hauth JL, Mann RM, Karssemeijer N, Platel B (2015) Automated localization of breast cancer in dce-mri. Med Image Anal 20(1):265–274
- Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V (2017) and Aaron C Courville. Improved training of wasserstein gans. In Advances in neural information processing systems, pp 5767–5777
- Guo M, Lu J, Zhou J (2018) Dual-agent deep reinforcement learning for deformable face tracking. In Proceedings of the European conference on computer vision (ECCV), pp 768–783
- Gupta A, Mendonca R, Liu YX, Abbeel P, Levine S (2018) Meta-reinforcement learning of structured exploration strategies. In Advances in neural information processing systems, pp 5302–5311
- Gupta S, Arbelaez P, Malik J (2013) Perceptual organization and recognition of indoor scenes from rgb-d images. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 564–571
- Gupta S, Girshick R, Arbeláez P, Malik J (2014) Learning rich features from rgb-d images for object detection and segmentation. In European conference on computer vision, pp 345–360. Springer
- Gygli M, Grabner H, Riemenschneider H, Van Gool L (2014) Creating summaries from user videos. In European conference on computer vision, pp 505–520. Springer
- Hamid Rezatofighi S, Milan A, Zhang Z, Shi Q, Dick A, Reid I (2015) Joint probabilistic data association revisited. In Proceedings of the IEEE international conference on computer vision, pp 3047–3055
- Han J, Yang L, Zhang D, Chang X, Liang X (2018) Reinforcement cutting-agent learning for video object segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 9080–9089
- Hang X, Hanchuan P (2013) App2: automatic tracing of 3d neuron morphology based on hierarchical pruning of a gray-weighted image distance-tree. Bioinformatics 29(11):1448–1454
- Haralick RM, Shapiro LG (1985) Image segmentation techniques. Computer Vision, Graphics, and Image Processing 29(1):100–132

- Hare S, Golodetz S, Saffari A, Vineet V, Cheng M-M, Hicks SL, Torr PHS (2015) Struck: structured output tracking with kernels. *IEEE Trans Pattern Anal Mach Intell* 38(10):2096–2109
- Hariharan B, Arbeláez P, Girshick R, Malik J (2014) Simultaneous detection and segmentation. In *European conference on computer vision*, pp 297–312. Springer
- Hariharan B, Arbeláez P, Girshick R, Malik J (2015) Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 447–456
- Haroon I, Imran S, Cody S, Mubarak S (2013) Multi-source multi-scale counting in extremely dense crowd images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 2547–2554
- Haroon I, Muhammad T, Kishan A, Dong Z, Somaya A-M, Nasir R, Mubarak S (2018) Composition loss for counting, density map estimation and localization in dense crowds. In *Proceedings of the European conference on computer vision (ECCV)*, pp 532–546
- Hase H, Azampour MF, Tirindelli M, Paschali M, Simson W, Fatemizadeh E, Navab N (2020) Ultrasound-guided robotic navigation with deep reinforcement learning. *arXiv preprint arXiv:2003.13321*
- Hasselt HV (2010) Double q-learning. In *Advances in neural information processing systems*, pp 2613–2621
- Hausknecht MJ, Stone P (2015) Deep recurrent q-learning for partially observable mdps. [arxiv:CoRR:abs/1507.06527](https://arxiv.org/abs/1507.06527)
- He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp 2961–2969
- He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 770–778
- Henriques JF, Caseiro R, Martins P, Batista J (2014) High-speed tracking with kernelized correlation filters. *IEEE Trans Pattern Anal Mach Intell* 37(3):583–596
- Hernandez-Leal P, Kaisers M, Baarslag T, de Cote EM (2017) A survey of learning in multiagent environments: Dealing with non-stationarity. [arxiv:CoRR:abs/1707.09183](https://arxiv.org/abs/1707.09183)
- Le Hoang NT, Duong CN, Han L, Luu K, Quach KG, Savvides M (2018) Deep contextual recurrent residual networks for scene labeling. *Pattern Recogn* 80:32–41
- Le Hoang NT, Quach KG, Luu K, Duong CN, Savvides M (2018) Reformulating level sets as deep recurrent neural network approach to semantic segmentation. *IEEE Trans Image Process* 27(5):2393–2407
- Hoiem D, Efros AA, Hebert M (2007) Recovering surface layout from an image. *Int J Comput Vision* 75(1):151–172
- Holliday JB, Le Ngan TH (2020) Follow then forage exploration: improving asynchronous advantage actor critic. In *International conference on soft computing, artificial intelligence and applications (SAI 2020)*, pp 107–118
- Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on computer vision and pattern recognition*, pp 7132–7141
- Hu JF, Zheng WS, Lai J, Zhang J (2015) Jointly learning heterogeneous features for rgb-d activity recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 5344–5352
- Huang L, Zhao X, Huang K (2019) Got-10k: a large high-diversity benchmark for generic object tracking in the wild. *IEEE Trans Pattern Anal Mach Intell*
- Humpire-Mamani GE, Setio Arnaud AA, van Ginneken B, Jacobs C (2018) Efficient organ localization using multi-label convolutional neural networks in thorax-abdomen ct scans. *Phys Med Biol* 63(8):085003
- Ibanez L, Schroeder W, Ng L, Cates J (2005) The itk software guide: updated for itk version 2:4
- Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*
- Jack CR Jr, Bernstein MA, Fox NC, Thompson P, Alexander G, Harvey D, Borowski B, Britson PJ, Whitwell JL, Ward C et al (2008) The alzheimer's disease neuroimaging initiative (adni): Mri methods. *J Magn Reson Imag* 27(4):685–691
- Jaderberg M, Simonyan K, Zisserman A et al (2015) Spatial transformer networks. *Advances in neural information processing systems* 2017–2025
- Jaderberg M, Vedaldi A, Zisserman A (2014) Deep features for text spotting. In *European conference on computer vision*, pp 512–528. Springer
- Jain A, Powers A, Johnson HJ (2020) Robust automatic multiple landmark detection. In *2020 IEEE 17th international symposium on biomedical imaging (ISBI)*, pp 1178–1182. IEEE
- Jain SD, Grauman K (2014) Supervoxel-consistent foreground propagation in video. In *European conference on computer vision*, pp 656–671. Springer

