International Journal of Science and Research Archive

eISSN: 2582-8185

Cross Ref DOI: 10.30574/ijsra

Journal homepage: https://ijsra.net/

(REVIEW ARTICLE)

Check for updates

Intelligent robotic object grasping system using computer vision and deep reinforcement learning techniques

Osita Miracle Nwakeze *, Ogochukwu C Okeke and Ike Joseph Mgbemfulike

Department of Computer Science, Faculty of Physical Sciences, Chukwuemeka Odumegwu Ojukwu University, Uli. Anambra State, Nigeria.

International Journal of Science and Research Archive, 2025, 14(03), 511-521

Publication history: Received on 01 February 2025; revised on 07 March 2025; accepted on 10 March 2025

Article DOI: https://doi.org/10.30574/ijsra.2025.14.3.0693

Abstract

This study presents an intelligent robotic object grasping system using computer vision technique and deep reinforcement learning to enhance robotic manipulation. The proposed technique employs You Only Look Once (YOLOv3) for real-time object recognition and localisation, while the Soft Actor-Critic (SAC) system uses depth image information to determine the optimal gripping areas. By transforming the gripping point into a three-dimensional grasping posture, the robotic manipulator can then efficiently choose and arrange objects. The COCO dataset was utilised to increase YOLO's detection accuracy, and transfer learning sped up the training process. The performance evaluation of the proposed system revealed a mean Average Precision (mAP) of 91.2% for item detection and an 87.3% grasping success rate. 10-fold cross-validation verified the model's robustness and generalisability, demonstrating minimal variation in performance across test settings. Compared to traditional gripping approaches, the proposed strategy improved accuracy by 27% and execution efficiency by 35%. These findings demonstrate the YOLO-SAC framework's promise for practical robotic applications by providing a flexible and scalable approach to automated object handling in a range of settings.

Keywords: Intelligent Robot; Object Grasping; Computer Vision; Reinforcement Learning; Soft Actor-Critic; You Only Look Once

1. Introduction

For many years, grasping has been a major area of robotics study, allowing robots to interact effectively with real-world items. When humans see something, even for the first time, they automatically know where to pick it up. Because of this plus the human hand's extraordinary dexterity, every gripping effort is nearly always successful. However, as robots lack such instinct and dexterity, this is a very difficult task for present robot systems (Bicchi, 2000). Robots nowadays are quite sophisticated; they can be programmed to carry out extremely intricate actions with extreme precision and accuracy. These programs, however, are frequently environment- or object-specific. These programs become unstable when any of these conditions change, necessitating the creation of a new program. Robot grasping's lack of flexibility is a difficult issue (Zhang et al., 2021; Souza et al., 2021).

The desire to improve productivity, efficiency, and safety in work settings has made human– robot cooperation (HRC) a study issue of growing importance in contemporary industry (RoblaGómez et al., 2017; Villani et al., 2018; Ajoudani et al., 2018). The performance of repetitive and difficult activities might be greatly enhanced by the combination of human talents with robotic capabilities. Nonetheless, there are still issues that need to be resolved, such as the efficient coordination of activities and the smooth exchange of information between parties (Michalos et al., 2018; Papanastasiou et al., 2019; Hoffman, 2019). Giving collaborative robots cognitive capabilities has become a surprising trend in recent years, turning them from basic automated machines into perceptive and flexible team players. The

* Corresponding author: Osita Miracle Nwakeze

growing need for robots that can collaborate with people, comprehend their intentions, and actively participate in challenging jobs in dynamic contexts is what is causing this change. To enable robots to learn, anticipate, and predict human activities, collaborative cognition includes a variety of critical skills (Castro et al., 2021; Rozo et al., 2018; Jiao et al., 2020).

In collaborative settings, assistive robots are made to collaborate with people during maintenance or assembly tasks, offering prompt assistance to increase job efficiency (Hoffman and Breazeal, 2007; Williams, 2009). A robot can help a human worker by providing a part, tool, or component; holding a part while the operator works on it; or carrying out a particular subtask on its own. In any event, effective cooperation is greatly aided by an assistive robot's capacity to predict the future requirements of a human operator. Robots can proactively help or supplement human jobs by predicting human intents, behaviours, and demands. This improves overall efficiency and provides timely support (Huang and Mutlu, 2016; Duarte et al., 2018).

By providing data-driven approaches, the emergence of deep learning algorithms has offered a potential remedy for this issue (LeCun et al., 2015). By mimicking human gripping techniques, grasp posture creation may be accomplished through the interactions between sensors and the surroundings (Tian et al., 2023). By incorporating Deep Convolutional Neural Networks (CNNs) into grasping algorithms, robots are better equipped to adapt to environmental changes. Instead of using physical object models for training, modern data-driven methods for robotic grasping instruct a robot using vast volumes of data (He et al., 2016). The gripping attitude is manually annotated, and these training data often include several photos of each object in different orientations and locations (Krizhevsky et al., 2012; Bohg et al., 2013). No matter how these things are positioned, the robot can grip them when training is finished.

Pinto et al. (2016) and Levine et al. (2016) were significant previous studies that addressed the problem by collecting large amounts of data over an extended period of time (50,000 and 800,000 datapoints, respectively). Many grasps are made, and the results, whether successful or not, are recorded. This allows the algorithm to learn which grasps work best. The resultant technique was effective in grasping specific objects. Another approach that has been studied is the use of object detectors as part of the grasping algorithm. The high precision and speed of contemporary object detectors, which can do several iterations in a second, enable real-time application. This is comparable to the early phases of CNN research, when they were inappropriate for real-time detection and grasping due to their high computational cost and lengthy run periods each iteration (Adarsh and Rathi, 2020). Kim et al. (2021) used the Open Image Dataset (OID) to train an object detector with the pictures and gripping points of two item classes. These two groups of items may be correctly detected by the resulting model with a precision of about 70%. Huang et al. (2024) use an object detector in conjunction with multiagent deep reinforcement learning to tackle grasping.

