IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 17, 2022, accepted February 3, 2022, date of publication February 15, 2022, date of current version February 28, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3151717

Moving Object Prediction and Grasping System of Robot Manipulator

CHING-CHANG WONG[1], MING-YI CHIEN[1], REN-JIE CHEN[1], HISAYUKI AOYAMA[2], AND KAI-YI WONG[3]

[1]Department of Electrical and Computer Engineering, Tamkang University, New Taipei City 25137, Taiwan
[2]Department of Mechanical and Intelligent Systems Engineering, The University of Electro-Communications, Tokyo 182-8585, Japan
[3]Department of Electrical Engineering, Chung Yuan Christian University, Taoyuan City 32023, Taiwan

Corresponding author: Kai-Yi Wong (kywong@cycu.edu.tw)

This work was supported in part by the Ministry of Science and Technology (MOST), Taiwan, under Grant MOST 109-2221-E-032-038 and Grant MOST 110-2221-E-032-046.

ABSTRACT In this paper, we designed and implemented a moving object prediction and grasping system that enables a robot manipulator using a two-finger gripper to grasp moving objects on a conveyor and a circular rotating platform. There are three main parts: (i) moving object recognition, (ii) moving object prediction, and (iii) system realization and verification. In the moving object recognition, we used the instance segmentation algorithm of You Only Look At CoefficienTs (YOLACT) to recognize moving objects. The recognition speed of YOLACT can reach more than 30 fps, which is very suitable for dynamic object recognition. In addition, we designed an object numbering system based on object matching, so that the system can track the target object correctly. In the moving object prediction, we first designed a moving position prediction network based on Long Short-Term Memory (LSTM) and a grasping point prediction network based on Convolutional Neural Network (CNN). Then we combined these two networks and designed two moving object prediction networks, so that they can simultaneously predict the grasping positions and grasping angles of multiple moving objects based on image information. In the system realization and verification, we used Robot Operating System (ROS) to effectively integrate all the programs of the proposed system for the camera image extraction, strategy processing, and robot manipulator and gripper control. A laboratory-made conveyor and a circular rotating platform and four different objects were used to verify that the implemented system could indeed allow the gripper to successfully grasp moving objects on these two different object moving platforms.

INDEX TERMS Moving object prediction, object grasping, long short-term memory (LSTM), convolutional neural network (CNN), you only look at the coefficients (YOLACT).

I. INTRODUCTION

Object picking and placing is a fundamental but challenging task in robot manipulation due to the various sizes, shapes, and other properties of objects [1]. In addition, the implementation of object picking and placing system in a dynamic environment is more challenging than that in a static environment. Nowadays, conveyors and circular rotating platforms are widely used in distribution, warehousing, manufacturing, and production in factories for automation and faster delivery [2]. Two common types of robot manipulators used to pick and place objects are the suction method using vacuum chucks and the grasping method using two-finger grippers. For the task of picking and placing objects of the robot manipulator, the development in a static environment has achieved good results. For technical and cost reasons, most scenes use the suction methods to suck and place objects in dynamic environments. The advantage of the suction methods is that the system sucks the object without the need to consider the suction angle of the object, and it does not need to have a good position prediction of the moving object. However, the grasping method can grasp objects more diversely than the suction method because it is less restricted by the surface shape of the object. Therefore, we investigate an object grasping method for a six-degree of freedom robot manipulator with a two-finger gripper.

In terms of object grasping using two-finger grippers, most applications only focus on grasping static objects, because the pose, moving trajectory, and grasping efficiency of the target object need to be considered. There are many challenges for a

The associate editor coordinating the review of this manuscript and approving it for publication was Christopher H. T. Lee.

robot manipulator using a two-finger gripper to grasp objects in a dynamic environment. The system using the grasping method needs to have the same ability to recognize moving objects as the system using the suction method, and it also needs to have a good ability to predict the future position of the moving object [3]–[5]. In addition, if the posture of the object changes during the movement, the system also needs to have the ability to predict the future posture of the moving object. How to combine technologies such as visual tracking and object grasping to enable the robot manipulator to successfully grasp moving objects on an object moving platform is a challenging question and a research direction worthy of discussion. There are many papers on robotic grasping, but few papers study both moving object prediction and moving object grasping. The method proposed by Allen *et al.* [6] can grasp a moving object, but it can only track a single object, and the range that can be grasped is only on a fixed track (a single fixed trajectory). And it uses the slope of the track as the grasping angle, which is not determined by the system. Therefore in this paper, we used an instance segmentation algorithm to recognize multiple moving objects on two different object moving platforms (a conveyor and a circular rotating platform), and designed two moving object prediction networks to simultaneously predict grasping positions and grasping angles of multiple objects. In addition, we used Robot Operating System (ROS) to realize the proposed system. In this study, we need to integrate methods such as moving object recognition, moving object prediction, and system realization. Some related works are introduced as follows:

In the related research of object recognition, the related models have been innovated continuously in recent years. From the earliest development, R-CNN (Region-based Convolutional Neural Networks) [7], Fast R-CNN [8], and Faster R-CNN [9] are two-stage high-precision methods. These methods take out the bounding box of the object and then classify it. The accuracy of two-stage methods is high, but the operation speed is slow. Therefore, one-stage high-speed methods such as YOLO (You Only Look Once) [10], SSD (Single Shot MultiBox Detector) [11], and R-FCN (Region-based Fully Convolutional Network) [12] were developed to meet the requirements of real-time object recognition. Among them, the YOLO series [10], [13], [14] are the more commonly used methods. Although the accuracy of one-stage methods is lower than that of two-stage methods, the impact of its lower recognition accuracy is within an acceptable range. Therefore, many networks for object recognition have developed. For example, the FCN [15], SegNet [16], and DeepLab [17] are the semantic segmentation networks, and YOLACT [18], FCIS (Fully Convolutional Instance-aware Semantic Segmentation) [19], PA-Net (Path Aggregation Network) [20], Mask R-CNN [21], and Mask Scoring R-CNN [22] are the instance segmentation networks. Both instance segmentation and semantic segmentation can obtain the contour of the object to achieve accurate object recognition. The difference between instance segmentation and semantic segmentation is that instance segmentation can independently segment objects of the same category, while semantic segmentation cannot. Therefore, we used instance segmentation to implement the object recognition.

In the related research of moving object prediction, the prediction of the moving position of a moving object is a very important part in dynamic environments. In order to make the predicted position more accurate, deep learning methods are used to predict. One of the most commonly used network architectures is the Long Short-Term Memory (LSTM) network [23]. This network can analyze time series relationships to make predictions efficiently. In addition, in order to obtain the suitable grasping position of the object, Convolutional Neural Networks (CNNs) are often used to learn. The convolutional layer, pooling layer, and fully connected layer are used to perform feature extraction and analysis, and to predict the grasping pose of the object. Therefore, we combined two network models of LSTM and CNNs to implement the moving object prediction.

In the related research of system realization, Robot Operating System (ROS) is an open source operating system that can improve the development efficiency of robot systems. Kumra *et al.* [24] used ROS to construct a neural network-based robot grasping system, Hernandez-Mendez *et al.* [25] constructed a 3-DOF robot manipulator based on ROS, Wang *et al.* [26] constructed a mobile robotic arm platform for detecting and grasping radiation sources based on ROS, and Wong *et al.* [27], [28] used ROS and Gazebo to design and simulate the motion planning and manipulation planning of the robot manipulator. Therefore, we used ROS to realize the proposed system.

The rest of this paper is organized as follows. Section II introduces the system architecture of this paper and describes the relationship of the proposed system. Section III introduces the moving object recognition method and the developed object numbering system used in this system. Section IV introduces the two proposed moving object prediction networks. Section V explains how to use ROS to integrate all the programs of the image, strategy, and control system, and how to develop the topic and service functions required by the system for hardware devices such as camera, robot manipulator, and two-finger gripper. Section VI introduces and discusses various tests and experimental results of the proposed system on four experimental objects and two object moving platforms, verifying the effectiveness of the proposed moving object prediction and grasping system. Finally, Section VII is the conclusion.