- Jain SD, Xiong B, Grauman K (2017) 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), pp 2117–2126. IEEE
- Jampani V, Gadde R, Gehler PV (2017) Video propagation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 451–461
- Jan P, Stefan S (2008) Reinforcement learning of motor skills with policy gradients. *Neural Netw* 21(4):682–697
- Jang WD, Kim C-S (2017) Online video object segmentation via convolutional trident network. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5849–5858
- Jens Kober J, Bagnell A, Peters J (2013) Reinforcement learning in robotics: a survey. *Int J Robot Res* 32(11):1238–1274
- Jia Z, Yang L, Szepesvari C, Wang M (2020) 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, pp 666–686. The Cloud
- Jialue F, Wei X, Ying W, Yihong G (2010) Human tracking using convolutional neural networks. *IEEE Trans Neural Netw* 21(10):1610–1623
- Jiang M, Deng C, Pan Z, Wang L, Sun X (2018) Multiobject tracking in videos based on lstm and deep reinforcement learning. *Complexity*
- Jie Z, Liang X, Feng J, Jin X, Lu W, Yan S (2016) Tree-structured reinforcement learning for sequential object localization. In Advances in neural information processing systems, pp 127–135
- Jinwon A, Sungzoon C (2015) Variational autoencoder based anomaly detection using reconstruction probability. *Spec Lect IE* 2(1):1–18
- 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
- Jun Koh Y, Kim C-S (2017) Primary object segmentation in videos based on region augmentation and reduction. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3442–3450
- Justin G, Reza EM (2015) Concurrent markov decision processes for robot team learning. *Eng Appl Artif Intell* 39:223–234
- Jégou S, Drozdal M, Vazquez D, Romero A, Bengio Y (2017) The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 11–19
- Kalchbrenner N, Blunsom P (2013) Recurrent continuous translation models. Association for Computational Linguistics
- Kempka M, Wydmuch M, Runc G, Toczek J, Jaśkowski W (2016) Vizdoom: A doom-based ai research platform for visual reinforcement learning. In 2016 IEEE conference on computational intelligence and games (CIG), pp 1–8. IEEE
- Keni B, Rainer S (2008) Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP J Image Video Process* 2008:1–10
- Kim KK, Cho SH, Kim HJ, Lee JY (2005) Detecting and tracking moving object using an active camera. In The 7th international conference on advanced communication technology, 2005, ICACT 2005, vol 2, pp 817–820. IEEE
- Kirwan D (2010) Nhs fetal anomaly screening programme. National Standards and Guidance for England 18
- Klein S, Staring M, Murphy K, Viergever MA, Pluim JPW (2009) Elastix: a toolbox for intensity-based medical image registration. *IEEE Trans Med Imaging* 29(1):196–205
- Konda VR, Tsitsiklis JN (2000) Actor-critic algorithms. In Advances in neural information processing systems, pp 1008–1014
- Krebs J, Mansi T, Delingette H, Zhang L, Ghesu FC, Miao S, Maier AK, Ayache N, Rui L, Ali K (2017) Robust non-rigid registration through agent-based action learning. In International conference on medical image computing and computer-assisted intervention, pp 344–352. Springer
- Kristan M, Leonardis A, Matas J, Felsberg M, Pflugfelder R, Cehovin Zajc L, Vojir T, Bhat G, Lukezic A, Eldesokey A, et al (2018) The sixth visual object tracking vot2018 challenge results. In Proceedings of the European conference on computer vision (ECCV)
- Kristan M, Matas J, Leonardis A, Felsberg M, Cehovin L, Fernandez G, Vojir T, Hager G, Nebehay G, Pflugfelder R (2015) The visual object tracking vot2015 challenge results. In Proceedings of the IEEE international conference on computer vision workshops, pp 1–23
- Krizhevsky A, Sutskever I, Hinton GE (2017) Imagenet classification with deep convolutional neural networks. *Commun ACM* 60(6):84–90

- Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp 1097–1105
- Krähenbühl P, Koltun V (2011) Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pp 109–117
- Kupcsik A, Deisenroth MP, Peters J, Loh AP, Vadakkepat P, Neumann G (2017) Model-based contextual policy search for data-efficient generalization of robot skills. *Artif Intell* 247:415–439
- Kupcsik A, Deisenroth M, Peters J, Neumann G (2013) Data-efficient generalization of robot skills with contextual policy search. In *AAAI*
- Kurutach T, Clavera I, Duan Y, Tamar A, Abbeel P (2018) Model-ensemble trust-region policy optimization
- Le N, Le T, Yamazaki K, Bui TD, Luu K, Savides M (2020) Offset curves loss for imbalanced problem in medical segmentation. *arXiv preprint* [arXiv:2012.02463](https://arxiv.org/abs/2012.02463)
- Le N, Yamazaki K, Truong D, Quach KG, Savvides M (2020) A multi-task contextual atrous residual network for brain tumor detection & segmentation. *arXiv preprint* [arXiv:2012.02073](https://arxiv.org/abs/2012.02073)
- LeCun Y (1998) The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist>
- LeCun Y, Bottou L, Orr GB, Müller K-R (1998) Efficient backprop. In *Neural networks: Tricks of the trade*, pp 9–50. Springer
- LeCun Y, Touresky D, Hinton G, Sejnowski T (1988) A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school*, pp 21–28. CMU, Pittsburgh, Pa: Morgan Kaufmann
- Lea C, Flynn MD, Vidal R, Reiter A, Hager GD (2017) Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE conference on computer vision and pattern recognition*, pp 156–165
- Lea C, Reiter A, Vidal R, Hager GD (2016) Segmental spatiotemporal cnns for fine-grained action segmentation. In *European conference on computer vision*, pp 36–52. Springer
- Lea C, Vidal R, Hager GD (2016) Learning convolutional action primitives for fine-grained action recognition. In *2016 IEEE international conference on robotics and automation (ICRA)*, pp 1642–1649. IEEE
- Lea C, Vidal R, Reiter A, Hager GD (2016) Temporal convolutional networks: a unified approach to action segmentation. In *European conference on computer vision*, pp 47–54. Springer
- Leal-Taixé L, Canton-Ferrer C, Schindler K (2016) Learning by tracking: Siamese cnn for robust target association. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp 33–40
- Leal-Taixé L, Fenzi M, Kuznetsova A, Rosenhahn B, Savarese S (2014) Learning an image-based motion context for multiple people tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 3542–3549
- Leal-Taixé L, Milan A, Reid I, Roth S, Schindler K (2015) Mottchallenge 2015: towards a benchmark for multi-target tracking. *arXiv preprint* [arXiv:1504.01942](https://arxiv.org/abs/1504.01942)
- Lee JW, Park J, Jangmin O, Lee J, Hong E (2007) A multiagent approach to q-learning for daily stock trading. *Trans Syst Man Cyber Part A* 37(6):864–877
- Lee H, Kim HE, Nam H (2019) Srm: a style-based recalibration module for convolutional neural networks. pp 1854–1862
- Leibo JZ, Zambaldi VF, Lanctot M, Marecki J, Graepel T (2017) Multi-agent reinforcement learning in sequential social dilemmas. [arxiv:CoRR:abs/1702.03037](https://arxiv.org/abs/1702.03037)
- Leo B (1996) Bagging predictors. *Mach Learn* 24(2):123–140
- Leo G (2006) Random walks for image segmentation. *IEEE Trans Pattern Anal Mach Intell* 28(11):1768–1783
- Levine S, Koltun V (2014) Learning complex neural network policies with trajectory optimization. In *Proceedings of the 31st international conference on machine learning*, pp 829–837
- Li B, Yan J, Wu W, Zhu Z, Xiaolin H (2018) High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 8971–8980
- Li B, Ouyang W, Sheng L, Zeng X, Wang X (2019) 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*, pp 1019–1028. Computer vision foundation/IEEE
- Li C, Zhong Q, Xie D, Pu S (2018) Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. *arXiv preprint* [arXiv:1804.06055](https://arxiv.org/abs/1804.06055)
- Li D, Chen Q (2020) Deep reinforced attention learning for quality-aware visual recognition. In *European conference on computer vision*, pp 493–509
- Li G, Yu Y (2015) Visual saliency based on multiscale deep features. *arXiv preprint* [arXiv:1503.08663](https://arxiv.org/abs/1503.08663)
- Li J, Luong MT, Jurafsky D (2015) A hierarchical neural autoencoder for paragraphs and documents. [arxiv:CoRR:abs/1506.01057](https://arxiv.org/abs/1506.01057)