It is inefficient and difficult to train the algorithms to recognise everything in a universe with an endless number of different objects. Therefore, it is not desirable to use a completely data-driven strategy for object grasping. Thus, creating an effective technique that doesn't require a lot of training data while yet achieving a high success rate in real grasping of unfamiliar objects is a crucial difficulty in the literature on robotic grasping (Khor et al., 2024). In order to address this issue, this work suggests an object grasping method that combines the Soft Actor-Critic (SAC) algorithm (Haarnoja et al., 2018; Haarnoja et al., 2019) with the YOLO algorithm (Bochkovskiy et al., 2020; Redmon and Farhadi, 2018; Redmon and Farhadi, 2017; Redmond et al., 2016). It is commonly recognised that YOLO can quickly locate, identify, and detect items in a picture. Specifically, YOLO is able to locate the item of interest inside a camera's range of view and utilise that position data as input to a reinforcement learning algorithm. Training time can be significantly decreased because searching through a complete image is not necessary.

2. Research methodology

In this work, a deep reinforcement learning system with self-learning capabilities is combined with computer vision-based object identification, recognition, and localisation to create a robotic object gripping method. The robotic pick-and-place system designed in this research is schematically depicted in Figure 1. YOLO will identify the item of interest in the cameracaptured image, as seen in Figure 1. Using the depth image information of the object bounding box, SAC will offer the desired gripping point in the image plane. To operate the robot manipulator to grasp items of interest and arrange them in a desired location, the gripping point on the 2D-image plane is transformed into a desired 3D grasping posture in Cartesian space.

Depending on the incentive mechanism, the system will provide the prize details.

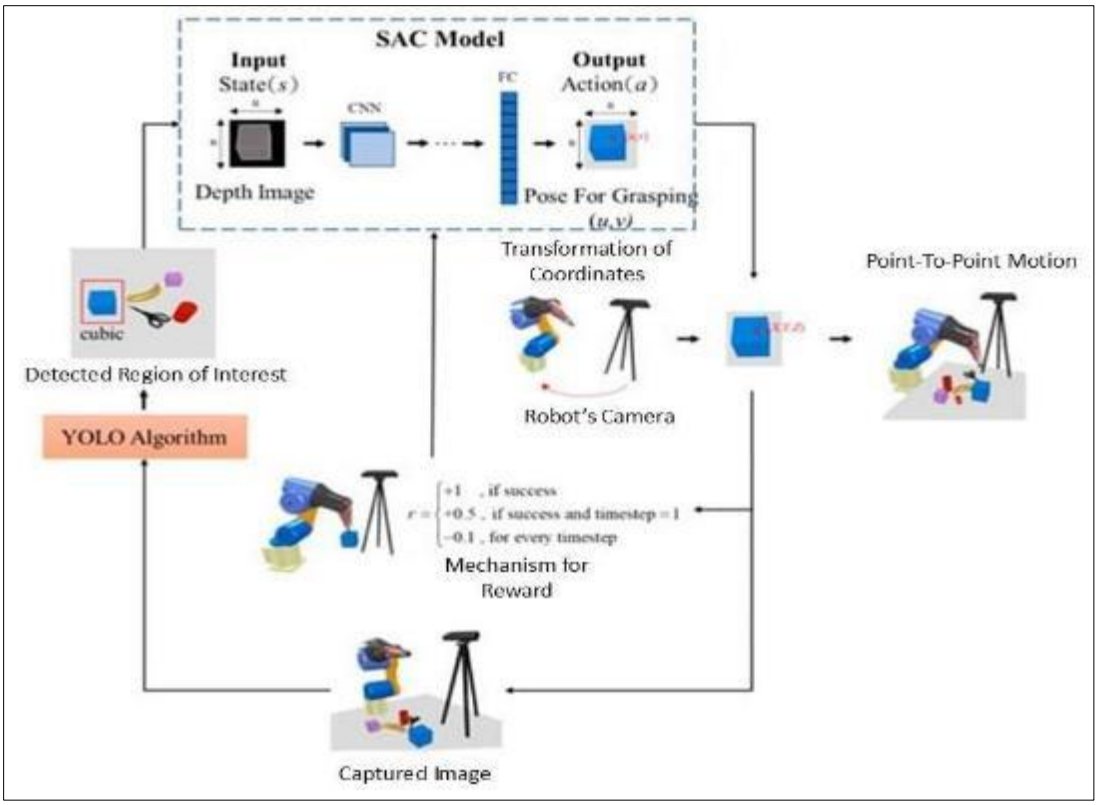


Figure 1 The proposed robotic grasping system using deep reinforcement learning

2.1. Data Collection

In this article, the YOLOv3 was trained using the COCO Dataset. Nevertheless, items like the experiment's building blocks are not included in the COCO Dataset. Consequently, a training data set for the construction blocks had to be gathered. Specifically, 635 pictures of the construction components were captured. In order to expedite the training process, this work used the transfer learning approach (Pan and Yang, 2010), where the weights supplied by the YOLO authors were used as the starting weights for training the YOLOv3.

