

YOLOv8 DQN Tracker：智能多目標追蹤解決方案

摘要

本文探討基於感測元件的控制應用，提出使用視覺資訊解決大量控制參數的問題，例如車輛自動駕駛或桌邊服務機器人。我們將我們的演算法應用在控制一維伺服馬達的動作，而伺服馬達會根據裝設的光學設備，取得目標物影像用以追蹤目標物。

1. 前言

隨著終端機及邊緣單元處理速度大幅提高，物聯網技術迎來新的里程碑，例如工廠物流車可透過控制中心資料庫和物聯網技術取得座標與平面圖資訊進行移動決策，自動駕駛車則透過視覺和光達資訊進行自主駕駛。然而，取得上帝視角的環境資料仍難以實現，需要大量感測元件，增加硬體成本且邊緣運算處理速度無法即時處理大量資料，目前常見做法是大量裝設感測元件進行特徵萃取，但實際用來進行行為決策的環境資料遠小於取得的環境資料。此外，現有的控制演算法難以控制大量的參數。

2. 研究目的

由於終端機以及邊緣單元的處理速度大幅成長，基於感測元件的控制應用迎來新的里程，例如工廠內的物流車輛可以根據其對應的控制中心資料庫，藉由物聯網取得其他物流車在廠內的座標以及取得廠內的平面圖，最後藉由上帝視角的資訊進行物流車的移動決策。或是自動駕駛車輛根據視覺影像以及光達資訊進行自動駕駛的輔助決策。而物流車和自動駕駛車輛這兩個例子最大的差別是：前者是透過控制中心的資料庫，也就是上帝視角的資料進行行為決策；後者則是透過自行從環境中得到的資料進行行為決策。

在一般的狀況中，我們是無法取得上帝視角的環境資料，例如自動駕駛車輛、餐廳的桌邊服務機器人、無人飛行器或是股票買賣的決策等，而我只能夠透過大量裝設感測元件來盡可能得到趨近上帝視角的環境資訊，但這樣做不僅會大量增加硬體成本，更重要的是邊緣運算的處理速度、資料的傳輸速度和中央處理器的處理速度無法「即時」處理如此龐大的資料。目前常見的做法是大量裝設感測元件再透過邊緣裝置進行特徵萃取，這樣做不僅可以盡可能取得趨近上帝視角的資料，亦可以減少資料傳輸的時間，然而這樣做的弊端不僅前面所提到的大量增加硬體成本，更重要的是實際用來進行行為決策的環境資料遠遠小於大量感測元件從環境中取得的環境資料。

除了資料取得的問題外，現在的控制演算法大多無法控制大量的參數。不同於學習式演算法，傳統的演算法是試圖建立「所有的」行為規則，這些演算法大多只能根據當下的環境資訊進行決策；或是諸如 PWM 控制、模糊控制等控制演算法，這些演算法可以根據過去的狀態以及結果進行決策以及執行後的結果預測，後者的這些演算法雖然可靠，但無法控制大量的參數，因此過去的控制應用大多都停留在工廠機具或是簡單任務的控制。

為了解決「環境資訊」和「大量控制參數」的問題，我們將著重在僅使用光學感測元件，也就是視覺資訊，來完成控制大量控制參數的任務，例如車輛自動駕駛或是桌邊服務機器人。由於訓練成本以及硬體成本的考量，我們將我們的演算法應用在控制一維伺服馬達的動作，而伺服馬達會根據裝設的光學設備，取得目標物影像用以追蹤目標物。