- Li K, Rath M, Burdick JW (2018) Inverse reinforcement learning via function approximation for clinical motion analysis. In 2018 IEEE international conference on robotics and automation (ICRA), pp 610–617
- Li L, Chu W, Langford J, Schapire RE (2010) A contextual-bandit approach to personalized news article recommendation. In Proceedings of the 19th international conference on World wide web, pp 661–670
- Li Y, Meriardo B (2010) Multi-video summarization based on video-mm. In 11th International workshop on image analysis for multimedia interactive services WIAMIS 10, pp 1–4. IEEE
- Li Y, Alansary A, Cerrolaza JJ, Khanal B, Sinclair M, Matthew J, Gupta C, Knight C, Kainz B, Rueckert D (2018) Fast multiple landmark localisation using a patch-based iterative network. In International conference on medical image computing and computer-assisted intervention, pp 563–571. Springer
- Liang-Chieh C, George P, Iasonas K, Kevin M, Alan L Y (2017) Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFS. *IEEE Trans Pattern Anal Mach Intell* 40(4):834–848
- Liao R, Miao S, de Tournemire P, Grbic S, Kamen A, Mansi T, Comaniciu D (2017) An artificial agent for robust image registration. In Thirty-First AAAI conference on artificial intelligence
- Liao X, Li W, Xu Q, Wang X, Jin B, Zhang X, Wang Y, Zhang Y (2020) 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, pp 9394–9402
- Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D (2015) Continuous control with deep reinforcement learning. arXiv e-prints [arXiv:1509.02971](https://arxiv.org/abs/1509.02971)
- Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D (2015) Continuous control with deep reinforcement learning. arXiv preprint [arXiv:1509.02971](https://arxiv.org/abs/1509.02971)
- Lin T-Y, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision, pp 2980–2988
- Lindeberg T (2013) Scale-space theory in computer vision, volume 256. Springer Science & Business Media
- Litjens G, Toth R, van de Ven W, Hoeks C, Kerkstra S, van Ginneken B, Vincent G, Guillard G, Birbeck N, Zhang J et al (2014) Evaluation of prostate segmentation algorithms for MRI: the promise12 challenge. *Med Image Anal* 18(2):359–373
- Liu D, Jiang T (2018) Deep reinforcement learning for surgical gesture segmentation and classification. In International conference on medical image computing and computer-assisted intervention, pp 247–255. Springer
- Liu H, Socher R, Xiong C (2019) Taming maml: efficient unbiased meta-reinforcement learning. In International conference on machine learning, pp 4061–4071
- Liu L, Hao L, Zou H, Xiong H, Cao Z, Shen C (2020) Sequential crowd counting by reinforcement learning. *Weighing counts*
- Liu L, Wu C, Lu J, Xie L, Zhou J, Tian Q (2020) Reinforced axial refinement network for monocular 3d object detection. In European conference on computer vision ECCV, pp 540–556
- Liu L, Wang H, Li G, Ouyang W, Lin L (2018) Crowd counting using deep recurrent spatial-aware network. arXiv preprint [arXiv:1807.00601](https://arxiv.org/abs/1807.00601)
- Liu T, Meng Q, Vlontzos A, Tan J, Rueckert D, Kainz B (2020) Ultrasound video summarization using deep reinforcement learning. arXiv preprint [arXiv:2005.09531](https://arxiv.org/abs/2005.09531)
- Liu W, Salzmann M, Fua P (2019) Context-aware crowd counting. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5099–5108
- Liu Z, Luo P, Wang X, Tang X (2015) Deep learning face attributes in the wild. In Proceedings of the IEEE international conference on computer vision, pp 3730–3738
- Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3431–3440
- Lorenzi M, Ayache N, Frisoni GB, Pennec X (2013) Alzheimer’s Disease Neuroimaging Initiative (ADNI), Lcc-demons: a robust and accurate symmetric diffeomorphic registration algorithm. *Neuro-image* 81:470–483
- Lotfi T, Tang L, Andrews S, Hamarneh G (2013) Improving probabilistic image registration via reinforcement learning and uncertainty evaluation. In International workshop on machine learning in medical imaging, pp 187–194. Springer
- Lowe DG (2004) Distinctive image features from scale-invariant keypoints. *Int J Comput Vis* 60(2):91–110
- Luo W, Sun P, Zhong F, Liu W, Zhang T, Wang Y (2017) End-to-end active object tracking via reinforcement learning. arXiv preprint [arXiv:1705.10561](https://arxiv.org/abs/1705.10561)
- Luong T, Sutskever I, Le QV, Vinyals O, Zaremba W (2014) Addressing the rare word problem in neural machine translation. [arxiv:CoRR:abs/1410.8206](https://arxiv.org/abs/1410.8206)