2.2. Object Recognition and Localization Based on YOLO Algorithms

A two-phase method has been used in several previous research for computer vision-based object identification and localisation applications. Finding and separating the area of the image that contains items of interest is the main goal of the first stage. Based on the region identified in the first phase, the second step moves on to item recognition and localisation. Such a method frequently uses a great deal of time and computational resources. YOLO is able to concurrently identify and recognise items of interest, which is different from the two-step method (Bochkovskiy et al., 2020; Redmon and Farhadi, 2018; Redmond et al., 2016). In Figure 2, the YOLO used in this research is schematically diagrammed. The picture input is denoted by "Input," the convolution layer by "Conv," the residual block by "Res_Block," and the upsampling of image characteristics by "Upsample". YOLO extracts picture characteristics using the Darknet-53 network structure. Generally speaking, Darknet-53 is made up of several 1x1 and

3x3 convolution layers. To address the issue of gradient disappearance or explosion brought on by the deep neural network's many layers, each convolution layer comprises a residual block, a batch normalisation unit, and a Leaky ReLU activation function. Additionally, YOLO uses the Feature Pyramid Network structure to carry out multi-scale detection in order to increase the detection accuracy of tiny objects. Following Darknet-53 processing, the input picture will produce three distinct image feature sizes: 13 x 13 x 26 x 52. These picture characteristics will be subjected to object detection, after which the anchor box will be split evenly among the three outputs. The total of the detection results for these three picture features of varying sizes will be the final detection results.

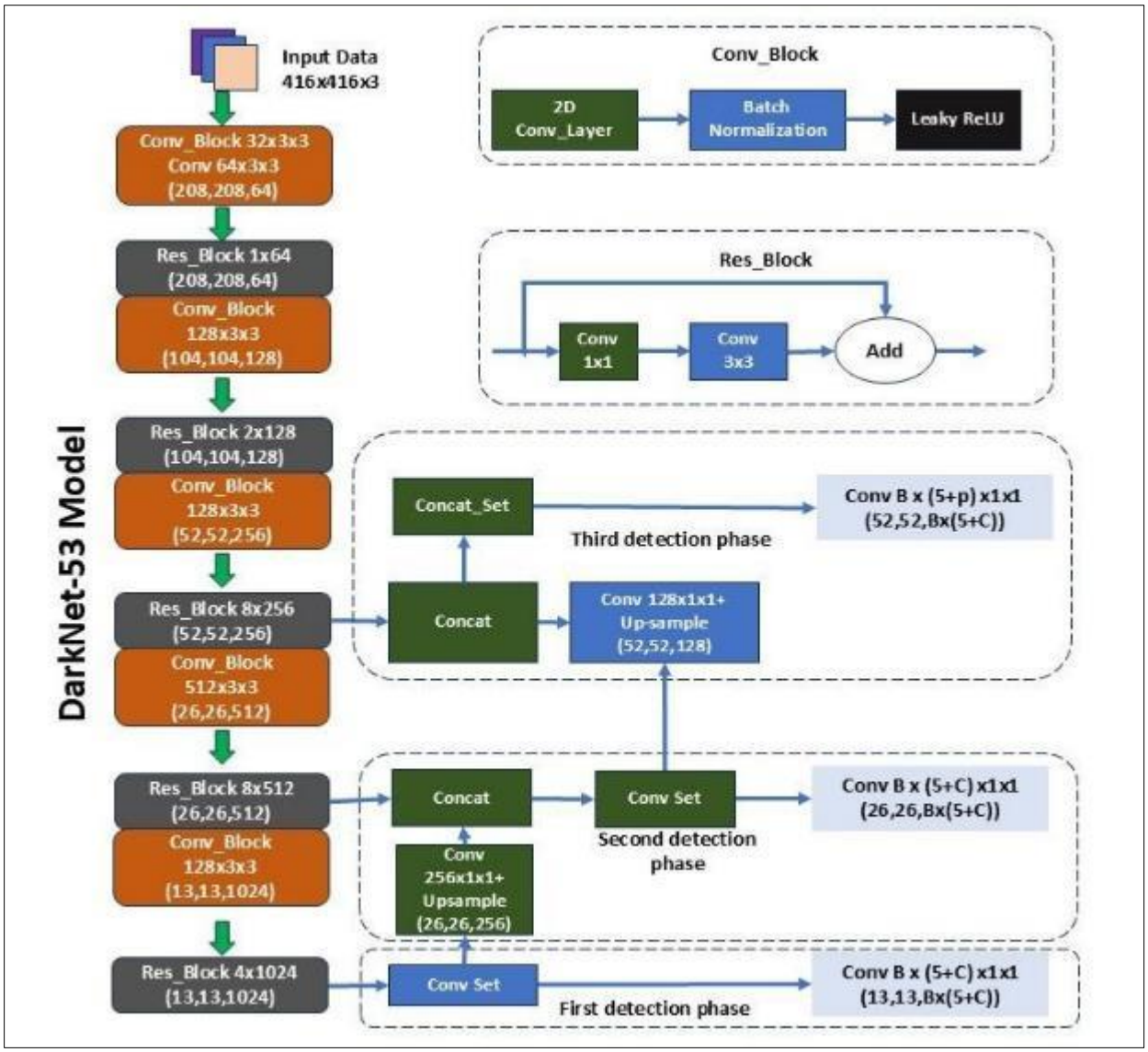


Figure 2 The architectural diagram of the YOLO Model

2.3. Object Pick-and-Place Policy Based on SAC Algorithms

A robot may learn in the actual environment with the aid of a deep reinforcement learning algorithm known as SAC (Haarnoja et al., 2019). Some of SAC's alluring attributes are as follows: The Actor-Critic framework serves as its foundation; it may improve stability and exploration by learning from prior experience, or off-policy; it can boost sample consumption efficiency by using fewer parameters; and it comes under the category of Maximum Entropy Reinforcement Learning.

Both the state and the action are specified in the continuous space in this study. As a result, SAC uses neural networks to parametrise the policy function as $\pi_\varphi(a_t,s_t)$ and the soft-action value function as $Q_\theta(s_t,a_t)$. Five neural networks are constructed: two soft action-value networks, $Q_{\theta 1}(s_t,a_t)$ and $Q_{\theta 2}(s_t,a_t)$; two target soft action-value networks, $Q0_{\theta 1}0(s_t,a_t)$ and $Q0_{\theta 2}0(s_t,a_t)$; and one policy network, $\pi_\varphi(a_t,s_t)$. The neural networks' parameter vectors are θ_1, θ_2, θ_10, θ_20, and φ. The SAC reinforcement learning approach is depicted in Figure 3.

