Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 56-2 (2023) 9781–9786

Robust Reinforcement Learning Based Visual Servoing with Convolutional Features ⋆

Haolin Fei * **Ziwei Wang** * **Andrew Kennedy** *

* *School of Engineering, Lancaster University, Lancaster, LA1 4YW, United Kingdom, (e-mail: {h.fei1, z.wang82, a.kennedy3}@lancaster.ac.uk).*

Abstract: Image-based visual servoing poses a significant challenge for robotic systems, as it involves detecting the object and controlling the robot arm based on image feedback. These tasks are further complicated by various interferences such as changes in ambient lighting, distractions, and background clutter. Recent research suggests that reinforcement learning is a promising approach to learning efficient control policies for such tasks. In this paper, we propose a data-driven approach for closed-loop visual servoing based on a reinforcement learning algorithm that does not require any prior knowledge of the task object or intrinsic camera parameters. Our method utilizes a convolutional neural network for object detection and a servoing strategy that enables the robot to determine the relative camera motion and position the camera at the desired pose. Our experimental results demonstrate that the proposed approach successfully steers the camera using only a single template image of the task object.

Keywords: Reinforcement leaning, imitation learning, visual servoing

1. INTRODUCTION

The ability to distinguish between task objects and other objects and precisely reach them to perform tasks is a hallmark of human dexterity. While robots have made significant strides in recent years, replicating this level of performance remains a formidable challenge. Current robots often rely on accurate target positions and inverse kinematics techniques to determine the appropriate joint configuration for reaching objects Bai et al. (2021). Unfortunately, this approach typically requires prior knowledge of the desired target location, making the system vulnerable to failures caused by even minor variations. Therefore, it is crucial to develop a more robust and adaptive approach that can handle the complexity of real-world scenarios.

To address the limitations of fixed target positions, robots can rely on sensors to provide perception information about their environment. By leveraging this information, robots can deduce the target location or estimate real-time errors, creating a more robust closed-loop system. However, the reliability of such systems is often compromised by environmental complexity. For example, variations in ambient light or distractions within the robot's field of view can interfere with detection results, making it challenging to design perception algorithms that operate effectively in diverse environments.

In the context of industry and early-stage research, a structured and controlled environment is often necessary to compensate for interferencePages et al. (2006). This may involve pre-defining the workspace, designing specific detection algorithms based on the 3D features of task objects or the environment, and setting up a simple background De Luca et al. (2008). However, such an approach sacrifices flexibility and adaptability, and may not be suitable for daily tasks or small-batch production with frequent changeovers. Moreover, handcrafting a model for object recognition and localization is time-consuming and impractical for every object in daily life. To achieve general-purpose manipulation, robots can benefit from using vision-based methods and learning to automate with minimal human supervision. This approach can increase the flexibility and adaptability of the system and enable the robot to perform a wide range of tasks, including those that were not explicitly programmed beforehand. Visual servoing has two main branches, namely image-based visual servoing (IBVS) and position-based visual servoing (PBVS). While PBVS often requires three-dimensional reconstruction Dong and Zhu (2015), which is highly sensitive to camera calibration parameters, IBVS relies less on camera calibration and is a better choice in an unstructured environment. However, controlling the robot with IBVS can be challenging since the action is applied directly to the image plane. To address this issue, data-driven methods have gained increasing attention in various fields, including manufacturing Xia et al. (2022), robotics Xue et al. (2020); Bacha et al. (2022); Huang et al. (2021), and optics Wu et al. (2020). Among these methods, reinforcement learning (RL) has shown great potential in generating control policies without requiring a model of the

⋆ This work was supported by Lloyd's Register Foundation, a charitable foundation, helping to protect life and property by supporting engineering-related education, public engagement and the application of research www.lrfoundation.org.uk.

10.1016/j.ifacol.2023.10.295

system. By learning implicitly from a pre-specified reward function and optimizing the reward through interactions with the environment, RL can converge to a learned policy that maximizes the reward Xiao et al. (2022).