- Luu K, Zhu C, Bhagavatula C, Ngan Le TH, Savvides M (2016) A deep learning approach to joint face detection and segmentation. In *Advances in face detection and facial image analysis*, pp 1–12. Springer
- Ma C, Huang JB, Yang X, Yang M-H (2015) Hierarchical convolutional features for visual tracking. In *Proceedings of the IEEE international conference on computer vision*, pp 3074–3082
- Ma K, Wang J, Singh V, Tamersoy B, Chang YJ, Wimmer A, Chen T (2017) Multimodal image registration with deep context reinforcement learning. In *International conference on medical image computing and computer-assisted intervention*, pp 240–248. Springer
- Mahasseni B, Lam M, Todorovic S (2017) Unsupervised video summarization with adversarial lstm networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 202–211
- Maicas G, Carneiro G, Bradley AP, Nascimento JC, Reid I (2017) Deep reinforcement learning for active breast lesion detection from dce-mri. In *International conference on medical image computing and computer-assisted intervention*, pp 665–673. Springer
- Mao J, Xu W, Yang Y, Wang J, Yuille AL (2014) Deep captioning with multimodal recurrent neural networks (m-rnn). [arXiv:CoRR:abs/1412.6632](https://arxiv.org/abs/1412.6632)
- Martinez-Marin T, Duckett T (2005) Fast reinforcement learning for vision-guided mobile robots. In *Proceedings of the 2005 IEEE international conference on robotics and automation*, pp 4170–4175
- de Marvao A, Dawes-Timothy JW, Shi W, Minas C, Keenan NG, Diamond T, Durighel G, Montana G, Rueckert D, Cook SA et al (2014) Population-based studies of myocardial hypertrophy: high resolution cardiovascular magnetic resonance atlases improve statistical power. *J Cardiovasc Magn Reson* 16(1):16
- Massimiliano P, Angelo C (2017) Head pose estimation in the wild using convolutional neural networks and adaptive gradient methods. *Pattern Recogn* 71:132–143
- Matas J, James S, Davison v (2018) Sim-to-real reinforcement learning for deformable object manipulation. *arXiv preprint* [arXiv:1806.07851](https://arxiv.org/abs/1806.07851)
- Mathe S, Pirinen v, Sminchisescu C (2016) Reinforcement learning for visual object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 2894–2902
- Matsopoulos GK, Mouravliansky NA, Delibasis KK, Nikita KS (1999) Automatic retinal image registration scheme using global optimization techniques. *IEEE Trans Inf Technol Biomed* 3(1):47–60
- Matteo H, Joseph M, Hado Van H, Tom S, Georg O, Will D, Dan H, Bilal P, Mohammad A, David S (2017) Rainbow: combining improvements in deep reinforcement learning. *arXiv preprint* [arXiv:1710.02298](https://arxiv.org/abs/1710.02298)
- Menze BH, Jakab A, Bauer S, Kalpathy-Cramer J, Farahani K, Kirby J, Burren Y, Porz N, Slotboom J, Wiest R et al (2014) The multimodal brain tumor image segmentation benchmark (brats). *IEEE Trans Med Imag* 34(10):1993–2024
- Miao S, Wang ZJ, Liao R (2016) A cnn regression approach for real-time 2d/3d registration. *IEEE Trans Med Imag* 35(5):1352–1363
- Miao S, Liao R, Pfister M, Zhang L, Ordy V (2013) 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*, pp 380–387. Springer
- Michael FJ, West Jay B (2001) The distribution of target registration error in rigid-body point-based registration. *IEEE Trans Med Imaging* 20(9):917–927
- Mikolov T, Kombrink S, Burget L, Cernocký J, Khudanpur S (2011) Extensions of recurrent neural network language model. In *ICASSP*, pp 5528–5531
- Milan A, Leal-Taixé L, Reid I, Roth S, Schindler K (2016) Mot16: a benchmark for multi-object tracking. *arXiv preprint* [arXiv:1603.00831](https://arxiv.org/abs/1603.00831)
- Milan A, Leal-Taixé L, Schindler K, Reid I (2015) Joint tracking and segmentation of multiple targets. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 5397–5406
- Milan A, Rezatofighi SH, Dick A, Reid I, Schindler K (2017) Online multi-target tracking using recurrent neural networks. In *Thirty-First AAAI conference on artificial intelligence*
- Minæe S, Abdolrashidi A, Su H, Bennamoun M, Zhang D (2019) Biometric recognition using deep learning: a survey. [arxiv:CoRR:abs/1912.00271](https://arxiv.org/abs/1912.00271)
- Ming-Ming C, Mitra Niloy J, Xiaolei H, Torr Philip HS, Shi-Min H (2014) Global contrast based salient region detection. *IEEE Trans Pattern Anal Mach Intell* 37(3):569–582
- Mingxin J, Tao H, Zhigeng P, Haiyan W, Yinjie J, Chao D (2019) Multi-agent deep reinforcement learning for multi-object tracker. *IEEE Access* 7:32400–32407

- Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G et al (2015) Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533
- Mnih V, Badia AP, Mirza M, Graves A, Lillicrap T, Harley T, Silver D, Kavukcuoglu K (2016) Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp 928–1937
- Mnih V, Badia AP, Mirza M, Graves A, Lillicrap T, Harley T, Silver D, Kavukcuoglu K (2016) Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd international conference on machine learning*, pp 1928–1937
- Mordatch I, Mishra N, Eppner C, Abbeel P (2016) Combining model-based policy search with online model learning for control of physical humanoids. In *2016 IEEE international conference on robotics and automation (ICRA)*, pp 242–248
- Morimoto J, Zeglin G, Atkeson CG (2003) 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), vol 2, pp 1927–1932
- Morimoto J, Atkeson CG (2009) Nonparametric representation of an approximated poincaré map for learning biped locomotion. In *Autonomous robots*, pp 131–144
- Mousavian A, Anguelov D, Flynn J, Košecká J (2017) 3D bounding box estimation using deep learning and geometry. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp 5632–5640
- Mueller M, Smith N, Ghanem B (2016) A benchmark and simulator for UAV tracking. In *European conference on computer vision*, pp 445–461. Springer
- Märki N, Perazzi F, Wang O, Sorkine-Hornung A (2016) Bilateral space video segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 743–751
- Nagabandi A, Clavera I, Liu S, Fearing RS, Abbeel P, Levine S, Finn C (2018) Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. *arXiv preprint arXiv:1803.11347*
- Nair A, McGrew B, Andrychowicz M, Zaremba W, Abbeel P (2018) Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp 6292–6299
- Nam H, Han B (2016) Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 4293–4302
- Narges A, Lingling T, Shahin S, Yixin G, Colin L, Bejar HB, Luca Z, Sanjeev K, René V, Hager Gregory D (2017) A dataset and benchmarks for segmentation and recognition of gestures in robotic surgery. *IEEE Trans Biomed Eng* 64(9):2025–2041
- Nassif AB, Shahin I, Attili I, Azzeh M, Shaalan K (2019) Speech recognition using deep neural networks: a systematic review. *IEEE Access* 7:19143–19165
- Navarro F, Sekuboyina A, Waldmannstetter D, Peeken JC, Combs SE, Menze BH (2020) Deep reinforcement learning for organ localization in ct. *arXiv preprint arXiv:2005.04974*
- Neil B, Nicholas HA, Darcie Thomas E (2008) Minimal-bracketing sets for high-dynamic-range image capture. *IEEE Trans Image Process* 17(10):1864–1875
- Netzer Y, Wang T, Coates A, Bissacco A, Wu B, Ng AY (2011) Reading digits in natural images with unsupervised feature learning
- Ng AY, Russell SJ (2000) Algorithms for inverse reinforcement learning. In *Proceedings of the seventeenth international conference on machine learning, ICML '00*, pp 663–670. San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc
- Ng AY, Russell SJ, et al (2000) Algorithms for inverse reinforcement learning. In *ICML*, vol 1
- Nguyen TT, Li Z, Silander T, Leong T-Y (2013) Online feature selection for model-based reinforcement learning. In *Proceedings of the 30th international conference on international conference on machine learning—vol 28*, pp I-498–I-506
- Nhan Duong C, Quach KG, Luu K, Le N, Savvides M (2017) Temporal non-volume preserving approach to facial age-progression and age-invariant face recognition. In *Proceedings of the IEEE international conference on computer vision*, pp 3735–3743
- Nicolas S, Kenji D (2003) Meta-learning in reinforcement learning. *Neural Netw* 16(1):5–9
- Niedzwiedz C, Elhanany I, Liu Z, Livingston S (2008) A consolidated actor-critic model with function approximation for high-dimensional pomdps. In *AAAI 2008 workshop for advancement in POMDP solvers*
- Ning Y, He S, Zhiyong W, Xing C, Zhang L-J (2019) A review of deep learning based speech synthesis. *Appl Sci* 9(19)
- Noam B, Thomas S (2019) Superhuman ai for multiplayer poker. *Science* 365(6456):885–890