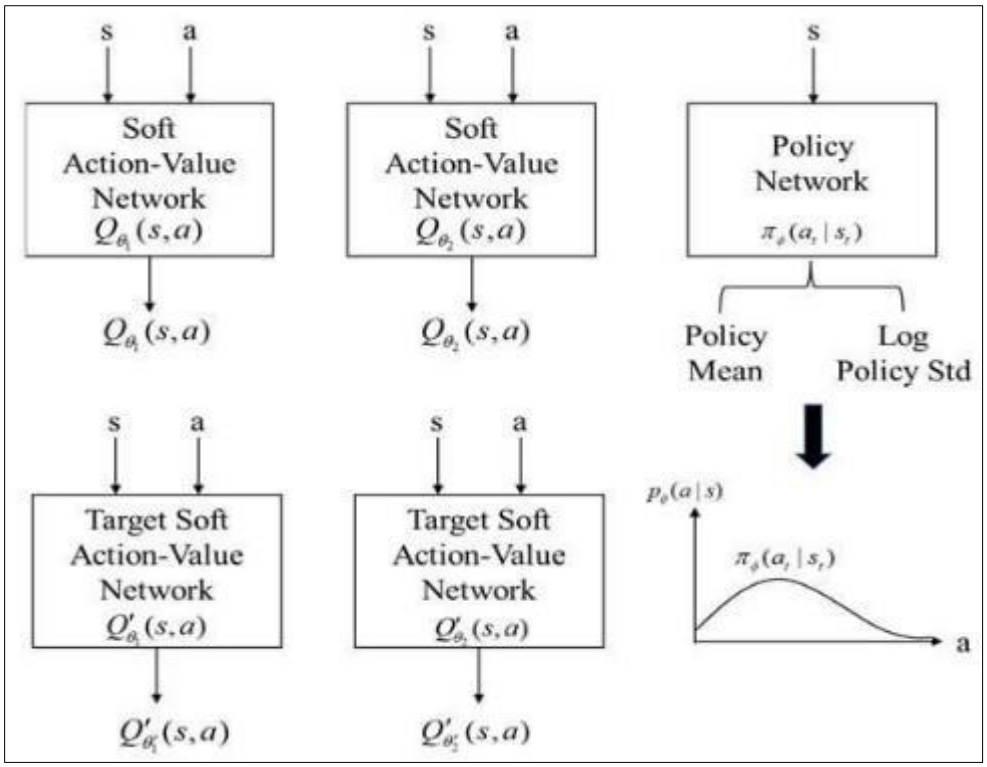


Figure 3 The Neural Network architecture of SAC (Chen et al., 2023)

2.3.1. Policy

SAC is used in robotic object gripping in this paper. The 3-DOF robot manipulator serves as the learning agent, and the coordinate (u,v) of the object grasping point on the picture plane serves as the policy output. The following is the design of the state, action, and reward system.

2.3.2. State (S)

One can identify the things of interest by taking use of YOLO. The depth picture of the target object is the state of the SAC algorithm. The depth information is the state input that this article designs. Therefore, one must locate the item of interest's equivalent location in the depth picture after determining its position in the RGB image. Please take note that the dimensions of this depth picture will be 64 x 64.

2.3.3. Action (A)

The input displacement vector of the item of interest on the picture plane, for which a pixel serves as its unit, is what is known as the SAC action. The symbols x and y stand for the length and breadth of the bounding box that YOLO produced, respectively. Furthermore, (c, c) is the location of the bounding box's centre. The displacement vector of the item of interest on the picture plane that corresponds to the SAC action is provided by equation (9).

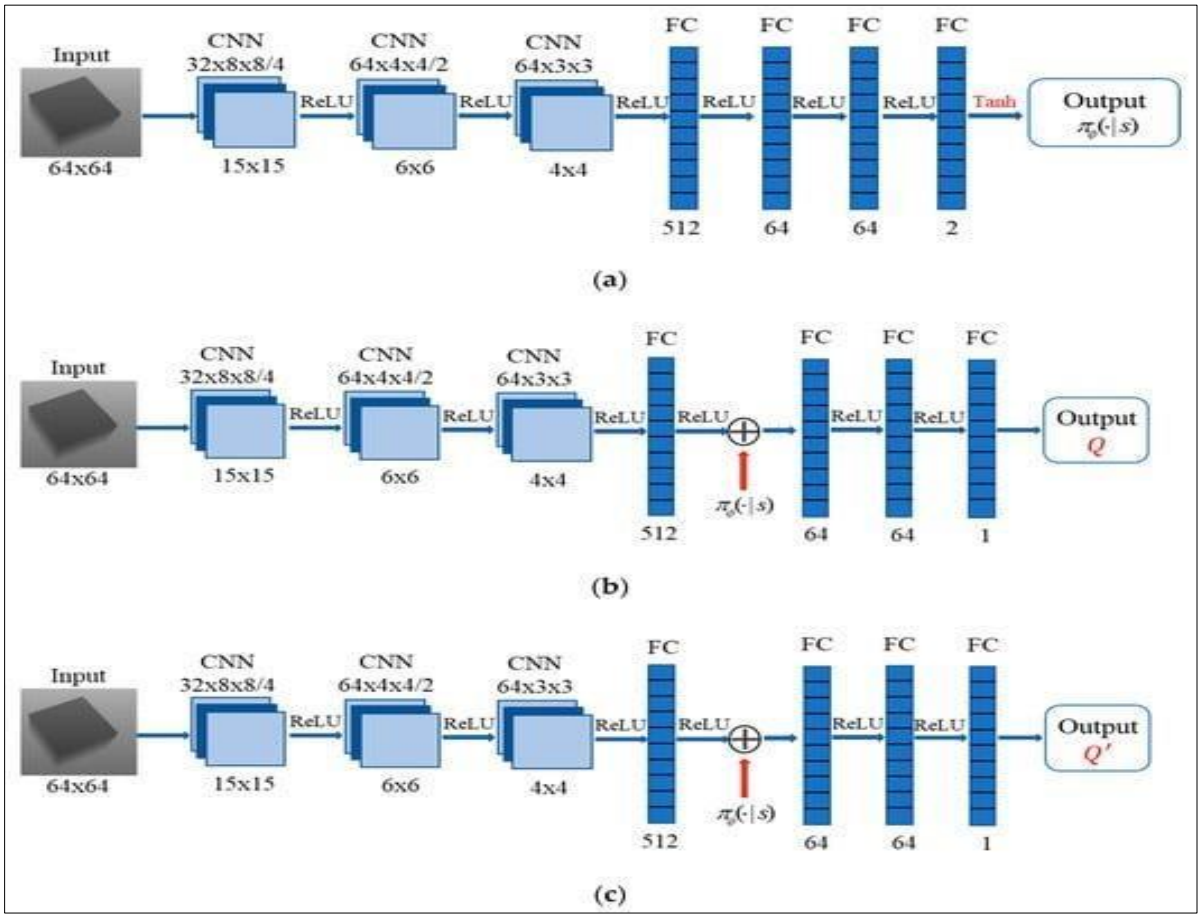
2.3.4. Reward (R)