II. SYSTEM ARCHITECTURE

We design and implement a moving object prediction and grasping system so that the manipulator can grasp moving objects. In the hardware part, three input/output devices are an RGB-D camera (RealSense D435), a six-degree of freedom robot manipulator (UR5), and a two-finger gripper (Robotiq 2F-85). In the software part, the Robot Operating System (ROS), which is easy to develop robot systems, is used to

integrate all programs designed for the proposed system. The overall system architecture is shown in Fig. 1. There are three main parts: (i) moving object recognition, (ii) moving object prediction, and (iii) control planning.

In the moving object recognition, three parts are planned and implemented: object detection, data augmentation, and object numbering. We first used the camera to obtain RGB images, and then used You Only Look At CoefficienTs (YOLACT) to implement the object detection of the proposed system to recognize the objects in the RGB image. Since YOLACT requires a lot of training data, we first used a data augmentation method to quickly generate some training data required for YOLACT's network training, so that the trained YOLACT can get more accurate object information, such as category, confidence value, bounding box, and mask of the object. In addition, we designed an object numbering system. The objects are numbered after being recognized by YOLACT, and the numbering will remain consistent.

In the moving object prediction, we combined the time series analysis capabilities of LSTM and the image recognition capabilities of CNN to design two moving object prediction networks. This architecture takes the mask image of the object obtained by YOLACT as input and obtains five future grasping positions and grasping angles of the moving object.

In the control planning, we planned and realized three parts: strategy, MoveIt, and gripper control. We designed strategy to control the robot manipulator and two-finger gripper, and used the MoveIt suite [29] of ROS to find the solution of the forward and inverse kinematics of the robot manipulator and do the trajectory planning. The gripper control cooperates with the movement of the robot manipulator to timely send out the control commands of the gripper so that the system can successfully complete the task of picking and placing moving objects.

III. MOVING OBJECT RECOGNITION

We used deep neural networks to implement a moving object recognition method to recognize moving objects. It requires a lot of training data, so a data augmentation method is used to increase the training data. During network training, we used a GeForce GTX 1070 GPU for training to speed up the object recognition operations. We also use GPU in the proposed system to speed up the system's operation speed for real-time object recognition. The proposed moving object recognition system can be separated into three main parts: (a) object detection, (b) data augmentation, and (c) object numbering. They are described as follows:

A. OBJECT DETECTION

We used YOLACT to implement the object detection of the proposed system. YOLACT is a one-stage instance segmentation algorithm proposed by the research team of the University of California Fornia [20]. Compared with two-stage instance segmentation algorithms such as FCIS [21] and Mask R-CNN [23], YOLACT aims to add a mask branch to the one-stage model to achieve the purpose of instance segmentation. YOLACT completes the task of instance segmentation by adding two parallel branches: The first branch uses Fully Convolutional Networks (FCN) to generate a series of prototype masks independent of a single instance. The second branch uses an additional head in the detection branch to predict the mask coefficients for the representation of the coding example in the circular mask space. Finally, after using Non-Maximum Suppression (NMS) for each instance, the final prediction results of object information and mask image are obtained by linearly combining the output results of the two branches. This method not only preserves the spatial correlation, but also maintains the structure of the one-stage model, so the operation speed in the object recognition will be faster than two-stage instance segmentation algorithms.

TABLE 1. Comparison of some instance segmentation algorithms.

Algorithm	Backbone	fps	mAP
YOLACT [19]	ResNet-101	33~45	31.3
FCIS [20]	ResNet-101	6~7	31.0
PA-Net [21]	ResNet-101	4~5	38.1
Mask R-CNN [22]	ResNet-101	8~9	38.1
Mask Scoring R-CNN [23]	ResNet-101	8~9	40.4

YOLACT's recognition operation speed can reach more than 33 fps. However, as shown in Table 1, under the same COCO Dataset [30] and the same backbone, the mean of Average Precision (mAP) obtained by YOLACT is lower than other instance segmentation algorithms. But for the moving object prediction and grasping system proposed in this paper, this accuracy is already within the acceptable range. Therefore, in consideration of real-time object recognition, we chose YOLACT to implement the object detection of this system.

B. DATA AUGMENTATION

Before using YOLACT for object recognition, we must first perform network training. Since the training data of the instance segmentation algorithm all needs to label the contour of the object, and the quantity of training data and the accuracy of the contour of the labeled object will affect the mask effect obtained by YOLACT. In the label processing of training data, if manual labeling is used, it will require a lot of manpower and time. Therefore, we adopt the method of data augmentation to generate the training data. In the preparation of training data, because the LabelMe tool allows the user to determine the points to be labeled, and the generated JSON file after the labeling is also very easy to provide for the use of subsequent data augmentation. Therefore, we used the image generated by LabelMe tool to augment the training data. We used two object moving platforms as the background and used the geometric transformation methods of the image processing to rotate, zoom, and shift the object data to synthesize and augment the training data. While retaining the original feature of the object, various training files can be randomly synthesized. In addition to processing the image, the same geometric transformation is performed on the points

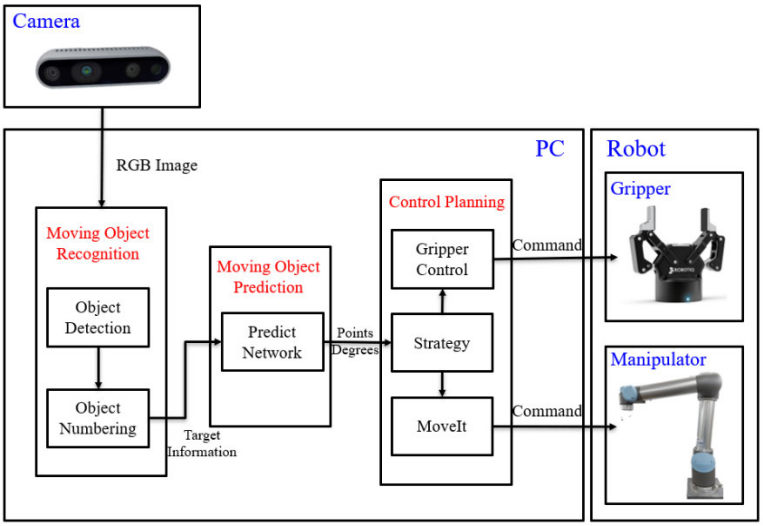

FIGURE 1. Architecture diagram of the proposed moving object prediction and grasping system.

in the JSON file generated by the LabelMe tool to complete the augmentation of the training data.

C. OBJECT NUMBERING

We used the match method to design an object numbering system. In the results of object recognition, because the recognition results between each frame and each frame are independent, and the object information output of YOLACT recognition is stored in an array. If the user wants to use the object information, the user must read it from the output array, but while reading the information, errors such as Table 2 may occur. When three consecutive images are recognized, the order of the object information stored in the YOLACT output array is different. When the user wants to use the object information of the Bottle 1, the first position stored in the output array must be used. This will use the object information of Bottle 2 and cause the wrong object information to be used for prediction. In addition, this sudden error of numbering information may occasionally occur on different objects. Therefore, we design an object numbering system to avoid the inconsistent order of the array information output by YOLACT.

First, the acquired object information must be stored. We store the information in a list to facilitate matching. With the object data list from the previous moment, the user can match the information currently obtained. The relevance between each other is obtained by matching, and the processing flow of object matching is shown in Fig. 2. In order to ensure the relevance between consecutive images, and to ensure that the information is not affected by the reading order of the recognition results. We will assign a number ID to the read object, and this ID will remain until the object disappears on the screen. Matching object is using the center point of the object for comparison. When the current information is obtained at this moment, the center point of all the objects at the previous moment will be subtracted. Find the minimum value of the target object at the moment and the center point of all objects at the previous moment. With this distance value, it will determine whether it is the same object name based on the object information of the smallest distance at the previous moment. If it is not the same object name, it will be determined as new object information, and a new ID will be given for storage. If it is the same object name, a new round of determination will be made to ensure that it is not affected by the same category of object. In this paper, the new round of determination is set to the moving distance between each frame to be less than 20 pixels, so that the system can recognize the object, and use these determinations to keep the information of the object in the continuous image before and after the sequence can be consistent.

TABLE 2. Output situation of continuously recognized objects.

Object	Current frame object information	Previous frame object information	Object information in the first two frames
Bottle1	Array order = 1, Point = (40,30)	Array order = 2, Point = (35,30)	Array order = 1, Point = (30,30)
Bottle2	Array order = 2, Point = (20,20)	Array order = 1, Point = (15,20)	Array order = 2, Point = (10,20)
Block	Array order = 3, Point =(70,70)	Array order = 3, Point = (65,70)	Array order = 3, Point = (60,70)
Tetra Pak	Array order = 4, Point = (95,10)	Array order = 4, Point = (90,10)	Array order = 4, Point = (85,10)

IV. MOVING OBJECT PREDICTION

When the system obtains the object numbering information, we design two moving object prediction networks to predict the grasping position and the grasping angle of the moving object. It is mainly divided into three parts: (a) LSTM-based moving position prediction network, (b) CNN-based grasping

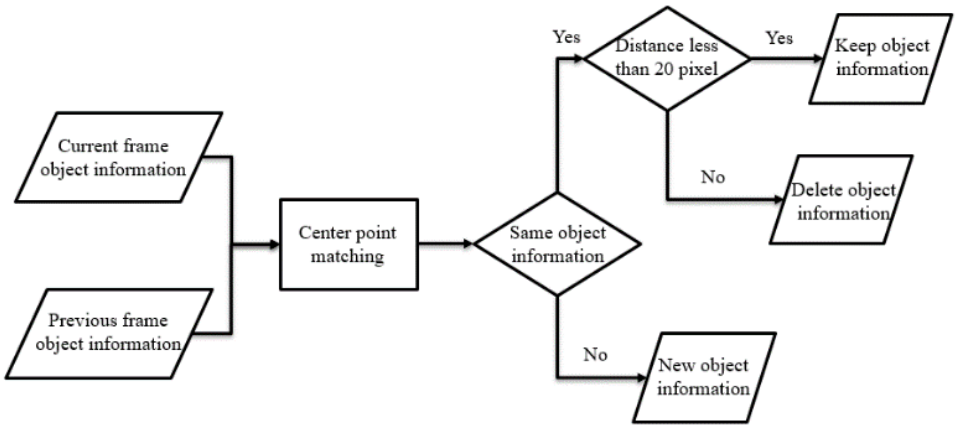

FIGURE 2. Flow chart of the object matching processing.

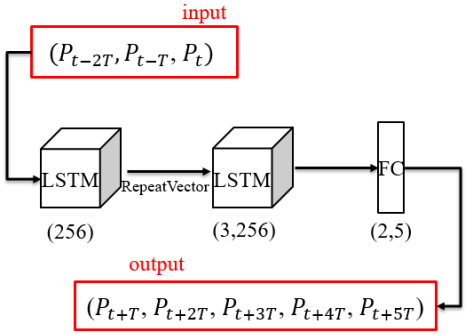

FIGURE 3. Architecture diagram of the proposed LSTM-based moving position prediction network.

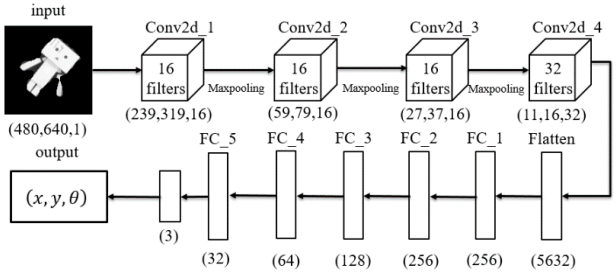

FIGURE 4. Architecture diagram of the proposed CNN-based grasping point prediction network.

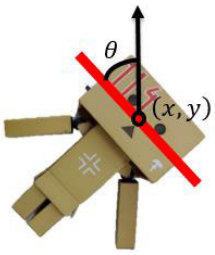

FIGURE 5. Schematic diagram of the grasping point.

point prediction network, and (c) moving object prediction network. They are described as follows:

A. LSTM-BASED MOVING POSITION PREDICTION NETWORK

We used Long Short-Term Memory (LSTM) to design a moving position prediction network as shown in Fig. 3. The network is composed of two LSTM layers and a fully connected layer. Through the learning of the network, the prediction network can analyze the three past positions of the object and obtain the future five grasping positions of the moving object. This network uses the center point of the bounding box Pt of the moving object obtained by YOLACT and the previous center point of the two bounding boxes ($Pt - 2T$, $Pt - T$) as inputs to predict the future position of the moving object. The outputs are the center points of the future five bounding boxes of the moving object ($Pt + T$, $Pt + 2T$, $Pt + 3T$, $Pt + 4T$, $Pt + 5T$), where P is the center point coordinate of the bounding box (x, y), t is the current time, and T is the time interval. We take every 20 frames (about 0.5 seconds) as an interval, so the position after 100 frames (about 2.5 seconds) can be predicted. This time interval can also be adjusted according to the needs of the system.

B. CNN-BASED GRASPING POINT PREDICTION NETWORK

We used Convolutional Neural Network (CNN) to design a grasping point prediction network as shown in Fig. 4, which is used to predict an appropriate grasping point of the moving object in the future. The diagram of the grasping point used in this paper is shown in Fig. 5, where x and y are the coordinate values of the object's suitable grasping point projected to the x-axis and y-axis of the camera coordinate system, θ is the angle formed by the angle of the grasping direction of the gripper and the vertical line. This network uses the object mask image obtained by YOLACT as the input. The outputs are the grasping position (x, y) and the grasping angle θ of the contour of the appropriate grasping point of the moving object. There are many ways to define a grasping point [31]–[33]. For example, a method of simplifying the contour of the picture was proposed

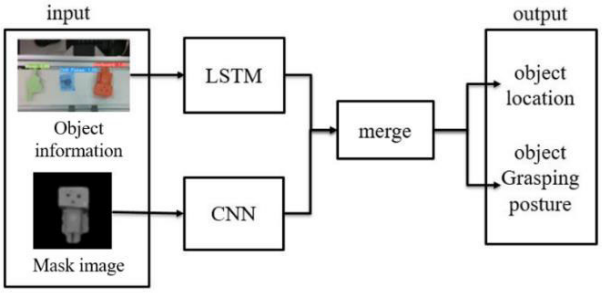

FIGURE 6. Simple schematic diagram of the proposed prediction network 1.

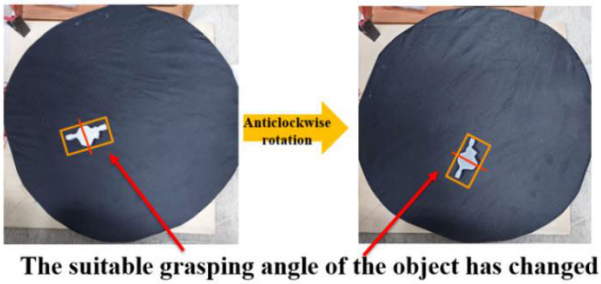

FIGURE 7. Schematic diagram of the change in the grasping angle of a moving object on a circular rotating platform.

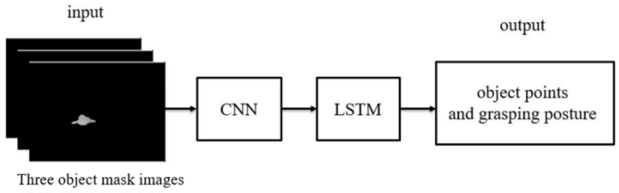

FIGURE 8. Simple schematic diagram of the proposed prediction network 2.

to improve the determination of the grasping position of unknown object [34], and GQ-CNN [35] was used to verify its effect. The definition method of grasping point used in this paper also uses the contour of the object to determine a suitable grasping point. The difference between the two methods is that the method described in [34] used a depth map to determine, and we used a mask image to determine.

C. MOVING OBJECT PREDICTION NETWORK

In predicting the grasping position and grasping angle of a moving target object, we combine LSTM-based moving position prediction network and CNN-based grasping point prediction network to design two moving object prediction networks, Prediction Network 1 and Prediction Network 2. They are described as follows:

The architecture of Prediction Network 1 is shown in Fig. 6, which is a network architecture that combines LSTM-based moving position prediction network and CNN-based grasping point prediction network in parallel. The input of Prediction Network 1 has two parts. The first part is to input the center point coordinate of the bounding box (x, y) of the object obtained by YOLACT into LSTM-based moving position prediction network. The second part is to input the object mask image obtained by YOLACT into CNN-based grasping point prediction network. The outputs of Prediction Network 1 are five center point positions of the bounding box of the moving object and one suitable grasping angle.

The advantage of the architecture of Prediction Network 1 is that LSTM and CNN networks can be trained at the same time. However, the CNN-based grasping point prediction network in Prediction Network 1 cannot perform time series analysis, so it cannot predict a correct grasping angle when the posture of the object changes. As shown in Fig. 7, if the posture of the moving object changes, Prediction Network 1 cannot correctly predict the grasping angle.

In order to solve the problem that Prediction Network 1 cannot correctly predict the grasping angle of a moving object when its posture changes, we propose Prediction Network 2, as shown in Fig. 8. This architecture is a network architecture that combines CNN-based grasping point prediction network and LSTM-based moving position prediction network in series. In addition, the TimeDistributed layer [36] is used to enable Prediction Network 2 to simultaneously predict the grasping position and the grasping angle of the suitable grasping point in the future. Since the first dimension of the TimeDistributed layer is time, we set the first dimension to 3. According to input 3 consecutive mask images to make the CNN-based grasping point prediction network predict the grasping position and the grasping angle, it can also make the LSTM network analyze its time relationship to predict the future suitable grasping position and grasping angle.

The inputs of Prediction Network 2 are three continuous mask images of the target object obtained by YOLACT, and the outputs are the name of the object and future five grasping positions $(x_1, y_1), (x_2, y_2), \cdots, (x_5, y_5)$ and grasping angles $(\theta_{sin1}, \theta_{cos1}), (\theta_{sin2}, \theta_{cos2}), \cdots, (\theta_{sin5}, \theta_{cos5})$. In terms of the angle θ of the grasping point, this paper only includes a semi-circular interval from $-90°$ to $90°$ and a total of $180°$ when marking the grasping angle, but the angle is actually a circular interval of $-180°$ to $180°$ and a total of $360°$. In order to make the two ranges of values are the same, double-angle formulas is used for training. For example, θ_1 is the angle of object 1 as defined in Fig. 5, then $\theta_{\cos 1} = \cos(2\theta_1), \theta_{\sin 1} = \sin(2\theta_1)$. This method can also make the training effect more in line with the actual angle. Its network architecture is shown in Fig. 9. We used the TimeDistributed layer to make the CNN-based grasping point prediction network have a time series relationship. The TimeDistributed layer is a layer wrapper, which can be used on any layer of the network, such as a convolution layer, a pooling layer, or a fully connected layer. In the same TimeDistributed layer, weight information can be shared with each other, so the input of the network architecture can be related to each other before and after. As shown in Fig. 10, it changed the original CNN network

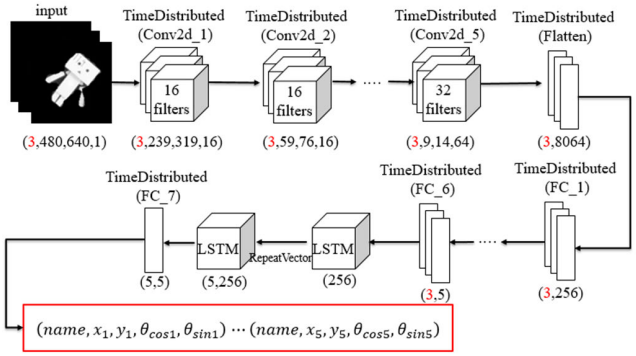

FIGURE 9. Architecture diagram of prediction network 2.

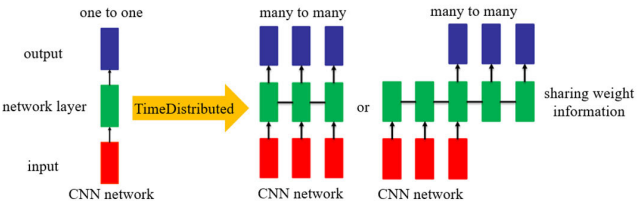

FIGURE 10. Description of time distributed layer used for CNN.

architecture from a single-input single-output method to a multi-input multi-output method. Therefore, the CNN-based grasping point prediction network can also have the same ability as the LSTM network to analyze time series.

Compared with Prediction Network 1, Prediction Network 2 only needs one type of object information as input, and its output is also changed from the center point of the bounding box of the future movement of objects in Prediction Network 1 to a suitable grasping point of the object in the future. The grasping angle is also a suitable grasping angle to grasp the moving object. Therefore, Prediction Network 2 is more suitable for the prediction of moving objects when the posture of the moving object changes, and the predicted grasping position is also more suitable for the position corresponding to the shape of the object.

V. SYSTEM REALIZATION BASED ON ROS

We used Robot Operating System (ROS), which can make system integration easier to implement and expand, to implement the proposed moving object prediction and grasping system. As shown in Fig. 11, it integrates all the programs of the camera image extraction, the strategy processing, and the robot manipulator and gripper control. A total of six nodes (RealSense, YOLACT, Strategy, MoveIt, UR5, and ROBOTIQ) as well as two topics (Topic 1, Topic 2) and three services (Service 1, Service 2, Service 3) are designed. They are described as follows:

In the part of the camera, we used RealSense as the image sensor of the system. Because the "YOLACT" only needs the RGB image information of the camera, and the "Strategy" only needs the target information processed by "YOLACT". These two kinds of messages only need one-way communication, so two topics named Topic 1 and Topic 2 are designed for this requirement. The node names of the publisher and subscriber of Topic 1 are "RealSense" and "YOLACT". The node names of the publisher and subscriber of Topic 2 are "YOLACT" and "Strategy". The message that the publisher "RealSense" will publish is the RGB image detected by the camera with a resolution of 640∗480. The message that the publisher "YOLACT" will publish is the future five grasping positions $(x_1, y_1), (x_2, y_2), \cdots, (x_5, y_5)$ and grasping angles $(\theta_{sin1}, \theta_{cos1}), (\theta_{sin2}, \theta_{cos2}), \cdots, (\theta_{sin5}, \theta_{cos5})$. Two subscribers will respectively subscribe to the messages on these topics according to the processing needs.

In the part of the robot manipulator, we used the robot manipulator named UR5 as the execution and operation equipment of the system. Because the strategy of the system needs to obtain information such as the path of the trajectory planning from MoveIt, and MoveIt needs to obtain the actual joint moving trajectory of the robot manipulator (UR5). This way can control the robot manipulator effectively and monitor the movement path of the robot manipulator. The communication between "Strategy" and "MoveIt" and between "MoveIt" and "UR5" requires two-way communication. Therefore, we plan two services named Service 1 and Service 2 in this section. The server and client nodes of Service 1 are "Strategy" and "MoveIt". The request sent by "Strategy" that plays the role of the server is the position (x, y, z) of the robot manipulator end point and the quaternion (w, x, y, z) of the posture. After the "MoveIt" that plays the role of the client receives the sent information, the response given is the result of forward and inverse kinematics and the planned moving path. The node names of the server and client of Service 2 are "MoveIt" and "UR5". The request sent by "MoveIt" that plays the role of the server is the joint trajectory of the robotic manipulator. After the "UR5" that plays the role of the client receives the response, the response given is the state value of 0 (false) or 1 (true) to indicate whether the robot manipulator is currently busy. In addition, we used the MoveIt suite provided by ROS to achieve the motion planning of the robot manipulator. The advantage of MoveIt is that it is very friendly to beginners. Users can easily use MoveIt to complete robot operations without the concept of robot operation or kinematics. Another advantage is that MoveIt make users to easily replace kinematics, trajectory planning, and collision detection modules based on the modules which they want. The robot manipulator used in this paper is UR5, and the MoveIt package for the built-in kinematics solver of UR5 is KDL of OrocosL, but the solution speed of this method is very slow, and it is easy to get different joint values at the same target position. The solution speed of Trac_ik method is very fast, and the value obtained is relatively stable, it is not easy to find different solutions. Therefore, we choose the Trac_ik method to solve the forward and inverse kinematics of the robot manipulator.

In the part of the gripper, we used the ROBOTIQ 2F-85 gripper as the end effector of the robot manipulator. Because the strategy of the system needs to obtain the information of

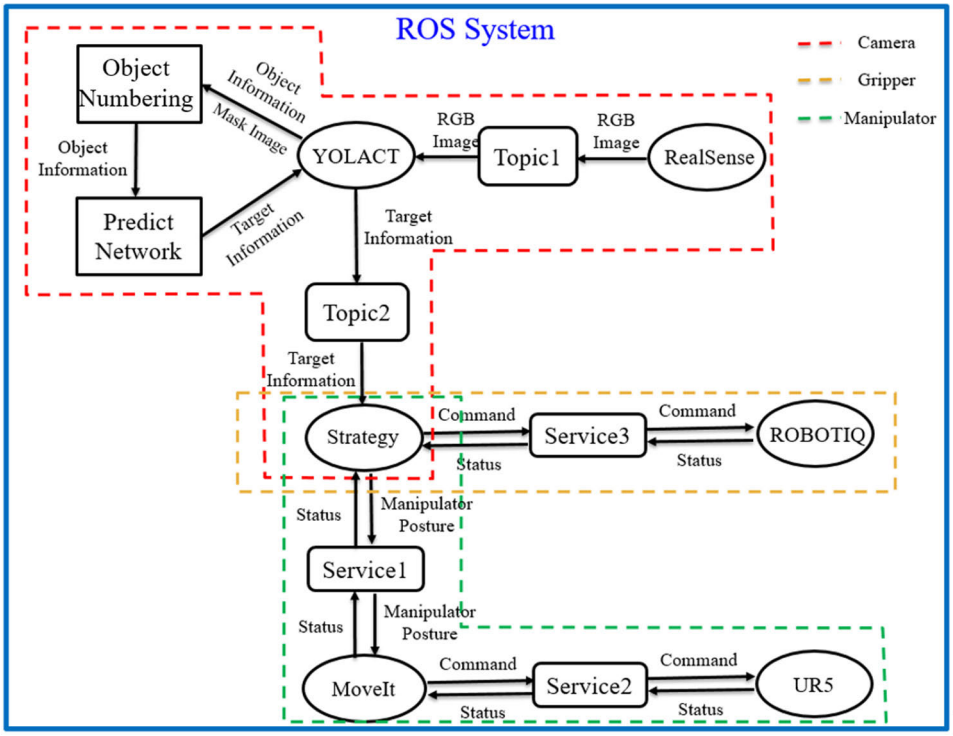

FIGURE 11. System architecture of the implemented moving object prediction and grasping system based on the robot operating system.

the grasping state of the gripper, it can control the gripper to grasp the object. The communication method between the strategy and the gripper requires a two-way communication method, so we plan a service named Service 3. The node name of the server is ''Strategy'', and the node name of the client is ''ROBOTIQ''. The request sent by the server ''Strategy'' is the state value of 0 (close) or 1 (open) for grasping or releasing. After the client ''ROBOTIQ'' receives the response, the response given is the status value of 0 (false) or 1 (true) to indicate whether the gripper is currently busy.

In summary, Table 3 is the role and message of 2 topics, 3 services, and 6 nodes (programs) using the Robot Operating System to integrate camera, robot manipulator, and gripper designed for the proposed system.

VI. EXPERIMENTAL RESULTS

We used two kinds of object moving platforms made by the laboratory to verify the effectiveness of the proposed system. As shown in Fig. 12, they are a conveyor with a length × width of 120 cm × 40 cm and a circular rotating platform with a radius of 26 cm. In this paper, the used YOLACT and the proposed moving object prediction networks are deep neural networks. Thus they need training data for the network training. In the preparation of training data, an automatic data generation method based on the LabelMe tool is proposed to collect a large amount of training data. The difference between manual annotation and the method with data augmentation is: manual labeling needs to label the contour of the object, so it will waste a lot of time on labeling the object. Moreover, when the background is different, it needs to be labelled again. The

TABLE 3. Node definition and its description of the implemented ROS-based moving object prediction and grasping system.

Topic/ Service	Node Name	ROS Role	Message/ Request/ Response
Topic 1	RealSense	Publisher	RGB Image: 640*480
	YOLACT	Subscriber	
Topic 2	YOLACT	Publisher	Target Information: $(x_i, y_i, \theta_{sini}, \theta_{cosi})$, i=1,2,…,5
	Strategy	Subscriber	
Service 1	Strategy	Sever	Command(Request) Pose of Manipulator: Point (x, y, z) and Quaternion(w, x, y, z)
	MoveIt	Client	Status (response) Control Command of Manipulator: Forward and Inverse Kinematics, Path
Service 2	MoveIt	Sever	Command (Request) Trajectory planning of manipulator: Joint Trajectory Action
	UR5	Client	Status (Response) Busy Status of Manipulator: 0 (False) or 1 (True)
Service 3	Strategy	Sever	Command (Request) Control Command of Gripper: 0 (close) or 1 (open)
	ROBOTIQ	Client	Status (Response) Busy Status of Gripper: 0 (False) or 1 (True)

(a) (b)

FIGURE 12. Two laboratory-made object moving platforms: (a) a conveyor and (b) a circular rotating platform.

(a) (b)

FIGURE 13. Types of training data for network training: (a) YOLACT and (b) proposed moving object prediction network.

method with data augmentation can change the background at any time without re-labeling, and can freely adjust the position and angle of the object to obtain new training data. As shown in Fig. 13 (a), the input of YOLACT is a single RGB image and the type of training data is a picture with the contour of the labeled object. Since each picture needs to be marked with the contour of the object, manual labeling will take a lot of time. On the other hand, as shown in Fig. 13 (b), the inputs of the proposed moving object prediction network are three consecutive mask images, and the type of training data is in the form of a mask image with a grasping point and an angle of the labeled object. Since only the grasping point and angle of the object need to be labeled, manual labeling will not take a lot of time, and it is easy to obtain training data by the data augmentation. Data used to train YOLOCAT and the proposed moving target prediction network in the data augmentation are respectively shown in Table 4 and Table 5, where the number of backgrounds is two, such as the conveyor and the circular rotating platform. We found that this method with data augmentation not only saves a lot of manpower and time in preparing training data for the network training, but also can augment the required data at any time according to the actual needs of the system. The training time for YOLACT using 5,000 images and the prediction network using 200,000 images is about 18 hours and 12 hours, respectively.

The experimental results are mainly divided into two parts: (a) comparison of the two proposed moving object prediction networks, and (b) prediction results of Prediction Network 2. They are described as follows.

TABLE 4. Data used to train YOLOCAT in data augmentation.

Item	Number and Time
Training data type	One RGB image photo
Number of objects	4
Number of backgrounds	2 (conveyor and circular rotating platform)
Manually annotate photos	100 sheets
Spend time (Manually annotate)	About 5 hours
Automatically generate photos	5,000 sheets
Spend time (Automatically generate)	About 2 minutes

TABLE 5. Data used to train the moving object prediction network in data augmentation.

Item	Number and Time
Training data type	Three consecutive mask pictures
Number of objects	4
Manually annotate photos	4 sheets
Spend time (Manually annotate)	About 1 minute
Automatically generate photos	200,000 sheets
Spend time (Automatically generate)	About 1 hour

A. COMPARISON OF THE TWO PROPOSED MOVING OBJECT PREDICTION NETWORKS

We used four experimental objects: (a) Bottle, (b) Metal Workpiece, (c) Danboard, and (d) Tetra Pak, as shown in Fig. 14. A comparison of the two moving object prediction networks of Prediction Network 1 and Prediction Network 2 is shown in Table 6. It can be seen from the table that although the input and output used by the two prediction networks are different, the predicted position error and angle error are all within the acceptable range for object grasping. During the movement, the mask image of the object obtained by the object detection method will have some influence due to environmental factors such as background reflection. Therefore, the future grasping position predicted by the Prediction Network 2 using the mask image of the object as the input will slightly deviate from the most suitable grasping position of the object at the current moment. As shown in Table 6, using both the center point of the bounding box of the object and the mask image as the inputs of Prediction Network 1, the predicted position error will be smaller. However, the position error and angle error of these two prediction networks will not have much influence on the object grasping. In addition, if the posture of the object changes during the movement, Prediction Network 1 cannot predict the correct grasping angle, but Prediction Network 2 can. For example, the posture of the moving object on the circular rotating platform will change during the movement. Therefore, we used Prediction Network 2 to carry out the prediction and grasping experiments of moving objects on the object moving platform of two different moving modes.

B. PREDICTION RESULTS OF PREDICTION NETWORK 2

The experimental environments for grasping moving objects set up on the conveyor and the circular rotating platform are shown in Fig. 15 and Fig. 16, respectively. A two-finger gripper is installed at the end-point of the robot manipulator, and a camera is set up above the gripper of the robot manipulator

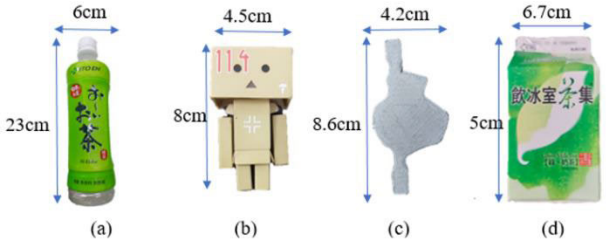

FIGURE 14. Four experimental objects: (a) bottle, (b) danboard, (c) metal workpiece, (d) tetra pak.

TABLE 6. Comparison of the two proposed moving object prediction networks.

Item	Prediction Network 1	Prediction Network 2
Input	Center point of the object bounding box & mask image	Mask image
Output	Object center point	Suitable grasping point
Conveyor x-axis average error	4.54 pixel (about 0.45 cm)	7.44 pixel (about 0.74 cm)
Conveyor y-axis average error	2.001 pixel (about 0.2 cm)	2.16 pixel (about 0.21 cm)
Conveyor angle average error	5.3 degree	5.22 degree

(eye-in-hand). In the experiment of the two object moving platforms, the initial state of the robot manipulator is located at a position of 61.5 cm from the camera directly above the platform. The actual process of the robot manipulator grasping the moving objects on the conveyor and circular rotating platform are shown in Fig. 17 and Fig. 18. The video of the proposed system for four different objects on a conveyor and a circular rotating platform can be viewed on this website: https://www.youtube.com/watch?v=XlocslHGi98. Taking the experiment of Bottle on the circular rotating platform as an example, the inputs of Prediction Network 2 for three consecutive mask images on the circular rotating platform are shown in Fig. 19. The outputs of the Prediction Network 2 are shown in Fig. 20, where the yellow dots are the grasping positions and the red line are the grasping angles. Fig. 20 (a) is the five grasping points and grasping angles of the network output, and Fig. 20 (b)∼(f) are five relationship diagrams between the predicted result and the actual state of this object.

The prediction results of these four objects on the conveyor and the circular rotating platform are respectively shown in Table 7 and Table 8. It can be seen that the proposed system can indeed correctly predict the moving trajectory of the object. The data analysis of the objects on the conveyor and the circular rotating platform are respectively shown in Table 9 and Table 10. It can be seen from the results that the proposed system can indeed make good predictions for four objects on two different object moving platforms. Since Prediction Network 2 only uses three mask images as inputs, we found that the color of the Metal Workpiece is greatly affected by the reflection of the ambient light

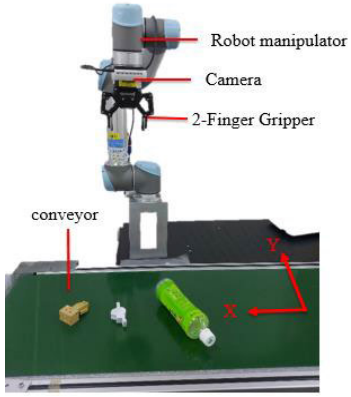

FIGURE 15. Experimental environment for the grasping of moving objects on the conveyor.

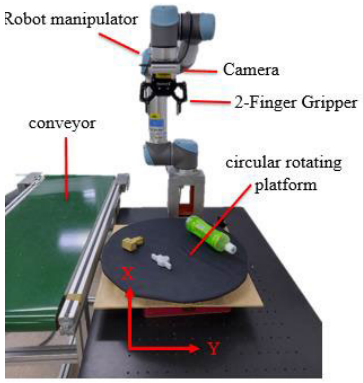

FIGURE 16. Experimental environment for the grasping of moving objects on the circular rotating platform.

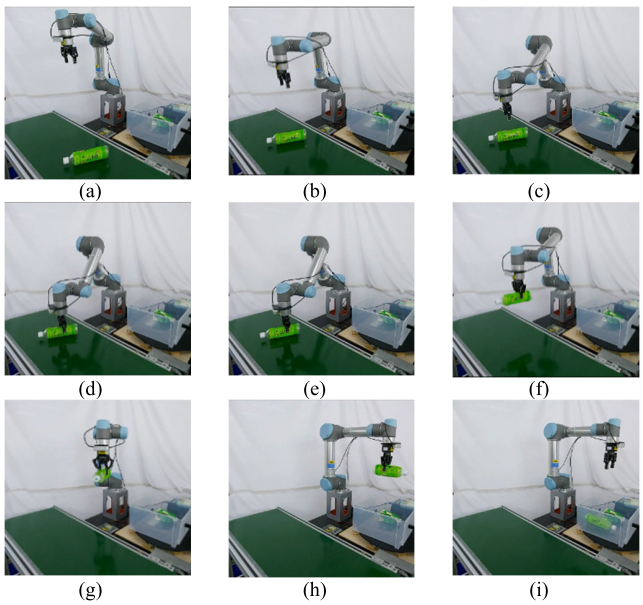

FIGURE 17. Experimental snapshots of the moving object grasping experiment on the conveyor.

source. Therefore, its shape damage of the mask image is more obvious than that of the other two objects, Danboard and Tetra Pak. Although its average error is relatively higher than the others, but these errors are within the acceptable

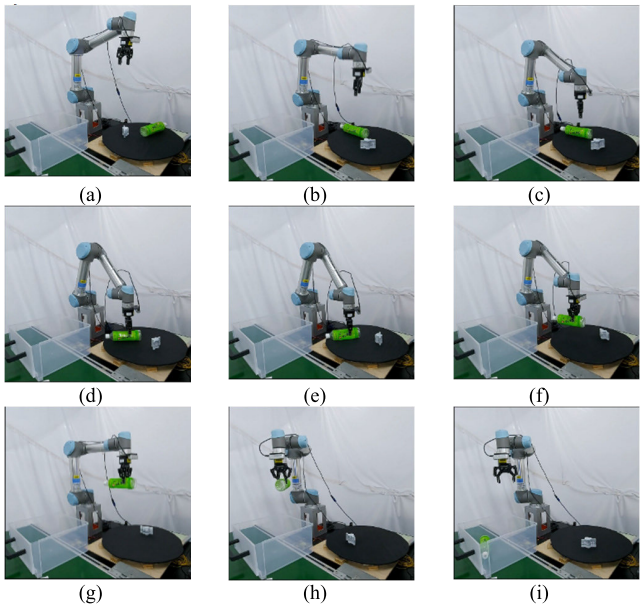

FIGURE 18. Experimental snapshots of the moving object grasping experiment on the circular rotating platform.

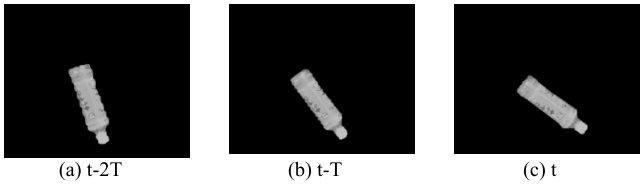

FIGURE 19. Inputs of prediction network 2 for the required three bottle mask images on the circular rotating platform.

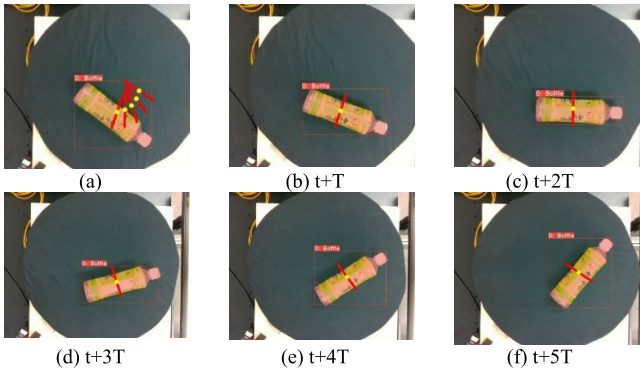

FIGURE 20. Outputs of prediction network 2 for the bottle on the circular rotating platform.

range to successfully grasp the object. Similarly, since the shape of the Bottle is cylindrical, it will sway slightly from side to side during the movement, so the obtained mask image will cause the failure. Therefore, the average error is relatively high, but the prediction results are mostly located on bottles. As shown in Table 11, we performed a total of 160 experiments, 80 on a conveyor and 80 on a circular rotating platform. That is, each of the four different objects was tested 20 times on a conveyor and a circular rotating platform. In these experiments, only 6 failures, and the overall success rate was 96.25%. These quantitative results can

TABLE 7. Experimental results of prediction network 2 for four different objects on the conveyor.

Object	Bottle	Danboard	Metal Workpiece	Tetra Pak
Predict Result				
Predict Point1				
Predict Point 2				
Predict Point 3				
Predict Point 4				
Predict Point 5				

TABLE 8. Experimental results of prediction network 2 for four different objects on the circular rotating platform.

Object	Bottle	Danboard	Metal Workpiece	Tetra Pak
Predict Result				
Predict Point1				
Predict Point 2				
Predict Point 3				
Predict Point 4				
Predict Point 5				

verify the effectiveness of the proposed system. From these 6 unsuccessful experiments, we found that the reasons for these failures are related to the light source of the environment and the shape of the objects. For example, objects such as

TABLE 9. Average prediction errors of prediction network 2 for each of the four different objects on the conveyor (50 experiments per object).

Object	Bottle	Danboard	Metal Workpiece	Tetra Pak
x-axis average error	8.07 pixel (~0.8 cm)	6.01 pixel (~0.86 cm)	8.6 pixel (~0.86 cm)	7.11 pixel (~0.71 cm)
y-axis average error	2.11 pixel (~0.2 cm)	2.02 pixel (~0.2 cm)	2.56 pixel (~0.25 cm)	1.98 pixel (~0.19 cm)
angle average error	5.3 degree	4.5 degree	6.3 degree	4.8 degree

TABLE 10. Average prediction errors of prediction network 2 for four different objects on the circular rotating platform (50 experiments per object).

Object	Bottle	Danboard	Metal Workpiece	Tetra Pak
x-axis average error	8.09 pixel (~0.8 cm)	7.54 pixel (~0.75 cm)	8.5 pixel (~0.85 cm)	7.473 pixel (~0.74 cm)
y-axis average error	9.7 Pixel (~0.97 cm)	6.5 pixel (~0.65 cm)	8.23 pixel (~0.82 cm)	8.32 pixel (~0.83 cm)
angle average error	8.3 degree	4.2 degree	5.5 degree	5.3 degree

TABLE 11. Object grasping success rate of prediction network 2 for four different objects on the conveyor and the circular rotating platform.

Object	Bottle	Danboard	Metal Workpiece	Tetra Pak
Object grasping success rate (20 experiments on the conveyor)	90%	100%	90%	100%
Object grasping success rate (20 experiments on the circular rotating platform)	95%	100%	95%	100%

Bottle and Metal Workpiece are more susceptible to the influence of the light source of the environment. In addition to the influence of the light source of the environment, the elongated shape of the bottle will shake slightly due to the movement of the object, which is one of the reasons why grasping object was not successful. From these actual moving object grasping experiments, these results indicate that the proposed system indeed let the robot manipulator grasp the target object successfully. We did not deliberately adjust the light source of the experimental environment. If we pay attention to the arrangement and adjustment of the light source, the object grasping success rate of the proposed system can be further improved.

VII. CONCLUSION

Conveyors and circular rotating platforms are two object moving platforms often used in production lines. We proposed a practical solution to predict and grasp moving objects on these two object moving platforms so that a robot manipulator can successfully and effectively grasp moving objects using a two-finger gripper. There are six main points in this paper: (a) A two-stage instance segmentation algorithm named YOLACT is used to implement the moving object detection of the proposed system to recognize multiple objects in the RGB image. (b) In the preparation of training data, an automatic data generation method based on the LabelMe tool is proposed to reduce the manpower and time required to collect a large amount of training data. (c) The proposed prediction network can simultaneously predict the future grasping position and grasping angle of multiple moving objects in the image at one time. (d) Since the proposed prediction network can simultaneously predict multiple moving objects in the image at one time, an object numbering system is proposed to ensure that the order of these consecutively recognized moving objects is consistent. (e) ROS is used to integrate all programs to implement the proposed system, so that the robot manipulator can successfully grasp objects not only on the conveyor, but also on the circular rotating platform. (f) We fabricated a conveyor and a circular rotating platform, and performed some practical experiments on these two object moving platforms using four different objects to verify the usability of the proposed system. The contributions of this paper can be summarized as follows: (i) In the design of object numbering system for the moving object recognition, when multiple objects are moving on the object moving platform, it is important for the system to accurately track the target object to be grasped. Therefore, an object numbering system based on object matching is proposed to ensure that the order of these recognized moving objects is consistent and the system can correctly track the target object. (ii) In the design of moving object prediction system, the time series analysis capabilities of Long Short-Term Memory (LSTM) and the image recognition capabilities of Convolutional Neural Network (CNN) are combined to design the proposed moving object prediction system. First, a LSTM-based moving position prediction network and a CNN-based grasping point prediction network are designed respectively. Then, the LSTM-based moving position prediction network and the CNN-based grasping point prediction network are respectively combined in parallel and in series to design Prediction Network 1 and Prediction Network 2. In addition, the TimeDistributed layer is used to make Prediction Network 2 have the ability to predict the future pose of the target object. Therefore, when the object moves on the circular rotating platform and the posture of this object changes, Prediction Network 2 can correctly predict the future grasping position and grasping angle of the moving object at the same time. (iii) In the system realization, we used ROS to efficiently integrate all programs of the image, strategy, and control for this system and clearly describe how to develop the topic and service functions required by the system for hardware devices of camera, robot manipulator, and two-finger gripper. In addition, we used the ROS suite named MoveIt to find the solution of forward and inverse kinematics of the robot manipulator, and to do the trajectory planning of the robot manipulator, so that the robot manipulator can effectively grasp the object. There are two main limitations of the proposed system: (a) The proposed system cannot

grasp unknown objects. Because the used object recognition method (YOLACT) can only recognize trained objects, the proposed system can only grasp trained objects. (b) The proposed system cannot predict moving objects without regular moving paths. Both object moving paths considered in this research are regular moving path, so a random moving path is out of the scope of this research. Thus, these two limitations of the proposed system do not affect the problem that this paper intends to solve. Related applications without these two limitations can be further investigated in the future.

REFERENCES

[1] J. Liang, J. Zhang, B. Pan, S. Xu, G. Zhao, G. Yu, and X. Zhang, "Visual reconstruction and localization-based robust robotic 6-DoF grasping in the wild," *IEEE Access*, vol. 9, pp. 72451–72464, 2021.

[2] Y. Zhang, L. Li, M. Ripperger, J. Nicho, M. Veeraraghavan, and A. Fumagalli, "Gilbreth: A conveyor-belt based pick- and-sort industrial robotics application," in *Proc. 2nd IEEE Int. Conf. Robot. Comput. (IRC)*, Jan. 2018, pp. 17–24.

[3] D. Stogl, D. Zumkeller, S. E. Navarro, A. Heilig, and B. Hein, "Tracking, reconstruction and grasping of unknown rotationally symmetrical objects from a conveyor belt," in *Proc. 22nd IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2017, pp. 1–8.

[4] F. Islam, O. Salzman, A. Agarwal, and M. Likhachev, "Provably constant-time planning and replanning for real-time grasping objects off a conveyor belt," 2020, *arXiv:2003.08517*.

[5] I. Akinola, J. Xu, S. Song, and P. K. Allen, "Dynamic grasping with reachability and motion awareness," 2021, *arXiv:2103.10562*.

[6] P. K. Allen, A. Timcenko, B. Yoshimi, and P. Michelman, "Automated tracking and grasping of a moving object with a robotic hand-eye system," *IEEE Trans. Robot. Autom.*, vol. 9, no. 2, pp. 152–165, Apr. 1993.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[8] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[12] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 379–387.

[13] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 187–213.

[14] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[15] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[16] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," 2015, *arXiv:1511.00561*.

[17] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," 2016, *arXiv:1606.00915*.

[18] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-time instance segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9157–9166.

[19] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2359–2367.

[20] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.

[21] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Conf. Comput. Vis.*, Oct. 2017, pp. 2980–2988.

[22] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring R-CNN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6409–6418.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] S. Kumra, S. Joshi, and F. Sahin, "Antipodal robotic grasping using generative residual convolutional neural network," 2019, *arXiv:1909.04810*.

[25] S. Hernandez-Mendez, C. Maldonado-Mendez, A. Marin-Hernandez, H. V. Rios-Figueroa, H. Vazquez-Leal, and E. R. Palacios-Hernandez, "Design and implementation of a robotic arm using ROS and MoveIt!" in *Proc. IEEE Int. Autumn Meeting Power, Electron. Comput. (ROPEC)*, Nov. 2017, pp. 1–6.

[26] T. Wang, Y. Zhao, L. Zhu, G. Liu, Z. Ma, and J. Zheng, "Design of robot system for radioactive source detection based on ROS," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2020, pp. 1397–1400.

[27] C.-C. Wong, S.-Y. Chien, H.-M. Feng, and H. Aoyama, "Motion planning for dual-arm robot based on soft actor-critic," *IEEE Access*, vol. 9, pp. 26871–26885, 2021.

[28] C.-C. Wong, L.-Y. Yeh, C.-C. Liu, C.-Y. Tsai, and H. Aoyama, "Manipulation planning for object re-orientation based on semantic segmentation keypoint detection," *Sensors*, vol. 21, no. 7, p. 2280, Mar. 2021.

[29] S. Chitta, I. Sucan, and S. Cousins, "MoveIt! [ROS topics]," *IEEE Robot. Autom. Mag.*, vol. 19, no. 1, pp. 18–19, Mar. 2012.

[30] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[31] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 1316–1322.

[32] Y. Xu, L. Wang, A. Yang, and L. Chen, "GraspCNN: Real-time grasp detection using a new oriented diameter circle representation," *IEEE Access*, vol. 7, pp. 159322–159331, 2019.

[33] D. Morrison, P. Corke, and J. Leitner, "Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach," 2018, *arXiv:1804.05172*.

[34] T. Mu, B. Yuan, H. Yu, and Y. Kang, "A robotic grasping algorithm based on simplified image and deep convolutional neural network," in *Proc. IEEE 4th Inf. Technol. Mechatronics Eng. Conf. (ITOEC)*, Dec. 2018, pp. 849–855.

[35] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-Net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," 2017, *arXiv:1703.09312*.

[36] *Keras Documentation: TimeDistributed Layer*. Accessed: Nov. 24, 2020. [Online]. Available: https://keras.io/api/layers/recurrent_layers/time_distributed/

CHING-CHANG WONG received the B.S. degree from the Department of Electronic Engineering, Tamkang University (TKU), Taiwan, in 1984, and the M.S. and Ph.D. degrees from the Department of Electrical Engineering, Tatung Institute of Technology, Taiwan, in 1986 and 1989, respectively. In 1989, he joined the Department of Electrical and Computer Engineering, TKU, where he served as the Department Chairperson, from 2006 to 2010. In 2007, he established the Robotics Engineering Institute. In 2011, he established the Intelligent Automation and Robotics Center. He is currently a Distinguished Professor. He has published and coauthored over 300 technical articles and 20 patents. His current research interests include intelligent control, humanoid robot, mobile robot manipulator, and deep reinforcement learning for robotic applications. He was elevated as a fellow of the Institution of Engineering and Technology (IET) in 2009, the Chinese Automatic Control Society (CACS) in 2015, and the Robotics Society of Taiwan (RST) in 2019. He was a recipient of the Outstanding Automatic Control Award from CACS, Taiwan, in 2009; and the Outstanding Robotics Engineering Award from RST in 2018. He received the Outstanding and Premium Research Award from the Ministry of Science and Technology (MOST), Taiwan, from 2011 to 2021. From 2009 to 2010, he served as the Chair for the IEEE Robotics and Automation Taipei Chapter.

MING-YI CHIEN was born in Tamsui, Taiwan, in 1995. He received the B.S. and M.S. degrees from the Department of Electrical and Computer Engineering, Tamkang University (TKU), Taiwan, in 2018 and 2021, respectively. He participated in the JUSST Exchange Study Program of The University of Electro-Communications (UEC), Tokyo, Japan, from October 2019 to September 2020. His major research interests include robot manipulator, robotic applications, and machine learning.

REN-JIE CHEN was born in Taoyuan, Taiwan, in 1993. He received the B.S. and M.S. degrees from the Department of Electrical and Computer Engineering, Tamkang University (TKU), Taiwan, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree. He participated in the Short Exchange Student Program of The University of Electro-Communications (UEC), Tokyo, Japan, from November 2019 to December 2019. His major research interests include robot manipulator, robotic applications, reinforcement learning, and machine learning.

HISAYUKI AOYAMA was born in Japan, in 1958. He received the bachelor's degree in mechanical engineering for production, the master's degree in precision machinery, and the Doctoral degree in precision engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 1981, 1983, and 1988, respectively. He worked with the Research Laboratory of Precision Machinery and Electronics, Tokyo Institute of Technology, from 1983 to 1988. He became an Associate Professor with the Department of Precision Engineering, Shizuoka University; and an Associate Professor with the Department of Mechanical Engineering and Intelligent Systems, The University of Electro-Communications, in 1997. He was a Visiting Researcher with the College of Manufacturing, Cranfield Institute of Technology, U.K., from 1989 to 1990; the Department of ECE, North Carolina State University, USA, in 1996; and the King Mongkut's Institute of Technology Ladkrabang, Thailand, in 2017. In 2002, he became a Professor, the Director of the Micro Robotics and Mechatronics Group, and the Director of the Global Alliance Laboratory Project. He is interested in such micro/precision engineering, micro metrology, and its industrial and biomedical applications.

KAI-YI WONG received the B.S. and M.S. degrees from the Department of Electrical and Computer Engineering, Tamkang University (TKU), Taiwan, in 2015 and 2017, respectively, and the Ph.D. degree from the Department of Mechanical and Intelligent Systems Engineering, The University of Electro-Communications (UEC), Tokyo, Japan, in 2021. She is currently an Assistant Professor with the Department of Electrical Engineering, Chung Yuan Christian University (CYCU), Taoyuan City, Taiwan. Her research interests include robotics, intelligent systems, machine learning, and nonlinear systems control.