- Okuma K, Taleghani A, De Freitas N, Little JJ, Lowe DG (2004) A boosted particle filter: multitarget detection and tracking. In European conference on computer vision, pp 28–39. Springer
- Olga R, Jia D, Hao S, Jonathan K, Sanjeev S, Sean M, Zhiheng H, Andrej K, Aditya K, Michael B et al (2015) Imagenet large scale visual recognition challenge. *Int J Comput Vision* 115(3):211–252
- Orlando JI, Fu H, Breda JB, van Keer K, Bathula DR, Diaz-Pinto A, Fang R, Heng P-A, Kim J, Lee JH, et al (2020) Refuge challenge: a unified framework for evaluating automated methods for glaucoma assessment from fundus photographs. *Med Image Anal* 59:101570
- Osa T, Pajarinen J, Neumann G, Bagnell JA, Abbeel P, Peters J (2018)
- Papazoglou A, Ferrari V (2013) Fast object segmentation in unconstrained video. In Proceedings of the IEEE international conference on computer vision, pp 1777–1784
- Paschalidis IC, Li K, Moazzez Estanjini R (2009) An actor-critic method using least squares temporal difference learning. In Proceedings of the 48th IEEE conference on decision and control (CDC) held jointly with 2009 28th Chinese Control Conference, pp 2564–2569
- Peixia L, Dong W, Lijun W, Huchuan L (2018) Deep visual tracking: Review and experimental comparison. *Pattern Recogn* 76:323–338
- Peng H, Ruan Z, Long F, Simpson JH, Myers EW (2010) V3d enables real-time 3d visualization and quantitative analysis of large-scale biological image data sets. *Nat Biotechnol* 28(4):348–353
- Pengpeng L, Erik B, Haibin L (2015) Encoding color information for visual tracking: Algorithms and benchmark. *IEEE Trans Image Process* 24(12):5630–5644
- Pengyu Z, Dong W, Lu H (2020) Review and experimental comparison, Multi-modal visual tracking
- Perazzi F, Khoreva A, Benenson R, Schiele B, Sorkine-Hornung A (2017) Learning video object segmentation from static images. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2663–2672
- Perazzi F, Pont-Tuset J, McWilliams B, Van Gool L, Gross M, Sorkine-Hornung A (2016) A benchmark dataset and evaluation methodology for video object segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 724–732
- Philippe T, Michael U (2000) Optimization of mutual information for multiresolution image registration. *IEEE Trans Image Process* 9(12):2083–2099
- Pieter A, Adam C, Andrew YN (2010) Autonomous helicopter aerobatics through apprenticeship learning. *Int J Robot Res* 29(13):1608–1639
- Pirinen A, Sminchisescu C (2018) Deep reinforcement learning of region proposal networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6945–6954
- Pirsiavash H, Ramanan D, Fowlkes CC (2011) Globally-optimal greedy algorithms for tracking a variable number of objects. In CVPR 2011, pp 1201–1208. IEEE
- Plaat A, Kusters W, Preuss M (2020) Deep model-based reinforcement learning for high-dimensional problems, a survey
- Potapov D, Douze M, Harchaoui Z, Schmid C (2014) Category-specific video summarization. In European conference on computer vision, pp 540–555. Springer
- Pourreza-Shahri R, Kehtarnavaz N (2015) Exposure bracketing via automatic exposure selection. In 2015 IEEE international conference on image processing (ICIP), pp 320–323. IEEE
- Prest A, Leistner C, Civera J, Schmid C, Ferrari V (2012) Learning object class detectors from weakly annotated video. In 2012 IEEE Conference on computer vision and pattern recognition, pp 3282–3289. IEEE
- Qi Y, Zhang S, Qin L, Yao H, Huang Q, Lim J, Yang M-H (2016) Hedged deep tracking. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4303–4311
- Rakelly K, Zhou A, Finn C, Levine S, Quillen D (2019) Efficient off-policy meta-reinforcement learning via probabilistic context variables. In International conference on machine learning, pp 5331–5340
- Rameswar Panda, Abir Das, Ziyang 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
- Redmon J, Farhadi A (2018) Yolo3: An incremental improvement. *arXiv preprint [arXiv:1804.02767](https://arxiv.org/abs/1804.02767)*
- Ren L, Lu J, Wang Z, Tian Q, Zhou J (2018) Collaborative deep reinforcement learning for multi-object tracking. In Proceedings of the European conference on computer vision (ECCV), pp 586–602
- Ren L, Yuan X, Lu J, Yang M, Zhou J (2018) Deep reinforcement learning with iterative shift for visual tracking. In Proceedings of the European conference on computer vision (ECCV), pp 684–700
- Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems, pp 91–99
- Reza M, Kosecka J, et al (2016) Reinforcement learning for semantic segmentation in indoor scenes. *arXiv preprint [arXiv:1606.01178](https://arxiv.org/abs/1606.01178)*