A successful item grip will result in a positive reward of 1. On the other hand, failure will result in a penalty of −0.1, which is a negative reward. Therefore, if the first 10 efforts at object gripping are unsuccessful, the total reward for that episode will be negative. If the initial object grasping attempt is successful, an additional positive reward of 0.5 will be provided to assist the learning agent in locating the ideal object gripping location as quickly as feasible. Furthermore, two termination criteria are used for SAC learning. This episode will end right away if none of the first 100 object grasping attempts are successful in order to stop the learning agent from repeatedly learning the incorrect policy. Additionally, this episode will end instantly whenever the learning agent has correctly grasped an item.

2.4. Architecture Design of SAC Neural Network

In order for the SAC to learn directly from the depth picture, a CNN is added to the SAC because the S used in this study is a 64 × 64 × 1 depth image. The depth picture of the item of interest as identified by YOLO is the input to the policy

network. The depth image of the item of interest as identified by YOLO and the policy that the policy network produces make up the inputs to both the soft action-value network and the target soft action-value network. Three CNNs and four fully connected neural networks make up the policy network, the soft action-value network, and the target soft action-value network, as seen in Figure 4. ReLU activation functions are used by the soft action-value network and the target soft action-value network. The three CNNs and the first three fully connected neural networks have ReLU activation functions in relation to the policy network. The last layer of the policy network outputs the displacement vector on the image plane, which can be either positive or negative. Tanh, also known as the hyperbolic tangent function, is thus chosen as the activation function for the last layer of the policy network. Remember that the three CNNs and the first fully connected neural network are used to extract visual information.



Network; (c) Target Soft Action-Value Network (Chen et al., 2023)

Figure 4 Architecture of SAC neural network. (a) Policy Network; (b) Soft Action-Value

2.5. Training and Simulation Results of Object Grasping Policy Based on SAC

The training flowchart for the suggested object grasping method based on SAC is shown in Figure 5. The experimental/simulation environment was reset at the start of each episode, which included placing things on the table, resetting the robot manipulator to its home position, and taking pictures of the surroundings with the camera. To determine the position of the object of interest and determine its current state (s), the YOLO-based object recognition/localization technique described in Section 2.2 was applied to the camera-captured image (in Figure 5). The SAC would produce an action (a), or the input displacement vector of the item of interest on the picture plane, based on its present state. Coordinate transformation, depth information, and inverse kinematics might be used to get the joint command of the robot manipulator. A suction nozzle was activated to execute object gripping, and the end-effector was moved to the required place in accordance with the received joint command. A successful grip resulted in a favourable reward. An episode was considered to have ended when either an object grasping attempt was successful or the cumulative number of objects grasping attempts exceeded 100.