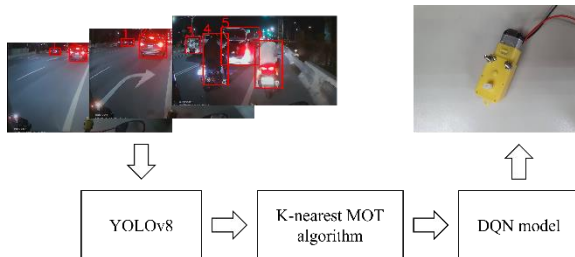


圖 1. 智能多目標追蹤解決方案架構圖

3. 文獻探討

大多數現有的 MOT 方法可以分為兩類：1) 離線和 2) 在線。第一類方法將所有檢測結果分為短軌跡段和一般軌跡，並將這些軌跡連接成完整軌跡。例如，Zamir 等人，通過使用廣義最小團圖，將所有檢測結果關聯起來，這些檢測結果以全域方式結合了外觀和運動信息。Maksai 等人，提出了一種跟踪具有非馬爾可夫行為約束的多個對象的方法。第二類中的方法使用當前幀和過去幀的檢測結果來估計目標的軌跡。例如 Yang 等人，通過解決具有非線性運動模式和 robust appearance constraints 的多目標跟踪能量最小化問題，引入了在線學習 CRF 模型。Choi 等人，提出了一個聚合的局部流描述符來準確測量不同檢測結果之間的親和力。我們的方法是使用當前幀的 YOLOv8 辨識結果與前一幀的 MOT 表進行 K-nearest 的標籤預測，在重疊或噪聲的部分則使用標籤所在 bounding box 面積和中心點距離差進行處理。

目前多參數的控制主要通過求解逆運動學方程移動或定位代理人相對於固定參照系的資料求解。這些演算法由被儲存在一個可編程的內存中，

用於進行多參數控制的行為決策。代理人可以在開環或外部資訊的反饋下進行控制。開環控制沒有外部傳感器或環境感知能力，嚴重依賴高度結構化的環境，這些環境需要經過非常靈敏的校準。如果一個部件輕微移動，控制系統可能必須停止並重新校準。在這種策略下，代理人通過跟隨記憶中的一系列位置，並在不同時間移動到它們編程的指定位置。在一些更先進的機器人系統中，外感反饋通過使用監控傳感器、力傳感器、甚至視覺或深度傳感器，持續監控代理人的狀態，例如位置和速度的狀態。然後將反饋與信息進行比較，存儲以更新決策，從而實現所需的代理人行為。這兩種傳統控制場景都嚴重依賴於基於硬件的解決方案。

隨著深度學習等現代人工智能技術的進步，以及機器人和機械學的最新發展，研究界和工業界一直在尋求更多基於軟件的控制解決方案，這些解決方案使用低成本傳感器，對操作要求較低環境和校準。關鍵是做出最少但有效的控制決策，並專注於穩健的算法和軟件。代替硬編碼方向來協調所有關節，控制策略可以通過學習獲得，然後相應地更新。強化學習(DRL)是最有前途的算法之一，因為它不需要預定義的訓練數據集，非常適合多參數控制的操作。強化學習方法可以使用來自機械臂實驗的輸入或是我們所使用的伺服馬達視覺資料，透過收集任何一種動態生成的經驗數據，並通過迭代更新特定的數據來訓練深度神經網絡(DNN)控制策略網絡的策略參數。

4. YOLOv8

YOLOv8 是一個 SOTA 模型，它建立在以前 YOLO 版本的成功基礎上，並引入了新的功能和改進，以進一步提升性能和靈活性。具體創新包括一個新的骨幹網絡、一個新的 Anchor-Free 檢測頭和一個新的損失函數，可以在從 CPU 到 GPU 的各種硬件平台上運行。

YOLOv8 與過去的版本的不同之處可歸結為以下五點。1) 它提供全新的 SOTA 模型，包括 P5 640 和 P6 128 分辨率的物體檢測網絡和基於 YOLACT 的實例分割模型。和 YOLOv5 一樣，也根據縮放因子提供了 N/S/M/L/X 尺度的不同尺寸模型，以滿足不同場景的需求 2) Head 部分相比 YOLOv5 改動較大，換成了目前主流的解耦頭結構，將分類和檢測頭分離，同時也從 Anchor-Based 換成了 Anchor-Free 3) Loss 計算方面採用了正樣本分配策略，並引入了 Distribution Focal Loss 4) 骨幹網絡和 Neck 部分可能參考了 YOLOv7 ELAN 設計思想，將 YOLOv5 的 C3 結構換成了梯度流更豐富的 C2f 結構，並對不同尺度模型調整了不同的通道數，屬於對模型結構細節微調，不再是無腦的一套參數應用所有模型，大幅提升了模型性能。不過這個 C2f 模塊中存在 Split 等操作對特定硬件部署沒有之前那麼友好 5) 訓練的數據增強部分引入

了 YOLOX 中的最後 10 epoch 關閉 Mosaic 增強的操作，可以有效地提升精度

5. K-nearest MOT 演算法

在 YOLOv8 完成影像 bounding box 的偵測任務後，K-nearest MOT 演算法會將相關參數與前一幀經由 K-nearest MOT 演算法所得的 MOT table 進行計算，迭代出新的 MOT table。其中，MOT table 除了帶有目標物的標籤外，同時包含原始的 bounding box 和計算出的用以進行 DQN 推論的環境參數。

5.1 原理介紹

多目標物的追蹤可以被化簡為物件辨識的問題，我們可以將當下的 bounding box 資料與前一幀的 bounding box 資料進行映射。其中，我們定義 MOT table 所包含的資料集合為 S，寫作

$$S = \{(x_n, y_n) \mid n=1,2,\dots,k_1\} \quad (1)$$

x_n 與 y_n 分別代表目標物的特徵向量以及所對應的標籤，在下一個時刻，代理人從視覺資料中獲得新的目標物以及新的特徵，定義為 S' ，寫作

$$S' = \{(x'_m, y_n) \mid n=1,2,\dots,k_1, m=1,2,\dots,k_2\} \quad (2)$$

x'_m 與 y_n 分別代表新的目標物的特徵向量以及演算法推論出的標籤，推論的過程選用 K-nearest 分類器，它將一組特徵分配給它在特徵空間中最近鄰的類別，即

$$C_n^{km}(x'_m) = y_n \quad (3)$$

x 的特徵空間維度，會由感測器的選擇以及選取的特徵所決定，理論上，高維度的影像特徵再經由深度學習演算法進行特徵萃取，會得到較好的結果。

5.2 模型設計

在模型設計上，為了確保模型能實時運行，我們定義 MOT table 最大標籤數量為 20，其中 MOT table 的存取方式採用 mod 20 的 hash table，此方式確保記憶體的存取效率，在 x 的特徵維度極高時，該設計會有良好的表現。

5.3 行為表現

我們將 K-nearest MOT 演算法應用於新聞畫面的物件追蹤，從圖 1 以及圖 2 可以觀察到，演算法從圖 1 開始進行多物件追蹤的任務，模型辨識出身穿橘色背心的站務人員、紅色衣服的民眾和白衣服的民眾，並且依序進行標籤，在下一幀的畫面中，標籤 2 和標籤 3 的影像重疊，但 K-nearest MOT

演算法所計算出的標籤仍舊正確，正確判斷出是標籤 2 遮擋住標籤 3 的物件，原先代表標籤 3 的目標從 MOT table 中剔除，並將標籤依序分配給下一個 YOLOv8 辨識到的物件。



圖 2. 將 MOT 演算法應用於新聞畫面。影片時間 0:28



圖 3. 將 MOT 演算法應用於新聞畫面。圖 1 的下一幀畫面

6. Deep Q network

DQN 模型是我們提出的解決方案中用以控制光學設備移動參數的模型。DQN 是屬於強化學習的一種，是除了監督學習和非監督學習之外的第三種基本的機器學習方法，基本的強化學習被建模為馬爾可夫決策過程。

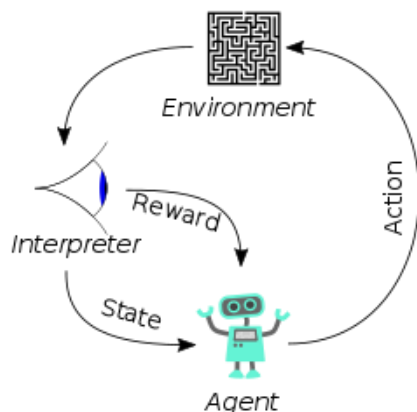


圖 4. 強化學習的訓練流程

6.1 原理介紹

我們定義環境和代理人的狀態集合 S ；代理人的行為集合 A ，則我們的決策結果就是根據當下的狀態 s_t 得到各行為選項的機率，再從中決定之，數學表示為：

$$\pi(a_t, s) = \Pr(a_t = a | s_t = s) \quad (4)$$

其中，狀態集合 S 可以是代理人前一個時刻的狀態以及行為選擇、視覺的資訊或是我們透過獎勵規則給予的「獎勵」。而 Q table 就是透過環境和代理人的狀態集合 S 以及根據獎勵規則所得到獎勵所建立的表格，在傳統的 Q learning 中，理論上我們只要 Q table 設計的夠大，能收集所有狀態集合 S 與所對應的獎勵，我們就能夠建立完美的代理人，但很遺憾的這樣的事是不切實際的，不只是因為我們無法收集所有的狀態集合來建立完美的 Q table，更重要的是環境永遠是不穩定的。

當我們的從環境中獲取的資訊較少時，我們會發現此時的狀態集合是有限集，但是我們無法從 Q table 中學習正確的行為規則；而當今天我們從環境中獲取大量資訊時，同時各種感測器的訊號為類比訊號時，我們確實能從 Q table 中學習正確的代理人行為規則，但事實上礙於各種感測器的取樣率問題，狀態集合 S 仍是有限集。

為了解決上述問題 Deep Q network 就是基於 Q learning 演算法所設計的深度學習模型，希望透過深度學習中激活函數的架構，能從有限大小的 Q table 學習正確的代理人規則。

6.2 模型設計

DQN 分別有不斷迭代的 evaluation network，以及根據 'replace_target_iter' 更新的 target network，如此的設計是希望該模型是從長期的經驗學習而非根據短期經驗進行決策，同時這樣的設計能幫助模型更好的收斂。

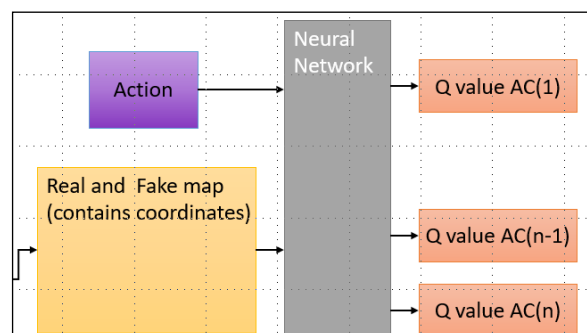


圖 5. DQN 架構圖

圖 5 中黃色的部分是 DQN 模型的核心，代表過去經驗的紀錄，其大小為 memory_size，

memory_size 越大代表越多過去的經驗越多，同時需要的儲存資源也越大，但其中可能儲存很多不重要的甚至是誤導模型的經驗。若 memory_size 設計太小則會使得模型沒有足夠的經驗應付多變的狀況，但若環境變化不大則可以將 memory_size 設小。因此適當的 memory_size 可以讓模型有足夠的經驗去決策行為，同時也能適當淘汰掉不佳的經驗。

6.3 行為表現

我們知道 Memory_size 的大小決定了模型能記憶的過去經驗數量，因此我們設計 500、200、100 三種不同 Memory_size。可以觀察到 Memory_size = 200 時有最好的追蹤效果，這是因為比起 Memory_size = 500，Memory_size = 200 能夠快速淘汰掉不佳的經驗，同時比起 Memory_size = 100，Memory_size = 200 又能夠記錄下足夠多的經驗以供 DQN 做決策。

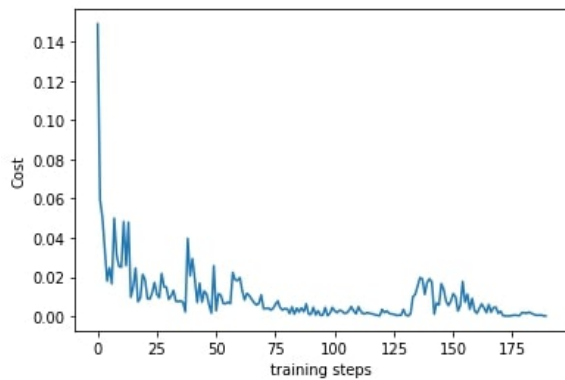


圖 6. Memory_size = 500

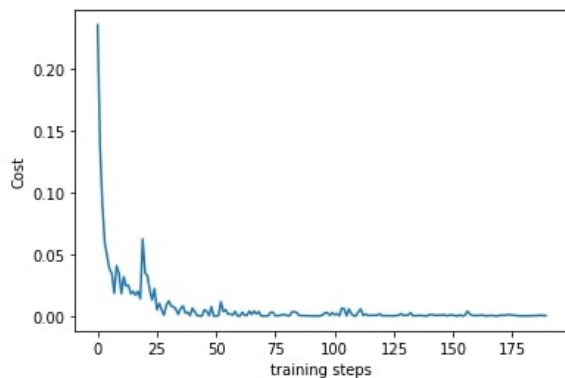


圖 7. Memory_size = 200

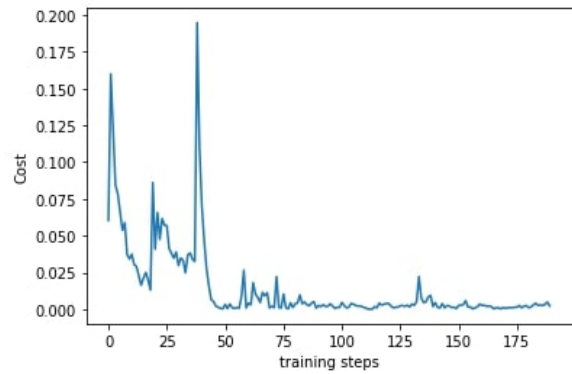


圖 8. Memory_size = 100

7. 機構與表現

機構實體如圖 8 所示，左上角為控制伺服馬達的單晶片，其接收來自終端機的訊號將其轉換成伺服馬達對應的控制訊號。為了快速部署，光學設備我們是直接採自己的手機，並透過 Google 開發的影片通信服務技術將手機鏡頭的影像傳送到終端機，由於該技術是使用 HTTP 協議，也就是說傳送到終端的鏡頭畫面會被呈現在網頁上，除了使用爬蟲獲得原始的影像資訊，我們則是使用 Pillow 套件直接擷取螢幕的影像作為 YOLOv8 的輸入。

在圖 9 中，手機鏡頭的影像被傳遞到左側螢幕的左下角，Pillow 擷取螢幕左下角的畫面作為 YOLOv8 的輸入，接著我們的智能多目標追蹤系統會將相關資料傳輸給 K-nearest MOT 演算法進行建立物件追蹤的標籤表，為了展示，我們將其對應的標籤結果以視覺方式呈現，接著 DQN 再根據標籤表以及選取的標籤輸出控制訊號，終端機將控制訊號傳輸給單晶片後由單晶片將訊號解碼後發出對應的控制訊號控制伺服馬達。

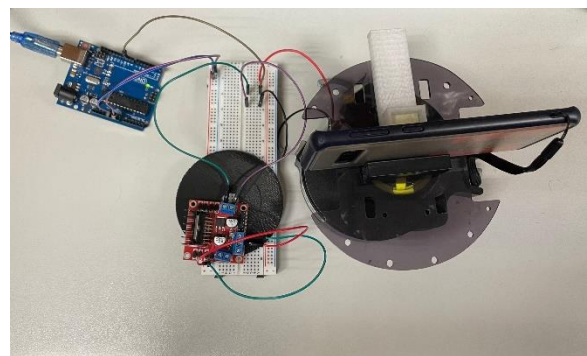


圖 9. 智能多目標追蹤解決方案應用在一維伺服馬達目標追蹤上的系統俯視圖

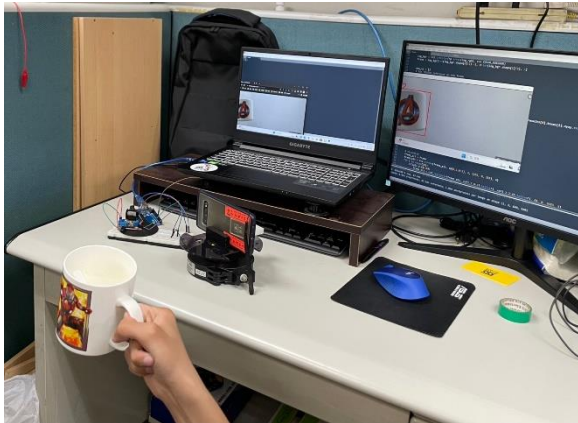


圖 10. 智能多目標追蹤系統實際工作圖

8. 結論

在實驗過程中，為了快速完成訓練以及進行部署，我們將 DQN 的狀態維度限制在 6 維，但仍舊可以完成一維伺服馬達的目標追蹤控制任務。

經由實驗我們確定了該架構在實時性以及控制任務上的可靠度，唯獨系統在目標物仍存在但 YOLOv8 沒有辨識出 bounding box 時，會出現 MOT table 丟失的問題，這是因為在我們的演算法中，如果 YOLOv8 回傳的 bounding box 為空，我們會認為影像中已經沒有目標物了，因此在未來我們會將影像辨識端整合進 MOT 演算法中，以確保辨識端的可靠性。

因為解決方案的可行性，在未來的任務中，我們希望將鏡頭安裝在無人遙控汽車上，藉由影像的目標追蹤結果，由 DQN 模型輸出控制訊號控制無人遙控車跟隨指定目標。

參考文獻

- [1] Fix, Evelyn; Hodges, Joseph L. (1951). Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties (PDF) (Report). USAF School of Aviation Medicine, Randolph Field, Texas. Archived (PDF) from the original on September 26, 2020.
- [2] Bradly C. Stadie, Sergey Levine, Pieter Abbeel: "Incentivizing Exploration In Reinforcement Learning With Deep Predictive Models", 2015; arXiv:1507.00814.
- [3] Henry Charlesworth, Giovanni Montana: "PlanGAN: Model-based Planning With Sparse Rewards and Multiple Goals", 2020; arXiv:2006.00900.
- [4] Danijar Hafner, Timothy Lillicrap, Jimmy Ba, Mohammad Norouzi: "Dream to Control: Learning Behaviors by Latent Imagination", 2019; arXiv:1912.01603.
- [5] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", NIPS, 2012.
- [6] R. L. Galvez, A. A. Bandala, E. P. Dadios, R. R. P. Vicerra and J. M. Z. Maningo, "Object Detection Using Convolutional Neural Networks", IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON, vol. 2018, pp. 2023-2027, October 2019.
- [7] W. Fang, L. Wang and P. Ren, "Tinier-YOLO: A Real-time Object Detection Method for Constrained Environments", IEEE Access, vol. 8, pp. 1935-1944, 2019.
- [8] R. Huang, J. Pedoeem and C. Chen, "YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers", IEEE Int. Canf. Big Data Big Data, pp. 2503-2510, 2019.
- [9] Qiwei Xie, Qian Long, Liming Zhang, Zhao Sun: "A Robust Real-Time Computing-based Environment Sensing System for Intelligent Vehicle", 2020; [http://arxiv.org/abs/2001.09678 arXiv:2001.09678].
- [10] Shangliang Xu, Xinxin Wang, Wenyu Lv, Qinyao Chang, Cheng Cui, Kaipeng Deng, Guanzhong Wang, Qingqing Dang, Shengyu Wei, Yuning Du, et al. PPYOLOE: An evolved version of YOLO. arXiv preprint arXiv:2203.16250, 2022.
- [11] Chien-Yao Wang, Alexey Bochkovskiy, and HongYuan Mark Liao. Scaled-YOLOv4: Scaling cross stage partial network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 13029–13038, 2021.
- [12] Mingxing Tan, Ruoming Pang, and Quoc V Le. EfficientDet: Scalable and efficient object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
- [13] Jianfeng Wang, Lin Song, Zeming Li, Hongbin Sun, Jian Sun, and Nanning Zheng. End-to-end object detection with fully convolutional network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 15849–15858, 2021.
- [14] Yunyang Xiong, Hanxiao Liu, Suyog Gupta, Berkin Akin, Gabriel Bender, Yongzhe Wang, Pieter-Jan Kindermans, Mingxing Tan, Vikas Singh, and Bo Chen. MobileDets: Searching for object detection architectures for mobile accelerators. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 3825–3834, 2021.
- [15] A. L. Strehl and M. L. Littman, An Analysis of Model-Based Interval Estimation for Markov Decision Processes. Journal of Computer and System Sciences, 74, 12091331.
- [16] J. Z. Kolter and A. Y. Ng, Near-Bayesian Exploration in Polynomial Time Proceedings of the 26th Annual International Conference on Machine Learning, pp. 18, 2009.
- [17] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, et al. Human-level Control Through Deep Reinforcement Learning. Nature, 518(7540):529533, 2015.
- [18] T. Lang, M. Toussaint, K. Keristing, Exploration in relational domains for model-based reinforcement

- learning Proc. AAMAS, 2014.
- [19] F. Doshi-Velez, D. Wingate, N. Roy, and J. Tenenbaum, Nonparametric Bayesian Policy Priors for Reinforcement Learning. NIPS, 2014.
 - [20] X. Guo, S. Singh, H. Lee, R. Lewis, X. Wang, Deep Learning for Real-Time Atari Game Play Using Offline Monte-Carlo Tree Search Planning. NIPS, 2014.
 - [21] A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy. Deep variational information bottleneck. arXiv preprint arXiv:1612.00410, 2016.
 - [22] Buesing, T. Weber, S. Racaniere, S. Eslami, D. Rezende, D. P. Reichert, F. Viola, F. Besse, K. Gregor, D. Hassabis, et al. Learning and querying fast generative models for reinforcement learning. arXiv preprint arXiv:1802.03006, 2018.
 - [23] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289, 2015.
 - [24] Henaff, W. F. Whitney, and Y. LeCun. Model-based planning with discrete and continuous actions. arXiv preprint arXiv:1705.07177, 2018.
 - [25] Kaiser, M. Babaeizadeh, P. Milos, B. Osinski, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine, et al. Model-based reinforcement learning for atari. arXiv preprint arXiv:1903.00374, 2019.
 - [26] E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.
 - [27] T. Wang and J. Ba. Exploring model-based planning with policy networks. arXiv preprint arXiv:1906.08649, 2019.
 - [28] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.
 - [29] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. arXiv preprint arXiv:1401.4082, 2014.
 - [30] A. Piergiovanni, A. Wu, and M. S. Ryoo. Learning real-world robot policies by dreaming. arXiv preprint arXiv:1805.07813, 2018.
 - [31] P. Parmas, C. E. Rasmussen, J. Peters, and K. Doya. PIPPS: Flexible model-based policy search robust to the curse of chaos. arXiv preprint arXiv:1902.01240, 2019.
 - [32] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.
 - [33] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows*. Prentice Hall, 1993.
 - [34] A. Andriyenko, S. Roth, and K. Schindler. Continuous Energy Minimization for Multi-Target Tracking. *IEEE TPAMI*, vol. 35, no. 1, 2014.
 - [35] S.-H. Bae and K.-J. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *CVPR* 2014.
 - [36] T. E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Multi-target tracking using joint probabilistic data association. In *19th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, volume 19, pages 807–812, Dec. 1980.
 - [37] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. A mobile vision system for robust multi-person tracking. In *CVPR* 2008.
 - [38] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32(9):1627–1645, 2010.
 - [39] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. J. V. Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *IEEE TPAMI*, 33(9):1820–1833, 2011.
 - [40] Aeberhard, S., Coomans, D., de Vel, O. Comparison of Classifiers in High Dimensional Settings. Tech. Rep. no. 92-02, (1992), Dept. of Computer Science and Dept. of Mathematics and Statistics, James Cook University of North Queensland, 1992.
 - [41] Cover, T.M., Hart, P.E. Nearest neighbor pattern classification. *IEEE Trans. Inform. Theory*, IT-13(1):21–27, 1967.
 - [42] Jozwik, A. A learning scheme for a fuzzy k-nn rule. *Pattern Recognition Letters*, 1:287–289, 1983.
 - [43] Keller, J.M., Gray, M.R., Givens, J.A. A fuzzy k-nn neighbor algorithm. *IEEE Trans. Syst. Man Cybern.*, SMC-15(4):580–585, 1985.
 - [44] Keller, J.M., Gray, M.R., Givens, J.A. A fuzzy k-nn neighbor algorithm. *IEEE Trans. Syst. Man Cybern.*, SMC-15(4):580–585, 1985.
 - [45] Keller, J.M., Gray, M.R., Givens, J.A. A fuzzy k-nn neighbor algorithm. *IEEE Trans. Syst. Man Cybern.*, SMC-15(4):580–585, 1985.
 - [46] Ren, Liangliang, Jiwen Lu, Zifeng Wang, Qi Tian, and Jie Zhou. 2018. “Collaborative Deep Reinforcement Learning for Multi-object Tracking.” *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, 586–602.
 - [47] Arulkumaran K, Deisenroth MP, Brundage M, Bharath AA. 2017. Deep reinforcement learning: a brief survey. *IEEE Signal Process. Mag.* 34(6):26–38
 - [48] R. Liu, F. Nageotte, P. Zanne, M. de Mathelin and B. Dresp-Langley, "Deep reinforcement learning for the control of robotic manipulation: A focussed mini-review", *Robotics*, vol. 10, no. 1, pp. 22, Jan. 2021.