- Richard A, Gall J (2016) Temporal action detection using a statistical language model. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3131–3140
- Rochan M, Ye L, Wang Y (2018) Video summarization using fully convolutional sequence networks. In Proceedings of the European conference on computer vision (ECCV), pp 347–363
- Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In International conference on medical image computing and computer-assisted intervention, pp 234–241. Springer
- Ros G, Koltun V, Codevilla F, Lopez A (2019) The carla autonomous driving challenge
- Rotman D (2013) Mit technology review. Retrieved from meet the man with a cheap and easy plan to stop global warming. <http://www.technologyreview.com/featuredstory/511016/a-cheap-and-easy-plan-to-stop-globalwarming>
- Rouet J-M, Jacq J-J, Roux C (2000) Genetic algorithms for a robust 3-d mr-ct registration. *IEEE Trans Inf Technol Biomed* 4(2):126–136
- Rumelhart DE (1998) The architecture of mind: a connectionist approach. *Mind Read* pp 207–238
- Runarsson TP, Lucas SM (2012) Imitating play from game trajectories: Temporal difference learning versus preference learning. In 2012 IEEE conference on computational intelligence and games (CIG), pp 79–82
- Sadeghian A, Alahi A, Savarese S (2017) Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In Proceedings of the IEEE international conference on computer vision, pp 300–311
- Sahba F (2016) Deep reinforcement learning for object segmentation in video sequences. In 2016 International conference on computational science and computational intelligence (CSCI), pp 857–860. IEEE
- Sahba F, Tizhoosh HR, Salama MMA (2006) A reinforcement learning framework for medical image segmentation. In The 2006 IEEE international joint conference on neural network proceedings, pp 511–517. IEEE
- Sahba F, Tizhoosh HR, Salama MMA (2007) Application of opposition-based reinforcement learning in image segmentation. In 2007 IEEE symposium on computational intelligence in image and signal processing, pp 246–251. IEEE
- Schulman J, Levine S, Abbeel P, Jordan M, Moritz P (2015) Trust region policy optimization. In International conference on machine learning, pp 1889–1897
- Schulman J, Levine S, Moritz P, Jordan MI, Abbeel P (2015) Trust region policy optimization. *arXiv e-prints*
- Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O (2017) Proximal policy optimization algorithms. *arXiv e-prints*
- Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O (2017) Proximal policy optimization algorithms. *arXiv preprint [arXiv:1707.06347](https://arxiv.org/abs/1707.06347)*
- Sefati S, Cowan NJ, Vidal R (2015) Learning shared, discriminative dictionaries for surgical gesture segmentation and classification. In MICCAI workshop: M2CAI, vol 4
- Sepp H, Jürgen S (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780
- Seung-Hwan B, Kuk-Jin Y (2017) Confidence-based data association and discriminative deep appearance learning for robust online multi-object tracking. *IEEE Trans Pattern Anal Mach Intell* 40(3):595–610
- Shafiee MJ, Chywl B, Li F, Wong A (2017) Fast yolo: a fast you only look once system for real-time embedded object detection in video. *arXiv preprint [arXiv:1709.05943](https://arxiv.org/abs/1709.05943)*
- Shaker MR, Yue S, Duckett T (2009) Vision-based reinforcement learning using approximate policy iteration. In 2009 international conference on advanced robotics, pp 1–6
- Shalabh B, Sutton Richard S, Mohammad G, Mark L (2009) Natural actor-critic algorithms. *Automatica* 45(11):2471–2482
- Shalev-Shwartz S, Shammah S, Shashua A (2016) Safe, multi-agent, reinforcement learning for autonomous driving. *arxiv:CoRR:abs/1610.03295*
- Shen J, Zafeiriou S, Chrysos GG, Kossaiifi J, Tzimiropoulos G, Pantic M (2015) The first facial landmark tracking in-the-wild challenge: Benchmark and results. In Proceedings of the IEEE international conference on computer vision workshops, pp 50–58
- Shi Y, Cui L, Qi Z, Meng F, Chen Z (2016) Automatic road crack detection using random structured forests. *IEEE Trans Intell Transp Syst* 17(12):3434–3445
- Silberman N, Hoiem D, Kohli P, Fergus R (2012) Indoor segmentation and support inference from rgbd images. In European conference on computer vision, pp 746–760. Springer
- Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. *arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556)*

- Sindagi VA, Patel VM (2019) Multi-level bottom-top and top-bottom feature fusion for crowd counting. In Proceedings of the IEEE international conference on computer vision, pp 1002–1012
- Singh SP, Wang L, Gupta S, Goli H, Padmanabhan P, Gulyás B (2020) 3d deep learning on medical images: a review
- Song G, Myeong H, Lee KM (2018) Seednet: automatic seed generation with deep reinforcement learning for robust interactive segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1760–1768
- Song Y, Vallmitjana J, Stent A, Jaimes A (2015) Tvsum: summarizing web videos using titles. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5179–5187
- Song Y, Ma C, Gong L, Zhang J, Lau RWH, Yang M-H (2017) Crest: Convolutional residual learning for visual tracking. In Proceedings of the IEEE international conference on computer vision, pp 2555–2564
- Stadie BC, Abbeel P, Sutskever I (2017) Third-person imitation learning. [arxiv:CoRR:abs/1703.01703](https://arxiv.org/abs/1703.01703)
- Subramanian J, Mahajan A (2019) Reinforcement learning in stationary mean-field games, pp 251–259. International foundation for autonomous agents and multiagent systems
- Sun S, Hu J, Yao M, Hu J, Yang X, Song Q, Wu X (2018) Robust multimodal image registration using deep recurrent reinforcement learning. In Asian conference on computer vision, pp 511–526. Springer
- Sundararajan K, Woodard DL (2018) Deep learning for biometrics: a survey, 51:3
- Sutton RS, Barto AG (2018) Reinforcement learning: an introduction. MIT Press
- Sutton RS, McAllester D, Singh S, Mansour Y (1999) Policy gradient methods for reinforcement learning with function approximation. In Proceedings of the 12th international conference on neural information processing systems, NIPS'99, pp 1057–1063
- Sutton RS, McAllester DA, Singh SP, Mansour Y (2000) Policy gradient methods for reinforcement learning with function approximation. In Advances in neural information processing systems vol 12, pp 1057–1063
- Szegedy C, Ioffe S, Vanhoucke V, Ilemi AA (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. In Thirty-first AAAI conference on artificial intelligent
- Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich V (2015) Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
- Szegedy C, Toshev A, Erhan D (2013) Deep neural networks for object detection. In Advances in neural information processing systems, pp 2553–2561
- Sæmundsson S, Hofmann K, Deisenroth KP (2018) Meta reinforcement learning with latent variable gaussian processes. [arXiv preprint arXiv:1803.07551](https://arxiv.org/abs/1803.07551)
- Tang Y, Tian Y, Lu J, Li P, Zhou J (2018) Deep progressive reinforcement learning for skeleton-based action recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5323–5332
- Tao R, Gavves E, Smeulders AWM (2016) Siamese instance search for tracking. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1420–1429
- Tian Z, Si X, Zheng Y, Chen Z, Li X (2020) Multi-step medical image segmentation based on reinforcement learning. J Ambient Intell Human Comput
- Tianyang X, Zhen-Hua F, Xiao-Jun W, Josef K (2019) Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual object tracking. IEEE Trans Image Process 28(11):5596–5609
- Todd H, Michael Q, Peter S (2011) A real-time model-based reinforcement learning architecture for robot control. [arxiv:CoRR:abs/1105.1749](https://arxiv.org/abs/1105.1749)
- Toro OJ, Müller H, Krenn M, Gruenberg K, Taha AA, Winterstein M, Eggel I, Foncubierta-Rodríguez A, Goksel O, Jakab A et al (2016) Cloud-based evaluation of anatomical structure segmentation and landmark detection algorithms: Visceral anatomy benchmarks. IEEE Trans Med Imaging 35(11):2459–2475
- Toromanoff M, Wirbel E, Moutarde F (2020) 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, pp 7153–716
- Toshev A, Szegedy C (2014) Deeppose: Human pose estimation via deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1653–1660
- Tsai Y-H, Yang M-H, Black MJ (2016) Video segmentation via object flow. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3899–3908
- Tsurumine Y, Cui Y, Yamazaki K, Matsubara K (2019) Generative adversarial imitation learning with deep p-network for robotic cloth manipulation. In 2019 IEEE-RAS 19th international conference on humanoid robots (humanoids), pp 274–280