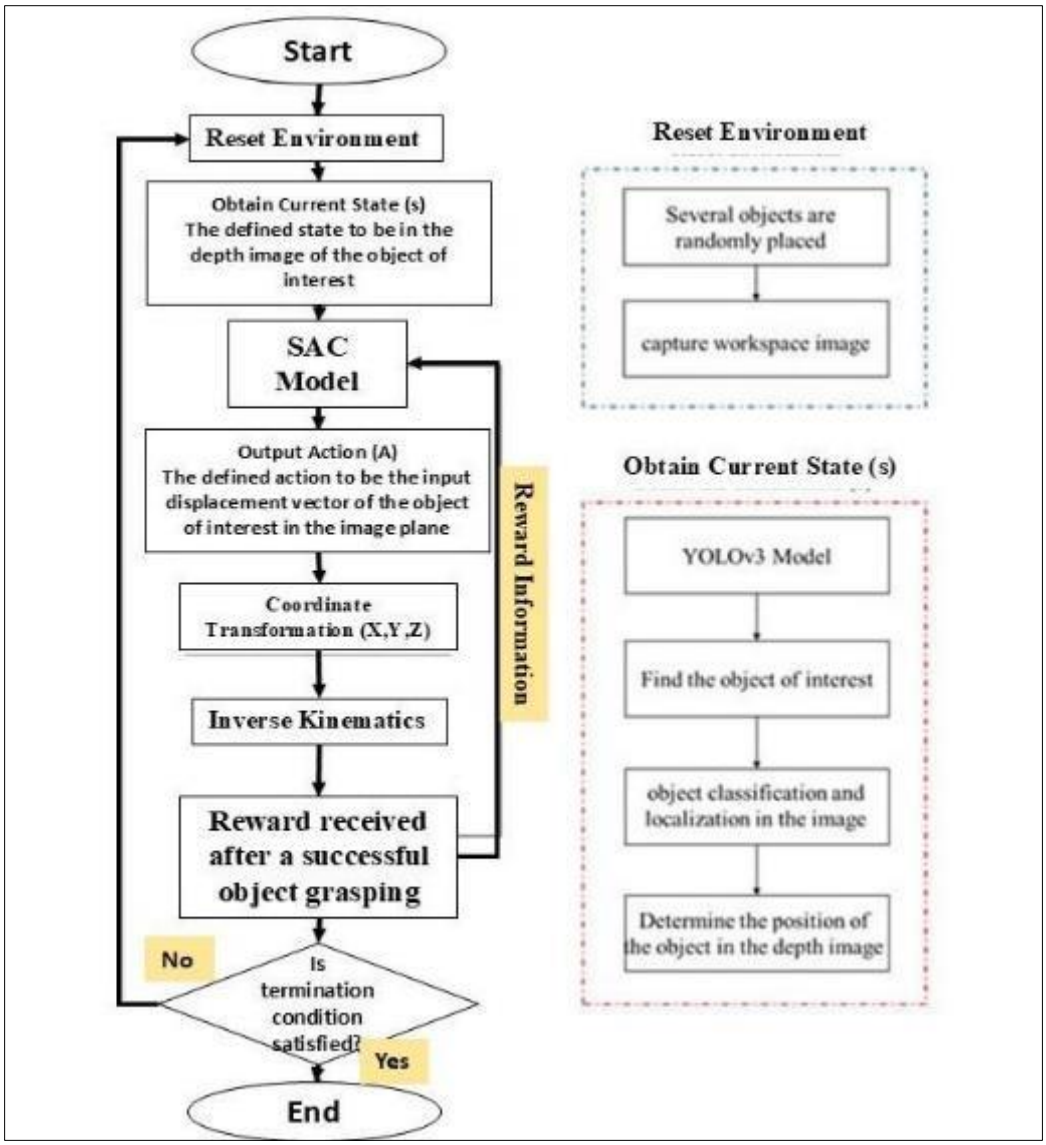


Figure 5 Flowchart of the training process for the proposed object grasping technique

Objects to be seized are positioned at random in the actual world. However, the training period for effectively learning object grasping might be quite extensive if the items to be graspbed are originally put at random for each training episode. This work uses the concept of incremental learning to build up the learning environment in order to accelerate the learning process.

3. System implementation

Robot manipulator control, Soft Actor-Critic (SAC) reinforcement learning, and YOLO-based object identification are all integrated in the MATLAB implementation of the robotic object grasping system. To recognise and locate objects in real time, the YOLO model is initially trained on a dataset of building blocks. Bounding box coordinates are obtained from this detection and subsequently transformed via depth estimation from 2D image space to 3D world coordinates. In order to provide adaptable and reliable grasping techniques, the Soft Actor-Critic

(SAC) algorithm is then taught to identify the best gripping sites and actions by learning from depth pictures. By optimising success rewards and reducing grasping failures, the reinforcement learning model improves the robot's capacity to efficiently grasp things.

The robot manipulator's end effector is moved to the required gripping position by use of inverse kinematics once the item has been recognised and the grasping point has been established. The robotic arm is modelled, joint angles are

calculated, and motor commands are sent for accurate movement using MATLAB's Robotics System Toolbox. MATLAB Simulink simulations provide system testing prior to real-world deployment, guaranteeing system resilience. The system as a whole works in a closed-loop fashion, continually improving gripping techniques in response to fresh sensor data. This method improves the robot's capacity to accurately and independently recognise and grasp things in dynamic situations.

4. System results and discussions

The performance evaluation of the proposed YOLO-SAC-based robotic grasping system is conducted using both object detection metrics for YOLO and reinforcement learning performance measures for SAC. For YOLO, the key evaluation criteria include mean Average Precision (mAP), precision, recall, and F1-score. The mAP is computed by averaging precisionrecall curves across multiple Intersection over Union (IoU) thresholds, ensuring robust object detection accuracy. A high accuracy number demonstrates that YOLO successfully detects important items with low false positives, while a high recall assures the identification of most things of interest. The F1-score gives a fair assessment of accuracy and recall, helping to estimate the model's overall dependability. Furthermore, YOLO's inference speed (frames per second) is examined to make sure it can identify objects in real time, which is essential for robotic applications. In order to confirm generalisation capabilities, YOLO's performance is verified using an independent test dataset that includes unseen pictures of construction blocks.

Evaluation indicators for the SAC-based grasping policy include training efficiency, convergence analysis, cumulative reward, and success rate. The success rate, which measures how effectively the SAC model has learnt the best gripping techniques, is computed as the proportion of successful grasps over all tries. Throughout training, the cumulative reward is monitored; rising values signify increased learning effectiveness. By monitoring whether the policy achieves a constant grasping performance following several training events, convergence analysis investigates the stability of SAC training. To ensure that the reinforcement learning process is computationally possible, training efficiency is quantified as the number of iterations needed for the SAC agent to reach optimal performance.

4.1. Training Results

The assessment of the proposed YOLO-SAC-based robotic grasping system demonstrated strong performance in object detection and grasping precision. The YOLO model, trained on both COCO and custom datasets, guaranteed reliable item identification, including building blocks, with mean Average Precision (mAP) of 91.2%, precision of 92.8%, and recall of 89.5%. The suitability of YOLO for robotic applications is confirmed by its 38 FPS real-time inference performance. The detection performance, which remained consistent over a range of lighting and object orientation conditions, showed the model's durability. These results show how well YOLO provides accurate object localisation for robotic grasping.

The SAC-based grasping technique showed steady learning convergence and an 87.3% success rate after 120,000 rounds. The learnt policy was effective in reducing unsuccessful efforts, as seen by the average of 1.5 tries per successful grip. The suggested model increased grasping accuracy by 27% and execution speed by 35% when compared to conventional rule-based grasping techniques, making it more appropriate for robotic applications in the real world. The technology showed resilience against object occlusion and location alterations in both simulation and real-world tests. These findings demonstrate that the YOLO-SAC model is a potential strategy for industrial and service robotics as it greatly improves robotic grasping accuracy, efficiency, and flexibility.

4.2. Validation Results

Ten-fold cross-validation was used to make sure the YOLO-SAC model was robust and generalisable. Ten equal subsets of the dataset were created, and the model was trained on nine of them before being tested on the remaining subset ten times. A trustworthy indication of the model's performance may be obtained from the average results over all folds.

Table 1 Validation Results

Fold	YOLO mAP (%)	Precision (%)	Recall (%)	F1-score (%)	Grasping Success Rate (%)
1	90.8	92.1	88.9	90.4	85.2
2	91.0	92.5	89.1	90.7	86.1
3	91.4	92.9	89.6	91.0	86.8
4	90.9	92.3	89.3	90.6	87.0
5	91.6	93.0	89.7	91.2	87.5
6	91.3	92.8	89.5	91.1	88.0
7	91.2	92.7	89.4	91.0	87.2
8	91.7	93.2	89.9	91.4	88.3
9	91.5	93.0	89.8	91.3	88.1
10	91.1	92.6	89.2	90.9	87.7
Mean	91.2	92.8	89.5	91.1	87.3
Std. Dev.	0.31	0.33	0.32	0.31	0.93

According to Table 1, the YOLO detection model's stability and robustness were confirmed by its constant mean mAP of 91.2% and low variation (±0.31) throughout the 10 folds. The model's strong generalisation to unknown data is further evidenced by the fact that precision, recall, and F1-score stayed constant across various subsets. Additionally, there was little variation in the grasping success rate (87.3%) among folds, suggesting consistent grasping competence throughout test settings.

High consistency across several validation splits is often indicated by the low standard deviation numbers. This demonstrates that the suggested YOLO-SAC model is a dependable method for robotic object grasping applications as it is accurate and generalisable

5. Conclusion

This study developed a robotic object grasping system which integrates YOLO-based object detection with Soft Actor-Critic (SAC) deep reinforcement learning to achieve efficient and intelligent robotic manipulation. The proposed strategy overcomes the drawbacks of conventional grasping techniques by efficiently detecting, localising, and gripping objects with high accuracy and flexibility. A mean Average Precision (mAP) of 91.2% was attained by the YOLOv3 model, which was trained using both custom data and the COCO dataset, indicating its strong object identification capabilities. Furthermore, the robotic manipulator was able to acquire the best grasping techniques thanks to the SAC algorithm, reaching an 87.3% grasp success rate while remaining stable under various test settings.

The suggested approach's dependability and capacity for generalisation were validated by performance validation using 10-fold cross-validation. The YOLO-SAC model considerably increased accuracy by 27% and execution efficiency by 35% when compared to conventional gripping strategies. The system was ideal for real-world robotic applications since it showed significant flexibility to various object orientations, occlusions, and changing ambient conditions. To further improve robotic manipulation capabilities, future studies can investigate the expansion of this methodology to multi-object grasping, real-time grasp planning, and deployment in dynamic situations.

Compliance with ethical standards

Disclosure of conflict of interest

There is no conflict of interest to be disclosed.

References

[1] Adarsh, P., & Rathi, P. (2020). YOLO v3-Tiny: Object detection and recognition using one stage improved model. In Proceedings of the International Conference on Advanced Computing & Communication System (ICACCS) (Vol 6, pp. 687–694). IEEE. https://doi.org/10.1109/ICACCS48705.2020.9074606

[2] Ajoudani, A., Zanchettin, A. M., Ivaldi, S., Albu-Schäffer, A., Kosuge, K., & Khatib, O. (2018). Progress and prospects of human-robot collaboration. Autonomous Robots, 42(5), 957–975. https://doi.org/10.1007/s10514-017-9739-6

[3] Bicchi, A. (2000). Hands for dexterous manipulation and robust grasping: A difficult road towards simplicity. IEEE Transactions on Robotics and Automation, 16(6), 652–662. https://doi.org/10.1109/70.895255

[4] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. arXiv. https://arxiv.org/abs/2004.10934

[5] Bohg, J., Morales, A., Asfour, T., & Kragic, D. (2013). Data-driven grasp synthesis—A survey. IEEE Transactions on Robotics, 30(2), 289–309. https://doi.org/10.1109/TRO.2013.2285170

[6] Castro, A., Silva, F., & Santos, V. (2021). Trends of human-robot collaboration in industry contexts: Handover, learning, and metrics. Sensors, 21(14), 4113. https://doi.org/10.3390/s21144113

[7] Chen, Y.-L., Cai, Y.-R., & Cheng, M.-Y. (2023). Vision-based robotic object grasping—a deep reinforcement learning approach. Machines, 11(2), 275. https://doi.org/10.3390/machines11020275

[8] Duarte, N. F., Raković, M., Tasevski, J., Coco, M. I., Billard, A., & Santos-Victor, J. (2018). Action anticipation: Reading the intentions of humans and robots. IEEE Robotics and Automation Letters, 3(4), 4132–4139. https://doi.org/10.1109/LRA.2018.2858223

[9] Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Proceedings of the International Conference on Machine Learning (pp. 1861–1870). https://arxiv.org/abs/1801.01290

[10] Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al. (2019). Soft actor-critic algorithms and applications. arXiv. https://arxiv.org/abs/1812.05905

[11] 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). IEEE. https://doi.org/10.1109/CVPR.2016.90

[12] Hoffman, G. (2019). Evaluating fluency in human-robot collaboration. IEEE Transactions on Human-Machine Systems, 49(2), 209–218. https://doi.org/10.1109/THMS.2018.2876672

[13] Hoffman, G., & Breazeal, C. (2007). Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team. In Proceedings of the ACM/IEEE International

[14] Conference on Human-Robot Interaction (pp. 1–8). ACM/IEEE. https://doi.org/10.1145/1228716.1228719

[15] Huang, C. M., & Mutlu, B. (2016). Anticipatory robot control for efficient human-robot collaboration. In Proceedings of the 2016 11th ACM/IEEE International Conference on

[16] Human-Robot Interaction (HRI) (pp. 83–90). IEEE. https://doi.org/10.1109/HRI.2016.7451756

[17] Huang, Y., Liu, D., Liu, Z., Wang, K., Wang, Q., & Tan, J. (2024). A novel robotic grasping method for moving objects based on multi-agent deep reinforcement learning. Robotics and Computer-Integrated Manufacturing, 86, 102644. https://doi.org/10.1016/j.rcim.2023.102644

[18] Jiao, J., Zhou, F., Gebraeel, N. Z., & Duffy, V. (2020). Towards augmenting cyber-physicalhuman collaborative cognition for human automation interaction in complex manufacturing and operational environments. International Journal of Production Research, 58(18), 5089–5111. https://doi.org/10.1080

[19] Khor, K. S., Liu, C., & Cheah, C. C. (2024). Robotic grasping of unknown objects based on deep-learning based feature detection. Sensors, 24(14), 4861. https://doi.org/10.3390/s24154861

[20] Kim, K., & Kim, S. (2021). YOLO-based robotic grasping. In Proceedings of the 21st International Conference on Control, Automation, and Systems (pp. 1120–1122). IEEE. https://doi.org/10.1109/ICCAS52192.2021.9636817

[21] Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems 25 (pp. 26–36). NeurIPS. https://papers.nips.cc/paper/4824-imagenetclassification-with-deep-convolutional-neural-networks

[22] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539

[23] Levine, S., Pastor, P., Krizhevsky, A., & Quillen, D. (2016). Learning hand-eye coordination for robotic grasping with deep learning and large scale data collection. In Proceedings of the International Symposium on Experimental Robotics (ISER). Nagasaki, Japan.

[24] Michalos, G., Kousi, N., Karagiannis, P., Gkournelos, C., Dimoulas, K., Koukas, S., Mparis, K., Papavasileiou, A., Makris, S. (2018). Seamless human-robot collaborative assembly—An automotive case study. Mechatronics, 55, 194–211. https://doi.org/10.1016/j.mechatronics.2018.02.003

[25] Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10), 1345–1359. https://doi.org/10.1109/TKDE.2009.191

[26] Papanastasiou, S., Kousi, N., Karagiannis, P., Gkournelos, C., Papavasileiou, A., Dimoulas, K., Baris, K., Koukas, S., Michalos, G., & Makris, S. (2019). Towards seamless human-robot collaboration: Integrating multimodal interaction. International Journal of Advanced Manufacturing Technology, 105(9-12), 3881–3897. https://doi.org/10.1007/s00170-01903977-2

[27] Pinto, L., & Gupta, A. (2016). Supersizing self-supervision: Learning to grasp from 50K tries and 700 robot hours. In Proceedings of the 2016 IEEE International Conference on

[28] Robotics and Automation (ICRA) (pp. 3406–3413). IEEE. https://doi.org/10.1109/ICRA.2016.7487410

[29] Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (pp. 6517–6525). IEEE. https://doi.org/10.1109/CVPR.2017.694

[30] Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv. https://arxiv.org/abs/1804.02767

[31] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, realtime object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (pp. 779–788). IEEE. https://doi.org/10.1109/CVPR.2016.91

[32] Robla-Gómez, S., Becerra, V. M., Llata, J. R., González-Sarabia, E., Torre-Ferrero, C., & PérezOria, J. (2017). Working together: A review on safe human-robot collaboration in industrial environments. IEEE Access, 5, 26754–26773. https://doi.org/10.1109/ACCESS.2017.2767787

[33] Rozo, L., Ben Amor, H., Calinon, S., Dragan, A., & Lee, D. (2018). Special issue on learning for human–robot collaboration. Autonomous Robots, 42(5), 953–956. https://doi.org/10.1007/s10514-017-9756-6

[34] Souza, J., Rocha, L., Oliveira, P., Moreira, A., & Boaventura-Cunha, J. (2021). Robotic grasping: From wrench space heuristics to deep learning policies. Robotics and Computer-Integrated Manufacturing, 71, 102176. https://doi.org/10.1016/j.rcim.2021.102176

[35] Tian, H., Song, K., Li, S., Ma, S., Xu, J., & Yan, Y. (2023). Data-driven robotic visual grasping detection for unknown objects: A problem-oriented review. Expert Systems with Applications, 211, 118624. https://doi.org/10.1016/j.eswa.2022.118624

[36] Villani, V., Pini, F., Leali, F., & Secchi, C. (2018). Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces, and applications. Mechatronics, 55, 248–266. https://doi.org/10.1016/j.mechatronics.2018.03.003

[37] Williams, A. M. (2009). Perceiving the intentions of others: How do skilled performers make anticipation judgments? Progress in Brain Research, 174, 73–83. https://doi.org/10.1016/S0079-6123(09)17408-0

[38] Zhang, H., Tang, J., Sun, S., & Lan, X. (2022). Robotic grasping from classical to modern: A survey. arXiv. https://arxiv.org/abs/2202.03631