In this paper, we propose an image-based visual servoing approach with an eye-in-hand configuration and a reinforcement learning (RL) policy as the controller. Our feature extraction module provides feedback for the RL controller, reducing the exploration difficulties for the RL algorithm and enabling faster convergence to a robust policy. Our novel algorithm combines the robustness of both the backbone convolutional neural network (CNN) and the RL algorithm. The contributions of this paper can be summarized as follows:

1) Our proposed algorithm demonstrates superior performance compared to pure RL-controlled methods and other traditional hand-designed feature extraction methods.
2) Our proposed feature mapping method has less computational complexity and cost compared to semantic segmentation-based feature extraction.
3) We demonstrate that moving the image processing to a separate unit to provide input to the RL algorithm allows for faster convergence rates.

2. RELATED WORKS

One of the main challenges in vision-based robot servoing tasks is to accurately distinguish the task object from other distractions in the environment. To address this, various feature detection algorithms have been proposed, including SIFT Lowe (2004), SURF Ta et al. (2009), FAST Viswanathan (2009), ORB Rublee et al. (2011), and others. In recent years, convolutional neural networks (CNNs) have been widely applied for feature extraction in a variety of vision tasks, such as object detection Galvez et al. (2018); Kang et al. (2016), recognition Liang and Hu (2015); Spoerer et al. (2017), and tracking Wang and Yeung (2013); Ning et al. (2017). CNNs are preferred for their robustness in dealing with occlusion, deformation, and other changes in object appearance. Additionally, recent studies have shown that enhancing the CNN's encoding of shape information can further improve the performance of template matching Gao and Spratling (2022).

Another challenge in visual servoing is the design of a robust controller that can handle estimation errors caused by the feature extractor and control the robot arm in the image plane. End-to-end methods have been explored to estimate the pose for visual servoing in previous works Bateux et al. (2017); Florence et al. (2019); Saxena et al. (2017); James et al. (2017); Rahmatizadeh et al. (2018). However, directly deriving the control policy from visual information can lead to a tremendous state or action space, which is often challenging for exploration, especially when dealing with sparse rewards, and the algorithm may fail to converge. Many end-to-end methods train the perception and control systems jointly Levine et al. (2016); Sadeghi et al. (2018); Levine et al. (2018); Pinto and Gupta (2016). Our method processes the raw visual information and controls the robot in two separate modules without any prior knowledge of the task object and the environment, rather than using observed images as input to infer the control command Sadeghi et al. (2018) or utilizing the adapted perception information to drive the robot with traditional control law Bateux et al. (2017). In this paper, we implement template matching algorithm to develop the feedback control with the deep features extracted from CNN. Compared with Sampedro et al. (2018), which also designed an RL-based IBVC controller to track a red round target for an unmanned aerial vehicle (UAV), our method does not have any prior knowledge of the target object and there are more distractions in our scenario.

3. METHODS

In this paper, we propose an IBVS method that enables a robot to servo to a desired image without the need for further programming or prior knowledge about the exact location of the object or the image Jacobian. Our approach is based on a template matching algorithm that utilizes the deep features extracted from a pre-trained Convolutional Neural Network (CNN) to track the object of interest. The proposed method processes the raw visual information and controls the robot in two separate modules: visual feedback and control. The visual feedback module extracts the deep features from the current image and compares them with the template features to obtain the error. The control module generates the control command based on the error and sends it to the robot controller to move the robot arm in the image plane towards the target object.

The proposed IBVS structure is illustrated in Fig. 1. It is designed to enable a robot to servo to a desired image without the need for further programming or prior knowledge about the exact location of the object or the image Jacobian. The framework consists of two main components: the RL agent and the environment that the RL agent interacts with. At each time step, the current extracted visual features $f(t)$ and the desired visual features $f^*(t)$ are used to calculate the error $e(t)$, which is the difference between $f(t)$ and $f^*(t)$. This error signal serves as the input to the controller used by the RL agent.

3.1 Feature Extraction

The objective of feature extraction is to derive the error $e(t)$ in the image by extracting features such as the change of the object position compared with the last frame image. This $e(t)$ is then used for controlling the robot's motion. Accurately matching the task object in the current frame is crucial for feature extraction. There are generally four methods used for feature extraction in the computer vision field: template matching, feature matching, shape/outline detection, and data-driven matching. Traditional template matching requires translation consistency, and it is sensitive to deformation and other appearance changes. Feature matching is less sensitive to complicated deformation but requires complicated shapes to extract enough feature points, and can be time-consuming. Shape detection uses morphological transformations to detect the outline but is susceptible to noise. In this paper, we use a CNN as the feature extractor. Compared to the other methods mentioned, the CNN feature extractor is less vulnerable to noise, and does not require intrinsic camera parameters

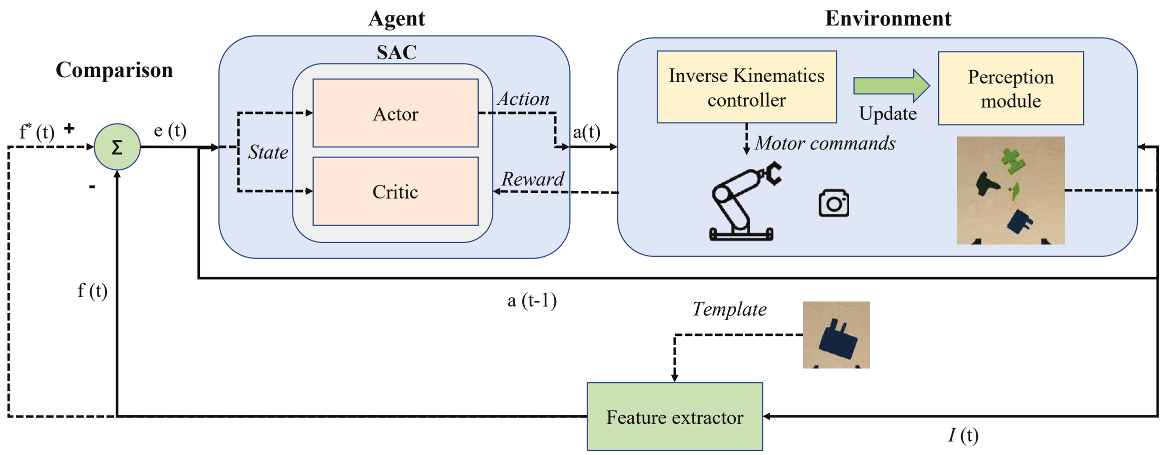

Fig. 1. The proposed RL-based IBVS scheme. The agent, which applies SAC algorithm, receives as input an error estimation and the previous action. The environment receives the end-effector velocity as an input and calculate the corresponding motor command with inverse kinematics and takes an image after the agent action being executed. The feature extractor, maps the provided template image with the current image and estimates the feature.

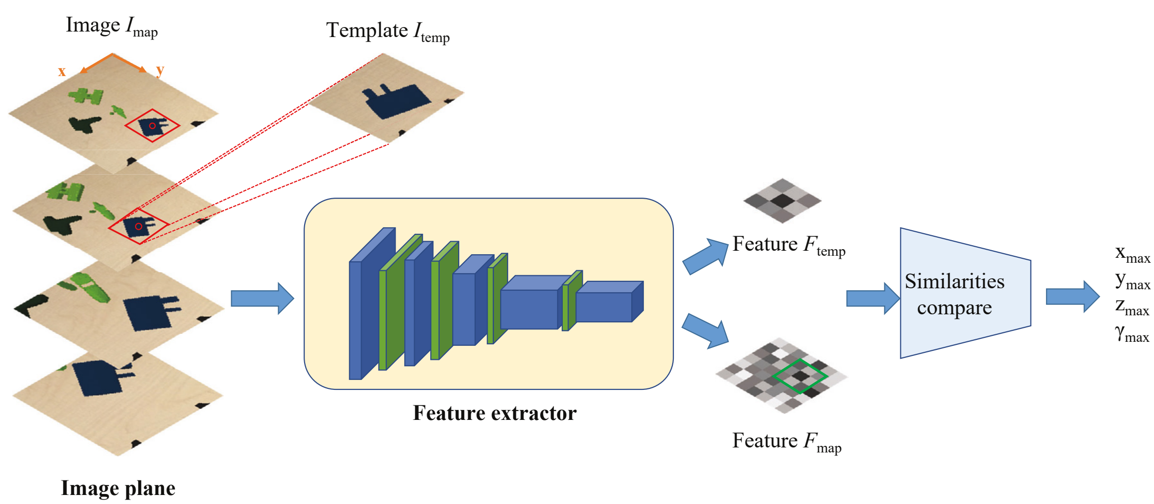

Fig. 2. Architecture of the proposed IBVS framework. The robot moves in the world frame while capturing images at different time steps. In each time step, the captured image is pre-processed and fed into a feature extractor to extract deep features. The similarity is computed by comparing the template feature and the current map, which produces the estimated position of the object in the image plane and a confidence level.

or the use of hand-eye calibration to obtain the extrinsic camera parameters. In particular, we are interested in a general model that can be applied directly to different unseen objects without human engineering, where there is no need to design hand-crafted models or fine-tune the parameters for a specific object.

To locate the target object in the current image, we use a deep CNN to extract feature vectors F_{temp} and $F_{map}(t)$ from a low-resolution template image I_{temp} and a full-resolution current image I_t, respectively. To compare the similarity between the two feature vectors, we use normalized cross-correlation (NCC) Yoo and Han (2009). Specifically, we compute the distance between F_{temp} and $F_{map}(t)$ as follows:

$$\gamma = \frac{\sum_i \left[F_{map} - \bar{F}_{map}\right]\left[F_{temp} - \bar{F}_{temp}\right]}{\sqrt{\left\{\sum_i \left[F_{map} - \bar{F}_{map}\right]^2 \sum_i [F_{temp} - \bar{F}_{temp}]^2\right\}}}, \quad (1)$$

where $\bar{F}map$ is the mean of $Fmap$ in the range under $Ftemp$ and $\bar{F}temp$ is the mean of the $Ftemp$. The coordinate of the matching point $(xmax, y_{max})$ is located at the peak γ_{max} in cross-correlation.

3.2 Reinforcement Learning based Controller

The IBVS system aims to minimize the error between the current and target image features. We address the problem of learning to move the robot to a desired location in an image plane without any prior knowledge of the object shape, pose information, or camera parameters. In our setup, only a single template image is provided, and any additional information provided by augmented reality markers or camera calibration is not available. Therefore, the feature Jacobian cannot be explicitly calculated. We formalize the problem as a Markov Decision Process (MDP) framework $< S, A, P_{s,a}, R >$ over discrete time steps in an environment E, where S, A, and R denote the sets of states, actions, and rewards. $P_{s,a}$ is the transition probability that describes the probability of the agent moving from state $s(t)$ to a new state $s(t+1)$ after taking the action $a(t)$. The final objective of the RL agent is to find a policy π that predicts an action $a(t) \in \mathcal{A}$ based on the observation of the state $s(t) \in \mathcal{S}$ at each time step t, which maximizes the expected reward $\mathcal{S} \times \mathcal{A} \to \mathbb{R}$. The cumulative expected reward can be written as:

$$\mathbb{E}_{(s(t),a(t))\sim\rho_\pi}\left[\sum_t R\left(s(t),a(t)\right)\right]. \tag{2}$$

In the previous section, we introduced the feature extraction scheme that results in a higher dimensional feature value. The environment receives (x_{max}, y_{max}) and NCC score γ_{max} from the feature extractor, which are combined with the last step action $a(t-1)$ to form a 4-dimensional vector:

$$s(t) = \{e_x(t), e_y(t), \gamma(t), a(t-1)\}, \tag{3}$$

where $e_x(t)$ and $e_y(t)$ represent the normalized error of the template center to the matched center in the image plane. The reward function is defined as:

$$I_t = \begin{cases} 0, & e(t) \geq e_{thresh} \\ -1, & e(t) < e_{thresh} \end{cases} \tag{4}$$

This function calculates the error in the image plane, given by $e(t) = \sqrt{(e_x(t))^2 + (e_y(t))^2}$, and provides feedback based on whether the threshold value is exceeded.

4. IMPLEMENTATION DETAILS AND RESULTS

In this section, we provide details on the implementation of our proposed image-based visual servoing method. To evaluate our approach in simulation, we utilized PyBullet Coumans and Bai (2016–2021), a real-time collision detection and multi-physics simulator. The simulated environment includes a 7 degree-of-freedom UR5e robot and a wrist camera with a field of view of 60° and an image size of 256×256 pixels. Before each episode of training, we captured a template image of size 32×32 pixels.We control the robot using a Cartesian space position controller, and the simulation includes a set of randomly generated simple rigid objects.

4.1 Feature Extractor Model

We use a NCC-based template matching approach, as proposed by Kim et al. (2017), as the base for our perception module. This approach takes scale adaptive deep convolutional feature vectors from both the template and the current frame image using a pre-trained VGG-16 network and measures the similarity of the output features to locate the target. We initialize the feature extraction module with the template image of the task object, and use the maximal correlation value on the map to find the target location in the image plane. It is important to note that the peak correlation value of the same image is not always located at the center of the image, so we must find this location at initialization. Once the feature extraction module has located the target, it provides features to the reinforcement learning algorithm, which generates a motor command that moves the robot in the image domain.

4.2 Training Methodology using Soft Actor-Critic RL

For training the agent, we use Soft Actor Critic (SAC) Haarnoja et al. (2018), which is an off-policy maximum entropy RL algorithm. In addition to maximizing the cumulative reward as described in Equation 2, SAC algorithm also aims to maximize the entropy of the action taken by the agent. The modified expected reward used in SAC is given as:

$$\mathbb{E}(s(t),a(t)) \sim \rho\pi\left[\sum_t R\left(s(t),a(t)\right) + \alpha H\left(\pi\left(\cdot \mid s(t)\right)\right)\right]. \tag{5}$$

Here, α is a temperature parameter that determines the trade-off between maximizing the expected reward and maximizing the entropy of the action distribution. We tune this parameter to balance exploration and exploitation during training. During training, we collect experience by running multiple episodes of the task, each consisting of a sequence of state-action pairs. The experience is then used to update the policy and value function networks using stochastic gradient descent. We train the Q-values, Actor, and Value function with the Adam optimizer using the same learning rate of 3e-4. Our RL agent is trained and tested using an episodic RL setting. At the beginning of each episode, a random object is generated on the table. The robot arm moves towards the object to fetch the template image and initialize the feature extractor. After initialization, the task scene is created, including resetting the robot motor to a fixed position and generating the target and distractors in the FoV of the wrist camera. Additional noise is added to the environment by varying the light direction, brightness, and diffusion coefficient. In each episode, the agent is rewarded according to Equation 4. Setting the error threshold e_{thresh} too large or too small can lead to task failures. We set the error threshold to 0.1. A looser definition for the threshold would lead to a jittery robot arm, as there is no penalty for the jitter, and a stricter definition would lead to longer exploration time before the algorithm converges. Early termination occurs in two scenarios: when the object is out of the FoV for ten time steps, and when the robot arm moves out of the preset boundaries. In both cases, the total reward is set to -200, which is the highest penalty term that one episode can receive.

Our experiments aim to address two primary questions: (1) Can an RL agent learn to servo the robot in the image plane from deep features? and (2) How well do deep features perform compared to traditional template

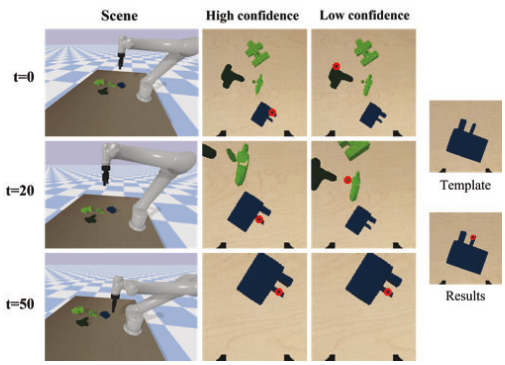

Fig. 3. Examples of visual servoing process in simulation.

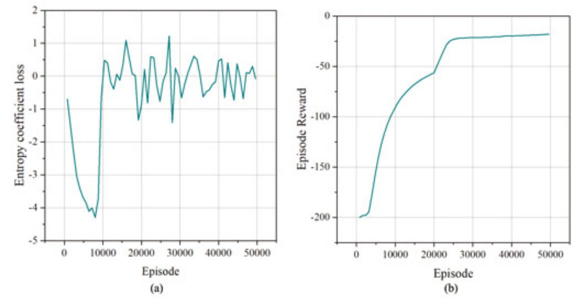

Fig. 4. Training performance curves: (a) Accumulated reward versus training episodes for the visual servoing RL agent. (b) Entropy coefficient loss versus training episodes for the SAC algorithm.

matching algorithms? To answer the first question, we created a simulated visual servoing scene for training the control policy. The results are shown in Fig. 3, which depicts the simulation scene at different time steps, along with the template matching results with high and low confidence. The right-most two images show the template image and the estimated results with the target location marked in a red circle, respectively. Despite the occasional noisy and unreliable data, the learned visual servoing RL agent is capable of identifying the target object in the presence of other distractors. Our RL-based agent can converge, and it exhibits stable behavior around 25,000 episodes, as shown in the training curve (see Fig. 4).

Table 1. Results of comparison of the image matching.

Algorithm	Time (s)	Match rate (%)
SIFT	0.12	88.1
ORB	0.03	80.3
CNN	0.13	90.7

To answer the second question, we conducted a comparison between the CNN-based template matching module and two traditional image matching techniques, namely SIFT and ORB. For this experiment, we used a fixed template image generated during the initialization of the algorithm in the simulator, and compared it with a map image taken by the robot in a random position while keeping all objects within the FoV. As the ground truth value is in the image domain, while we can only assess the world frame coordination of the object via the simulator, we designed a quick segmentation algorithm to extract the target object from the image. Specifically, we used the simulator's capability of taking segmentation images to create a mask of the template object and then applied a connected component labeling algorithm to find the target object in the segmented binary image. The results are presented in Table 1. We did not take into account the initialization time and training time of the CNN. Our findings suggest that the CNN-based template matching method has a higher match rate and is more robust to deformation and occlusion compared to SIFT and ORB.

5. CONCLUSION

In this paper, we presented a novel RL-based approach for visual servoing that can learn to approach the target automatically with only a single template image. Our approach leverages the robust deep feature extraction capabilities of CNNs to calculate the error signal in the image plane. By training the RL network in a simulated environment where the environment state can be easily accessed, we achieved more robust results compared to conventional feature-based and shape-based feature extraction methods. Our experimental results demonstrate the effectiveness of our approach in identifying the target object amidst distractors and noisy data. Future work could explore the use of our approach in real-world scenarios and investigate the potential benefits of transfer learning to enhance the efficiency of training.

ACKNOWLEDGEMENTS

The author would like to acknowledge the support of The Welding Institute (TWI) and Stefano Tedeschi.

REFERENCES

Bacha, S.C., Bai, W., Wang, Z., Xiao, B., and Yeatman, E.M. (2022). Deep reinforcement learning-based control framework for multilateral telesurgery. *IEEE Transactions on Medical Robotics and Bionics*, 4(2), 352–355.

Bai, W., Zhang, N., Huang, B., Wang, Z., Cursi, F., Tsai, Y.Y., Xiao, B., and Yeatman, E.M. (2021). Dual-arm coordinated manipulation for object twisting with human intelligence. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 902–908.

Bateux, Q., Marchand, E., Leitner, J., Chaumette, F., and Corke, P. (2017). Visual servoing from deep neural networks. *arXiv preprint arXiv:1705.08940*.

Coumans, E. and Bai, Y. (2016–2021). Pybullet, a python module for physics simulation for games, robotics and machine learning. `http://pybullet.org`.

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

Dong, G. and Zhu, Z. (2015). Position-based visual servo control of autonomous robotic manipulators. *Acta Astronautica*, 115, 291–302.

Florence, P., Manuelli, L., and Tedrake, R. (2019). Self-supervised correspondence in visuomotor policy learning. *IEEE Robotics and Automation Letters*, 5(2), 492–499.

Galvez, R.L., Bandala, A.A., Dadios, E.P., Vicerra, R.R.P., and Maningo, J.M.Z. (2018). Object detection using convolutional neural networks. In *TENCON 2018-2018 IEEE Region 10 Conference*, 2023–2027. IEEE.

Gao, B. and Spratling, M.W. (2022). Robust template matching via hierarchical convolutional features from a shape biased cnn. In *The International Conference on Image, Vision and Intelligent Systems (ICIVIS 2021)*, 333–344. Springer.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, 1861–1870. PMLR.

Huang, Z., Wang, Z., Bai, W., Huang, Y., Sun, L., Xiao, B., and Yeatman, E.M. (2021). A novel training and collaboration integrated framework for human–agent teleoperation. *Sensors*, 21(24), 8341.

James, S., Davison, A.J., and Johns, E. (2017). Transferring end-to-end visuomotor control from simulation to real world for a multi-stage task. In *Conference on Robot Learning*, 334–343. PMLR.

Kang, K., Ouyang, W., Li, H., and Wang, X. (2016). Object detection from video tubelets with convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 817–825.

Kim, J., Kim, J., Choi, S., Hasan, M.A., and Kim, C. (2017). Robust template matching using scale-adaptive deep convolutional features. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 708–711. IEEE.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1), 1334–1373.

Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., and Quillen, D. (2018). Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International journal of robotics research*, 37(4-5), 421–436.

Liang, M. and Hu, X. (2015). Recurrent convolutional neural network for object recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3367–3375.

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

Ning, G., Zhang, Z., Huang, C., Ren, X., Wang, H., Cai, C., and He, Z. (2017). Spatially supervised recurrent convolutional neural networks for visual object tracking. In *2017 IEEE international symposium on circuits and systems (ISCAS)*, 1–4. IEEE.

Pages, J., Collewet, C., Chaumette, F., and Salvi, J. (2006). An approach to visual servoing based on coded light. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, 4118–4123. IEEE.

Pinto, L. and Gupta, A. (2016). Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *2016 IEEE international conference on robotics and automation (ICRA)*, 3406–3413. IEEE.

Rahmatizadeh, R., Abolghasemi, P., Bölöni, L., and Levine, S. (2018). Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration. In *2018 IEEE international conference on robotics and automation (ICRA)*, 3758–3765. IEEE.

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

Sadeghi, F., Toshev, A., Jang, E., and Levine, S. (2018). Sim2real viewpoint invariant visual servoing by recurrent control. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4691–4699.

Sampedro, C., Rodriguez-Ramos, A., Gil, I., Mejias, L., and Campoy, P. (2018). Image-based visual servoing controller for multirotor aerial robots using deep reinforcement learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 979–986. IEEE.

Saxena, A., Pandya, H., Kumar, G., Gaud, A., and Krishna, K.M. (2017). Exploring convolutional networks for end-to-end visual servoing. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 3817–3823. IEEE.

Spoerer, C.J., McClure, P., and Kriegeskorte, N. (2017). Recurrent convolutional neural networks: a better model of biological object recognition. *Frontiers in psychology*, 8, 1551.

Ta, D.N., Chen, W.C., Gelfand, N., and Pulli, K. (2009). Surftrac: Efficient tracking and continuous object recognition using local feature descriptors. In *2009 IEEE conference on computer vision and pattern recognition*, 2937–2944. IEEE.

Viswanathan, D.G. (2009). Features from accelerated segment test (fast). In *Proceedings of the 10th workshop on image analysis for multimedia interactive services, London, UK*, 6–8.

Wang, N. and Yeung, D.Y. (2013). Learning a deep compact image representation for visual tracking. *Advances in neural information processing systems*, 26.

Wu, Z., Wu, X., and Zhu, Y. (2020). Structured illumination-based phase retrieval via generative adversarial network. In *Quantitative Phase Imaging VI*, volume 11249, 14–22. SPIE.

Xia, M., Shao, H., Huang, Z., Zhao, Z., Jiang, F., and Hu, Y. (2022). Intelligent process monitoring of laser-induced graphene production with deep transfer learning. *IEEE Transactions on Instrumentation and Measurement*, 71, 1–9.

Xiao, B., Lam, H.K., Xuan, C., Wang, Z., and M. Yeatman, E. (2022). Optimization for interval type-2 polynomial fuzzy systems: A deep reinforcement learning approach. *IEEE Transactions on Artificial Intelligence*, 1–12.

Xue, T., Wang, Z.w., Zhang, T., Bai, O., Zhang, M., and Han, B. (2020). Fixed-time constrained acceleration reconstruction scheme for robotic exoskeleton via neural networks. *Frontiers of Information Technology & Electronic Engineering*, 21(5), 705–722.

Yoo, J.C. and Han, T.H. (2009). Fast normalized cross-correlation. *Circuits, systems and signal processing*, 28(6), 819–843.