- Uijlings JRR, Van De Sande KEA, Gevers T, Smeulders AWM (2013) Selective search for object recognition. *Int J Comput Vis* 104(2):154–171
- Uzkent B, Yeh C, Ermon S (2020) Efficient object detection in large images using deep reinforcement learning. In *The IEEE winter conference on applications of computer vision*, pp 1824–1833
- Valmadre J, Bertinetto L, Henriques J, Vedaldi A, Torr PHS (2017) End-to-end representation learning for correlation filter based tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 2805–2813
- Van Hasselt H, Guez A, Silver D (2016) Deep reinforcement learning with double q-learning. In *Thirtieth AAAI conference on artificial intelligence*
- Van Hove L (2001) Optimal denominations for coins and bank notes: in defense of the principle of least effort. *J Money Credit Bank* pp 1015–1021
- Vecchio G, Palazzo S, Giordano D, Rundo F, Spampinato C (2020) Mask-rl: Multiagent video object segmentation framework through reinforcement learning. *IEEE Trans Neural Netw Learn Syst*
- Vijayanarasimhan S, Ricco S, Schmid C, Sukthankar R, Fragkiadaki K (2017) Sfm-net: learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*
- Vinyals O, Babuschkin I, Chung J, Mathieu M, Jaderberg M, Czarnecki W, Dudzik A, Huang A, Georgiev P, Powell R, Ewalds T, Horgan D, Kroiss M, Danihelka I, Agapiou J, Oh J, Dalibard V, Choi D, Sifre L, Sulsky Y, Vezhnevets S, Molloy J, Cai T, Budden D, Paine T, Gulcehre C, Wang Z, Pfaff T, Pohlen T, Yogatama D, Cohen J, McKinney K, Smith O, Schaul T, Lillicrap T, Apps C, Kavukcuoglu K, Hassabis D, Silver D (2019) AlphaStar: mastering the real-time strategy game starCraft II. <https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/>
- Vlontzos A, Alansary A, Kamnitsas K, Rueckert D, Kainz B (2019) Multiple landmark detection using multi-agent reinforcement learning. In *International conference on medical image computing and computer-assisted intervention*, pp 262–270
- Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. *IEEE Trans Image Process* 13(4):600–612
- Wang G, Zuluaga MA, Li W, Pratt R, Patel PA, Aertsen M, Doel T, David AL, Deprest J, Ourselin S, et al (2018) Deepigeos: a deep interactive geodesic framework for medical image segmentation. *IEEE Trans Pattern Anal Mach Intell* 41(7):1559–1572
- Wang H, Wang Y, Zhou Z, Ji X, Gong D, Zhou J, Li Z, Liu W (2018) Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 5265–5274
- Wang JX, Kurth-Nelson Z, Tirumala D, Soyer H, Leibo JZ, Munos R, Blundell C, Kumaran D, Botvinick M (2016) Learning to reinforcement learn. *arxiv:CoRR:abs/1611.05763*, 2016
- Wang L, Lu H, Ruan X, Yang M-H (2015) Deep networks for saliency detection via local estimation and global search. In *2015 IEEE conference on computer vision and pattern recognition (CVPR)*, pp 3183–3192. IEEE
- Wang M, Deng W (2020) Mitigating bias in face recognition using skewness-aware reinforcement learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 9322–9331
- Wang M, Deng W, Hu J, Tao X, Huang Y (2019) Racial faces in the wild: Reducing racial bias by information maximization adaptation network. In *Proceedings of the IEEE international conference on computer vision*, pp 692–702
- Wang N, Yeung D-Y (2013) Learning a deep compact image representation for visual tracking. In *Advances in neural information processing systems*, pp 809–817
- Wang T, Bao X, Clavera I, Hoang J, Wen Y, Langlois E, Zhang S, Zhang G, Abbeel P, Ba J (2019) Benchmarking model-based reinforcement learning. *arxiv:CoRR:abs/1907.02057*
- Wang Y, Dong M, Shen J, Wu Y, Cheng S, Pantic M (2020) Dynamic face video segmentation via reinforcement learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 6959–6969
- Wang Z, Zhang J, Lin M, Wang J, Luo P, Ren J (2020) Learning a reinforced agent for flexible exposure bracketing selection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 1820–1828
- Wang Z, Schaul T, Hessel M, Van Hasselt H, Lanctot M, De Freitas N (2015) Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581*
- Weiming H, Xi L, Wenhan L, Xiaoqin Z, Stephen M, Zhongfei Z (2012) Single and multiple object tracking using log-euclidean riemannian subspace and block-division appearance model. *IEEE Trans Pattern Anal Mach Intell* 34(12):2420–2440
- Wickelgren WA (1973) The long and the short of memory. *Psychol Bull* 80(6):425


- Williams RJ (1992) Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach Learn* 8(3–4):229–256
- Wirth C, Fürtkranz J (2015) On learning from game annotations. *IEEE Trans Comput Intell AI Games* 7(3):304–316
- Wohlhart P, Lepetit V (2015) Learning descriptors for object recognition and 3d pose estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 3109–3118
- Woo S, Park J, Lee J-Y, Kweon IS (2018) Cbam: Convolutional block attention module. In *European conference on computer vision*, pp 3–19
- Wu Y, Lim J, Yang M-H (2013) Online object tracking: a benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 2411–2418
- Xia L, Chen C-C, Aggarwal JK (2012) 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
- Xiahai Z, Juan S (2016) Multi-scale patch and multi-modality atlases for whole heart segmentation of mri. *Med Image Anal* 31:77–87
- Xiang S, Li H (2017) On the effects of batch and weight normalization in generative adversarial networks. *arXiv preprint arXiv:1704.03971*
- Xiang Y, Alahi A, Savarese S (2015) Learning to track: online multi-object tracking by decision making. In *Proceedings of the IEEE international conference on computer vision*, pp 4705–4713
- Xiao F, Lee YJ (2016) Track and segment: an iterative unsupervised approach for video object proposals. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 933–942
- Xie Q, Luong M-T, Hovy E, Le QV (2020) Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp 10687–10698
- Xiong H, Lu H, Liu C, Liu L, Cao Z, Shen C (2019) From open set to closed set: Counting objects by spatial divide-and-conquer. In *Proceedings of the IEEE international conference on computer vision*, pp 8362–8371
- Xu H, Su F (2015) 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*, pp 387–394. ACM
- Xu N, Price B, Cohen S, Yang J, Huang TS (2016) Deep interactive object selection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 373–381
- Xu Y-S, Fu T-J, Yang H-K, Lee C-Y (2018) Dynamic video segmentation network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 6556–6565
- Xuanang X, Fugen Z, Bo L, Dongshan F, Xiangzhi B (2019) Efficient multiple organ localization in ct image using 3d region proposal network. *IEEE Trans Med Imaging* 38(8):1885–1898
- Yamakazi K, Viet-Khoa Vo-Ho AS, Le NTH, Tran T (2021) Agent-environment network for temporal action proposal generation. In *International conference on acoustics, speech and signal processing*
- Yamazaki K, Rathour VS, Le T (2021) Invertible residual network with regularization for effective medical image segmentation. *arXiv preprint arXiv:2103.09042*
- Yan W, Lei Z, Lituan W, Zizhou W (2018) Multitask learning for object localization with deep reinforcement learning. *IEEE Trans Cogn Deve Syst* 11(4):573–580
- Yan Z, Yuan Y, Zuo W, Tan X, Wang Y, Wen S, Ding E (2019) Perspective-guided convolution networks for crowd counting. In *Proceedings of the IEEE international conference on computer vision*, pp 952–961
- Yang Z, Huang L, Chen Y, Wei Z, Ahn S, Zelinsky G, Samaras D, Hoai M (2020) Predicting goal-directed human attention using inverse reinforcement learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*
- Yi W, Jongwoo L, Ming-Hsuan Y (2015) Object tracking benchmark. *IEEE Trans Pattern Anal Mach Intell* 37(9):1834–1848
- Yong KD, Moongu J (2014) Data fusion of radar and image measurements for multi-object tracking via kalman filtering. *Inf Sci* 278:641–652
- Yoon JH, Lee CR, Yang MH, Yoon KJ (2016) Online multi-object tracking via structural constraint event aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 1392–1400
- Yoshihisa T, Yunduan C, Eiji U, Takamitsu M (2019) Deep reinforcement learning with smooth policy update: application to robotic cloth manipulation. *Robot Auton Syst* 112:72–83
- Yoshua B, Patrice S, Paolo F (1994) Learning long-term dependencies with gradient descent is difficult. *IEEE Trans Neural Netw* 5(2):157–166

- You C, Lu J, Filev D, Tsiotras P (2019) Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning. *Robot Auton Syst* 114:1–18
- Yu C, Liu J, Nemati S (2019) Reinforcement learning in healthcare: a survey. *arXiv preprint arXiv:1908.08796*
- Yu T, Quillen D, He Z, Julian R, Hausman K, Finn C, Levine S (2020) Meta-world: a benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp 1094–1100
- Yun S, Choi J, Yoo Y, Yun K, Choi JY (2017) Action-decision networks for visual tracking with deep reinforcement learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 2711–2720
- Yunliang C, Said O, Manas S, Mark L, Shuo L (2015) Multi-modality vertebra recognition in arbitrary views using 3D deformable hierarchical model. *IEEE Trans Med Imaging* 34(8):1676–1693
- Yushi C, Xing Z, Xiuping J (2015) Spectral-spatial classification of hyperspectral data based on deep belief network. *IEEE J Select Top Appl Earth Observ Remote Sens* 8(6):2381–2392
- Zdenek K, Krystian M, Jiri M (2011) Tracking-learning-detection. *IEEE Trans Pattern Anal Mach Intell* 34(7):1409–1422
- Zengyi Q, Jinglu W, Yan L (2019) Monogrnet: a geometric reasoning network for monocular 3d object localization. *Proc AAAI Confer Artif Intell* 33(01):8851–8858
- Zha D, Lai K-H, Zhou K, Hu X (2019) Experience replay optimization. *arXiv preprint arXiv:1906.08387*
- Zhang J, Li W, Ogunbona PO, Wang P, Tang C (2016) Rgb-d-based action recognition datasets: a survey. *Pattern Recogn* 60:86–105
- Zhang-Wei H, Chen Yu-M, Shih-Yang S, Tzu-Yun S, Yi-Hsiang C, Hsuan-Kung Y, Brian Hsi-Lin H, Chih-Chieh T, Yueh-Chuan C, Tsu-Ching H, et al. Virtual-to-real: learning to control in visual semantic segmentation. *arXiv preprint arXiv:1802.00285*
- Zhang D, Maei H, Wang X, Wang Y-F (2017) Deep reinforcement learning for visual object tracking in videos. *arXiv preprint arXiv:1701.08936*
- Zhang D, Yang L, Meng D, Xu D, Han J (2017) Spsfn: 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*, pp 4429–4437
- Zhang K, Chao W-L, Sha F, Grauman K (2016) Video summarization with long short-term memory. In *European conference on computer vision*, pages 766–782. Springer
- Zhang Y, Zhou D, Chen S, Gao S, Ma Y (2016) Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 589–597
- Zhao H, Qi X, Shen X, Shi J, Jia J (2018) Icnnet for real-time semantic segmentation on high-resolution images. In *Proceedings of the European conference on computer vision (ECCV)*, pp 405–420
- Zhao H, Shi J, Qi X, Wang X, Jia J (2017) Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 2881–2890
- Zheng Y, Liu D, Georgescu B, Nguyen H, Comaniciu D (2015) 3d deep learning for efficient and robust landmark detection in volumetric data. In *International conference on medical image computing and computer-assisted intervention*, pp 565–572. Springer
- Zhewei H, Wen H, Shuchang Z (2019) Learning to paint with model-based deep reinforcement learning. In *Proceedings of the IEEE international conference on computer vision*, pp 8709–8718
- Zhiheng H, Wei X, Kai Y (2015) Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*
- Zhiwu H, Chengde W, Thomas P, Van Gool L (2017) Deep learning on lie groups for skeleton-based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 6099–6108
- Zhong-Qiu Z, Shou-Tao X, Dian L, Wei-Dong T, Zhi-Da J (2019) A review of image set classification. *Neurocomputing* 335:251–260
- Zhou B, Zhao H, Puig X, Fidler S, Barriuso A, Torralba A (2017) Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 633–641
- Zhou K, Qiao Y, Xiang T (2018) Deep reinforcement learning for unsupervised video summarization with diversity-representativeness reward. In *Thirty-Second AAAI conference on artificial intelligence*
- Zhou K, Xiang T, Cavallaro A (2018) Video summarisation by classification with deep reinforcement learning. *arXiv preprint arXiv:1807.03089*
- Zhu X, Xiong Y, Dai J, Yuan L, Wei Y (2017) Deep feature flow for video recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 2349–2358
- Zou WY, Wang X, Sun X, Lin Y (2014) Generic object detection with dense neural patterns and regionlets. *arXiv preprint arXiv:1404.4316*

van Beek P (2018) Improved image selection for stack-based hdr imaging. arXiv preprint [arXiv:1806.07420](https://arxiv.org/abs/1806.07420)
van Hasselt H, Guez A, Silver D (2015) Deep reinforcement learning with double q-learning. arXiv e-prints, [arXiv:1509.06461](https://arxiv.org/abs/1509.06461)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Ngan Le^{1,2}  · Vidhiwar Singh Rathour^{1,2} · Kashu Yamazaki^{1,2} · Khoa Luu^{1,2} · Marios Savvides^{1,2}

Vidhiwar Singh Rathour
vsrathou@uark.edu

Kashu Yamazaki
kyamazak@uark.edu

Khoa Luu
khoaluu@uark.edu

Marios Savvides
msavvid@ri.cmu.edu

¹ Department of Computer Science and Computer Engineering, University of Arkansas, Fayetteville, AR 72703, USA

² Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA