Autonomous Recognition Grasping Robot Based on Speech Input

Yaoyu Cheng, Lijie Sheng, Zishang Zhang, Chenbo Bao, Zhongtang Zhang

***Abstract*—This paper studies the task of grasping arbitrary objects from the known categories by free-form language instructions. We bring these disciplines together on this open challenge, which is essential to human-robot interaction. Critically, the key challenge lies in inferring the category of objects from linguistic instructions and accurately estimating the 6-DoF information of unseen objects from the known classes. In this paper, we propose a language-guided 6-DoF category-level object localization model to achieve robotic grasping by comprehending human intention. To this end, we propose a novel two-stage method. Particularly, the first stage grounds the target in the RGB image through language description of names and attributes of objects. The second stage extracts and segments point clouds from the cropped depth image and estimates the full 6-DoF object pose at category-level. Under such a manner, our approach can locate the specific object by following human instructions, and estimate the full 6-DoF pose of a category-known but unseen instance which is not utilized for training the model. In the experiment, we designed several world scenes in Webots virtual environment, and then input our demand for grasping objects through voice, and judge whether the robot arm can grasp smoothly. Finally, the experimental results show that the capture success rate of our project is close to 90%, and it only takes about 60 seconds on average.**

I. Introduction

Understanding natural language instructions is an essential skill for domestic robots, releasing humans from pre-defining a specific target for robot grasping by programming. This inspires the task of making robots understand human instructions. In this task, the robot demands to localize the target object by parsing the names, potential attributes, and spatial relations of objects from the language descriptions. Thus it is non-trivial to make robotic grasping by linguistic description, as this task requires mature techniques from Computer Vision (CV), Natural Language Processing (NLP), and robotics. In this paper, we bring these disciplines together on this open challenge, which is essential to human-robot interaction.

To summarize, in this paper, we propose a category-level 3D object localization model to grasp unseen instances via natural language description. Take an RGB-D image and a natural language description as input, our goal is to infer the 6-DoF pose of the most likely object that matches the description. Firstly, The speech input is processed by NLP method, and the characteristic labels of the object description are extracted. Then, the classification and definition of the objects placed in the big world are defined by computer vision. Second, this project will use YOLOv5 to recognize objects in RGB images and match the features extracted by speech features before to confirm the object capture. Thirdly, the RGB image and the depth image were cut, and the optimal captured 6D posture was calculated using GQCNN. Finally, inverse kinematics is used to control the manipulator to grasp. As shown in Fig. 1.

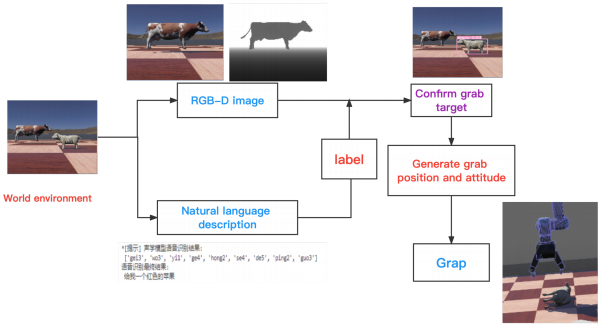


Fig. 1. Visualization results from variants of our systems in Webots.

The key challenge lies in inferring the category of objects from linguistic instructions, and accurately estimating the 6-DoF information of unseen objects from the known classes. Specifically, the vanilla object pose estimation approaches[14][4][15] attempt to estimate the poses of objects from the image, while we aim at locating specific objects using a natural language description. Here we employ convolutional neural networks (CNN) and connectionist temporal classification (CTC) to parse the linguistic instructions and generate features related to the features of the captured objects. Furthermore, the other thing that we focus on is, how do we match the object labels that we get with NLP to the actual objects that are laid out in the world. We decided to use the combination of YOLOv5 and computer vision to classify all objects in the world environment, that is, give them different labels that conform to objective basis, and then match this label with the voice label obtained through NLP, so as to obtain the actual information of the object we want to capture. Finally, this project plans to use GQCNN to confirm the 6D posture of the captured object based on the depth information of the confirmed object, so as to better ensure the flexibility of the captured object.

II. Related Work

A. Chinese Speech Recognition System

The commonly used Chinese speech recognition process is divided into four steps: feature extraction, matching the speech spectrogram and corresponding pinyin using acoustic models, and decoding the Chinese text corresponding to pinyin.

The most important part of this is feature extraction. Typically, feature extraction is given to human ear auditory

structures. As we all know, human speech originates from the initial sound produced by the vocal apparatus in the body. It is filtered by the shape of the vocal tract formed by other objects, including the tongue and teeth, to produce a wide variety of speech sounds. Traditional speech feature extraction algorithms are based on this. And with some digital signal processing algorithms, they are able to include the relevant features more accurately, thus helping the subsequent speech recognition process. Common speech feature extraction algorithms include MFCC, FBank, LogFBank, etc. The thematic flow of these programs is shown in Fig. 2.

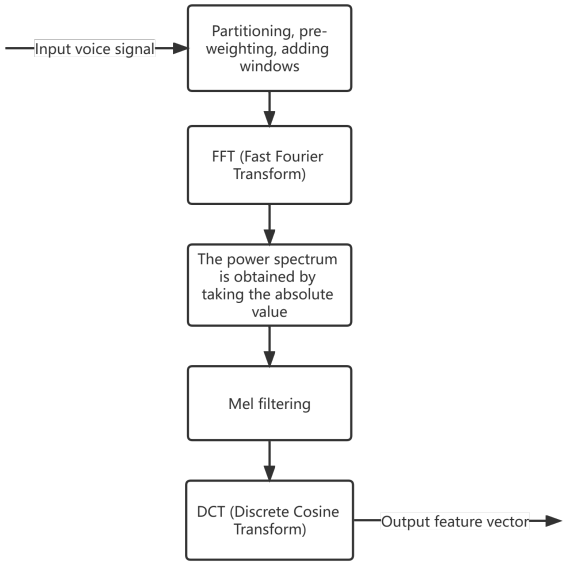


Fig. 2. Flow chart of common feature extraction algorithms.

To improve the speech recognition rate, it is necessary to overcome the varieties of diversity faced by the speech signal, including the diversity of speakers (the speaker itself, as well as the speakers), the diversity of the environment, etc. A convolutional neural network provides translational invariant convolution in time and space. And by applying the idea of convolutional neural network to acoustic modeling of speech recognition, the invariance of convolution can be used to overcome the diversity of speech signal itself. From this point of view, it can be considered that the entire time-frequency spectrum obtained from the analysis of the speech signal is treated like an image, and the deep convolutional network widely used in images is used for its recognition.

Based on these information, we try to implement Chinese speech recognition using convolutional neural network as the core network. In this project, the acoustic model is trained by using Convolutional Neural Network (CNN) and Connectivity Temporal Classification (CTC) methods to transcribe sounds into Chinese pinyin by using a large Chinese speech dataset, and to convert the pinyin sequences into Chinese text by language model. After finishing the speech recognition, we also need to do the keyword processing operation on the obtained speech, where the TextRank algorithm is used to help us extract the effective information in the article. Examples include the captured object, the color of the captured object, the shape of the captured object, etc.

B. Target Detection

The purpose of this part is to identify all objects in RGB images and make specific recognition according to the labels generated by NLP. Concerning the potential application of real-time target detection, we plan to use YOLO, a typical one-stage algorithm, to accomplish the above goals.

So far, YOLO algorithm has eight generations. The first generation[5] was proposed by the founder Joseph Redmon in 2015. At the third generation, he announced to stop supporting project development for YOLO. Due to the open source nature of the project, Alexey Bochkovskiy and others developed the fourth generation according to Darknet [9], which has better performance. However, the disadvantage is that the code is miscellaneous and the volume of generated results is too large. The fifth generation, developed by the American company Ultralytics LLC [16], greatly improved the performance and added some features like hyperparameter optimization. The latest generation of the network, namely YOLOv8, supports a full range of vision AI tasks. Due to the content of work and compatibility reasons, YOLOv5 is chosen here as the engine for target detection.

C. Grasp Planning.

Given an object and reachability constraints due to the environment, grasp planning considers finding a gripper configuration that maximizes a success (or quality) metric. Methods can be categorized based on success criteria into two types: analytic methods [12], which evaluate performance according to physical models such as the ability to resist external wrenches [11], and empirical (or data-driven) methods [3], which typically rely on human labels [2] or the ability to lift objects in physical trials [10]. More importantly, in this article, we tend to use Empirical approaches, which commonly utilize machine learning techniques to develop models that directly map robotic sensor readings to success labels derived from human evaluations or physical trials. Human labels have gained popularity due to their empirical correlation with physical success [2], despite the potential cost associated with acquiring them for large datasets.

III. Data

A. Chinese Speech Recognition System

Our speech recognition system follows a traditional end-to-end training approach, where we feed speech into the VGG learning network, which will output phoneme free phonetic sounds corresponding to the speech. Then we use the CTC method to combine these phonetic sounds and connect them into appropriate sentences. These generated sentences will be compared with the training set results for back-propagation to correct the VGG network parameters. So our core training set is the translation of a large number of Chinese speech

messages with their corresponding Chinese characters and pinyin.

In our search for a training set, we originally wanted to choose the thchs30 training set because it contains many dialects and essentially all syllables, but given its release date in 2001 and its small capacity of 30h recording time, it could not meet the training needs of an end-to-end model. We finally chose the aishell training set, which was released in 2017 and contains over 170 hours of speech recorded by 400 speakers.

TABLE I
COMPARISON OF THCHS30 AND AISHELL

Data Set	Speaker	Hour	Utterance
thchs30	30	27.33h	10893
aishell	340	150h	120098

Although we introduced in the previous paper and did not do much processing on the input speech information to ensure the integrity of the information, we still need to do some operations to extract the frequency domain features of the speech to facilitate the lagging network to learn the features.

We start by framing and windowing the normal speech signal. The sliced signal still needs to be Fourier transformed, but we discard the manual filtering and weighting operations. Instead, we choose to directly input the "time-frequency" amplitude spectrum obtained after the fast Fourier transform and modulation into the neural network as a logarithm.

For our statistical N meta-linguistic model generation algorithm for the pinyin-to-text part, we directly used open-source categorized statistical single-word and two-word phrases instead, as shown below:

1	3941753		1	3941753	
2	中国	16703	2	的	213994
3	公司	13638	3	一	72404
4	我们	10492	4	是	69047
5	一个	9240	5	在	63563
6	可以	7954	6	不	61699
7	服务	7748	7	了	60856
8	有限	7517	8	有	59005
9	工作	6489	9	人	52706
10	游戏	6459	10	中	52005
11	信托	6283	11	国	50797

Fig. 3. Data in N meta-linguistic model generation algorithm.

For the keyword extraction part, it is only necessary to carry out data pre-processing operations such as word separation, lexical annotation and removal of deactivated words for the long sentences previously recognized by speech. Finally a total of n candidate keywords are obtained, and the keywords are constructed as a graph. Then the probability of keywords can be solved iteratively.

B. Target Detection

For the section of target detection, The data were collected from the MsCOCO dataset. The dataset provides 200,000 images with corresponding labels in 80 categories, and is widely used in the field of target detection. With the help of the dataset, the model could learn the characteristics of our objects to be grasped and decide whether an unknown object is or is not the required one.

After learning and training with the dataset, we can use the trained model to meet demands. The input for the model under working environment are the image and depth data acquired from the camera in the simulation environment. The image should be in general formats like .png so that the algorithm could recognize it as the input. Also, the keyword from our speech input should also be involved to filtrate the various objects identified in the image. With the filtration, the location of only one identified object could be produced and delivered to the section of grasp planning.

C. Grasp Planning

For the section of grasp planning, research in this domain has predominantly focused on establishing associations between human labels and graspable regions in RGB-D images[8] or point clouds [7]. Lenz et al.[8] compiled a dataset comprising over 1,000 RGB-D images, annotated with successful and unsuccessful grasping regions by humans. This dataset has been utilized to train efficient CNN-based detection models [13].

Additionally, GQCNN requires two inputs: the camera intrinsic matrix and the depth information containing the grasped object. The camera intrinsic matrix corresponds to the depth camera used in the recorded data. The depth information is an array data returned by the depth camera, with the file extension .npy. It is important to note that the depth information required as input to the model should only include a single object. If there are two or more objects present, it may result in incorrect grasping behavior. In such cases, preprocessing steps, such as cropping the depth information and masking irrelevant values, are required.

IV. METHODS

A. Speech recognition system based on deep learning implementation

1) Feature Extraction: The extraction algorithm of MFCC features we mentioned above is not so much about extracting features, as it is just about pre-processing the sound signal. Conventional MFCC features, after Fourier transform, are present with various types of artificially designed filters, such as Mel filters. These artificial auditory feature-based speech feature extraction are based on some prior knowledge. For example, people are not sensitive to hearing high frequency signals, then this type of processing will cause a large loss of the speech signal in the frequency domain, especially in the high frequency region. And in order to reduce the computation of slicing operation, those traditional speech feature extraction algorithms will also produce very large time window offset in the time domain. So it will also lead to the problem of sound information loss, especially when the speaker speaks faster.

We choose to leave further feature extraction to a subsequent neural network model. The neural network can automatically learn Mel filter-like extraction features during the training

process, which often outperforms traditional feature extraction algorithms in current practical applications since it contains more information that traditional algorithms discard.

2) Acoustic models implemented by convolutional neural networks: In the past years, speech recognition has made great breakthroughs. IBM, Microsoft, Baidu and many other organizations have launched their own Deep CNN models to improve the accuracy of speech recognition. The process of trying Deep CNN is also roughly divided into two strategies: one is the acoustic model based on Deep CNN structure in HMM framework, CNN can be VGG, residual connected CNN network structure, or CLDNN structure. The other one is the end-to-end structure, which is very popular in the last two years. Examples are end-to-end modeling using CNN or CLDNN in CTC framework, or coarse-grained modeling unit techniques such as Low Frame Rate and Chain model, which are proposed recently.[6]

We tried to train acoustic models based on Keras and TensorFlow frameworks, using VGG's deep convolutional neural network as a network model.[1]VGG was proposed by the Visual Geometry Group at Oxford. Its main work is to demonstrate that increasing the depth of the network can affect the final performance of the network to some extent.

In brief, in VGG, three 3x3 convolutional kernels are used instead of 7x7 convolutional kernels, and two 3x3 convolutional kernels are used instead of 5x5 convolutional kernels, the main purpose of this is to improve the depth of the network while guaranteeing to have the same perceptual field, and to improve the neural network to some extent. 5x5 convolution is viewed as a small fully connected network sliding in the 5x5 region. We can first convolve with a 3x3 convolutional filter, and then connect this 3x3 convolutional output with a fully connected layer, which can also be seen as a 3x3 convolutional layer. This way we can cascade (superimpose) two 3x3 convolutions instead of one 5x5 convolution. With such a network, we tried to construct the prediction of speech spectrum features to pinyin.

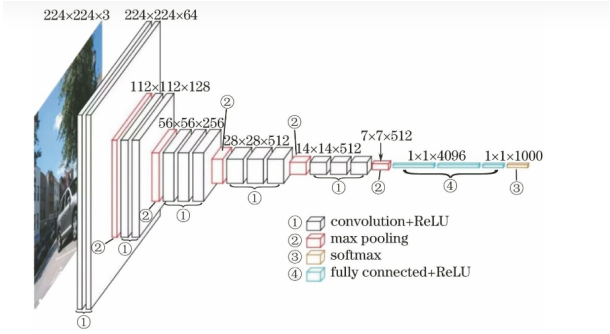


Fig. 4. VGG network structure diagram.

3) CTC decoding: The output of the acoustic model of speech recognition system often contains a large number of consecutive repetitive symbols, so we need to merge consecutive identical conformations into the same symbol, and then remove the silence separator marker to get the final actual sequence of phonetic symbols of speech.

The core of CTC sorting is the introduction of the space character to solve the problem of repetition in the corresponding characters, while counting all the syllable combinations that can form a word and accumulating the probability to arrive at the most likely result.

The CTC calculation is based on the following equation:

$$p(Y \mid X) = \sum_{A \in \mathcal{A}_{X,Y}} \prod_{t=1}^{T} p_t\left(a_t \mid X\right)$$

Based on the designed directed graph shown in Fig. 5., we can obtain all possible combinations of a sequence and calculate the probability of the corresponding sequence. The maximum probability is the phonetic output of our corresponding speech.

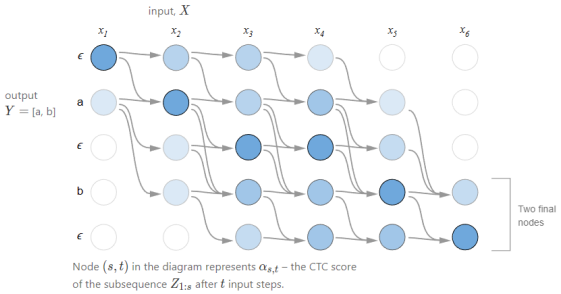


Fig. 5. CTC directed graphs.

4) Statistical language models: from Chinese pinyin to text: Theoretically, if S is a meaningful sentence consisting of a sequence of words w1,w2,...wn (n is the length of the sentence), then the probability that the text S holds, i.e., the probability P(S), is the probability that the first word occurs multiplied by the probability that the second word occurs under the condition that the first word occurs, multiplied by the probability that the third word occurs under the condition that the first two words occur and then the probability of the third word occurring under the conditions of the first two words, all the way to the last word. The probability of occurrence of each word is related to all the previous words, so we have the following formula:

$$P(S) = P(w_1, w_2, \ldots, w_n) = P(w_1)^* P(w_2 \mid w_1)^* P(w_3 \mid w_1, w_2) \ldots P(w_n \mid w_1, w_2, \ldots, w_{n-1})$$

But such iterative probabilities are difficult to calculate, so based on Markov assumptions, the probability of the current word can have a fairly good accuracy rate if only the previous word is considered. And in practice, it is usually enough to consider the first two words, so the formula can be simplified as follows:

$$P(S) = P(w_1, w_2, \ldots, w_n) = P(w_1)^* P(w_2 \mid w_1)^* P(w_3 \mid w_2) \ldots P(w_n \mid w_{n-1})$$

As for the acquisition of frequency, we follow the theorem of large numbers that relative frequency is equal to probability, as long as the statistics are sufficient. Based on the dataset of

the corpus, we can then obtain the probability of a single word and the probability of a phrase based on a word.

As for the probability-based word selection, we choose the Viterbi algorithm to help us do dynamic planning. The Viterbi algorithm simplifies the entire process, iterating through the previous possibilities in terms of nodes, and assigning the minimum value to the calculation after this node. The entire workflow is shown in Fig. 6.. In terms of efficiency compared to brute force traversal of all paths, the Viterbi algorithm removes the paths that do not meet the shortest path requirement when it reaches each column, greatly reducing the time complexity. Also, since all nodes are traversed, there are no incorrect solutions. We may also try to solve this problem later by adopting the idea of reverse regression for the DP problem.

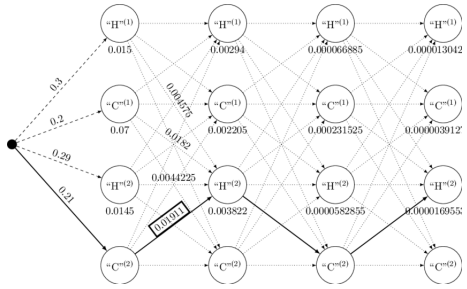


Fig. 6. Work flow of Viterbi.

5) Keyword Extraction: The TextRank algorithm is a graph-based ranking algorithm for keyword extraction and document summarization. Its basic idea is to consider a document as a network of words, and the links in this network represent the semantic relationships between words. Then the importance of the word is the degree of contribution of the words around the word. After several iterations, the importance of all the words will tend to a stable value, and the words with large importance are the keywords. The iterative formula is as follows:

$$WS\left(V_i\right)=(1-\mathrm{d})+\mathrm{d}*\sum_{V_j\in In(V_i)}\frac{W_{ji}}{\sum_{V_k\in Out(V_j)}W_{jk}}WS\left(V_j\right)$$

B. Target Detection

The network of YOLOv5 is divided into input layer, backbone layer (benchmark network), neck layer and output layer. A total of four models are included natively with YOLOv5, namely YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. They are ranked by the width and depth of the network. Since the scene and objects we used for experiment are relatively simple and clear, the model of YOLOv5s is sufficient for our use. The original amount of 80 classes can cover our mission of target detection.

With the key word already acquired, it is urgent to identify the corresponding object and find out its location. The input image from the camera will be split into an S*S grid when entering the algorithm. When the center of an object falls into a grid, it will be detected. A total of M targets can be detected out of one grid, and each target returns five values, including the position values (x, y, w, h), as well as the confidence value of the prediction. So the output for the input image is a tensor of S*S(5*M+N). Meanwhile, the irrelevant objects should be seen as disturbances and ignored in the process. To realize the goal, restrictions are given to the model to output the location of the given object only. To handle the process better, the original output of a labeled image has been replaced by the data of boundaries of the identified feature. The boundaries are a set of integers in pixels, which directly decides the area of the feature on the image and helps set the boundaries for the range image. With the boundaries set, the effective range data can be split from the complete data and transmitted to the phase of grasp planning, where the pose of grasp is computed.

C. Grasp Planning

In order to perform grasp pose estimation from camera images, we have identified two models: GQCNN and another neural network system proposed by Lenz et al.[8]. However, these two neural network systems have different input requirements, where GQCNN only requires depth map input, while Lenz's model [8] requires RGBD image input. After comparing the two models, we opted for the utilization of the GQCNN neural network in our project to streamline the input. We obtained the pre-trained CNN model from their official website and employed it for grasp pose detection.

1) GQCNN: GQCNN takes 2.5D depth images as input and produces four parameters as output: the x, y coordinates and the depth information in the z-direction of the predicted grasping position, and the angle between the grasping gripper and the horizontal direction. Therefore, in our project, it is appropriate to preprocess the depth information captured by the depth camera within the simulator and pass it to the GQCNN model for grasp pose estimation. Due to the sensitivity of GQCNN to camera parameters, if the input camera parameters do not correspond to the provided depth values, there may be a phenomenon of grasping point displacement. By referring to the RGB camera and RangeFinder API documentation of Webots, we obtained the formula for calculating the focal length of the camera in Webots.

$$f_x = (Width/2)/tan(fieldOfView/2)$$

$$f_y = (Height/2)/tan(fieldOfView/2)$$

The variables "Width," "Height," and "CameraFieldOfView" in the formula represent the parameters of the camera in Webots.

D. Manipulator grasping

First of all, according to the information received from the range finder in the previous step, we determined the position of the object, which is also the position where the end effector need to reach. At the same time, we determined the position and the pose of the end effector through the GQCNN neural network.

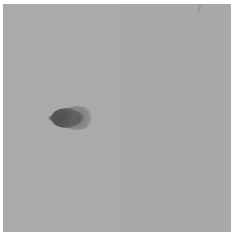
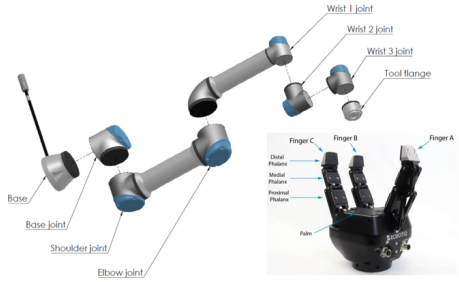


Fig. 7. The depth image of the grasped object.

Fig. 8. Illustration of grasp pose points predicted by GQCNN.

Next, we continued to solve the inverse kinematics of DH in Python environment. Considering the applicability of the manipulator and the open source content, we selected UR5e and pioneer-3-gripper with the DH table to obtain the position information of each joint after calculation.



UR5e							
Kinematics	theta [rad]	a [m]	d [m]	alpha [rad]	**Dynamics**	Mass [kg]	Center of Mass [m]
Joint 1	0	0	0.1625	π/2	Link 1	3.761	[0, -0.02561, 0.00193]
Joint 2	0	-0.425	0	0	Link 2	8.058	[0.2125, 0, 0.11336]
Joint 3	0	-0.3922	0	0	Link 3	2.846	[0.15, 0.0, 0.0265]
Joint 4	0	0	0.1333	π/2	Link 4	1.37	[0, -0.0018, 0.01634]
Joint 5	0	0	0.0997	-π/2	Link 5	1.3	[0, 0.0018,0.01634]
Joint 6	0	0	0.0996	0	Link 6	0.365	[0, 0, -0.001159]

Fig. 9. Parameters of mechanical arm and gripper

Then, after setting the input parameter mode, we pass the joint angle into Webots internal controller with the same variable name. Then we write the corresponding relation and grasping logic of the robot arm joint to manipulate the robot arm. (Refer to file p_controller.c for details.) In this step, considering the working range and singular solution of the manipulator, we set the manipulator at the height of 0.5m to improve the grasping success rate.

Finally, we can see that with the simulated physics, the object can be directly grasped to verify the accuracy of the grasping position and DH solution.

V. Experiments

In order to better highlight the robustness and high success rate of this project in speech recognition capture, first of all, we designed a series of verification experiments specifically for speech recognition. Specifically, the first part is the verification of speech-to-text ability, that is, to judge whether our speech recognition function can successfully convert speech signals into text completely and accurately. The second part is to verify that key information can be extracted smoothly in some special cases, in this case, long text and some special dialects of inland China, such as Cantonese.

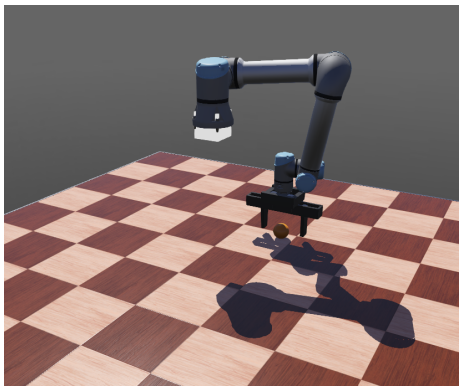


Fig. 10. Parameters of mechanical arm and claw

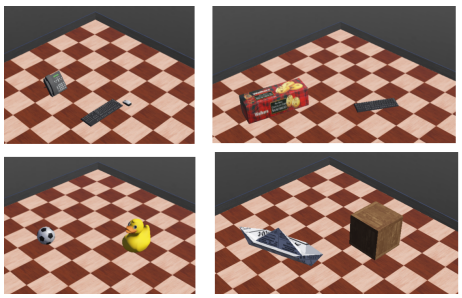


Fig. 11. Experimental scene

Secondly, in order to verify the overall realizability of this project, we designed different grasping scenarios and different grasping targets and conducted several experiments respectively, as shown in Fig. 11. Our experimental scene was built in the Webots environment, and each scene contains at least two objects in our daily life, so as to verify the consistency and success rate of the overall process of our project. Specifically, we take an RGB image and a speech signal as input, and then process it layer by layer to see if the robot arm can smoothly pick up the desired objects from the scene with multiple objects. Here, the condition of success is that the object can be smoothly formulated to grasp and extract to a certain height. Finally, in the experiment of our overall process, the difference of success rate of each part was highlighted. We calculated the specific success rate difference of each part.

A. Speech recognition system

1) Chinese Speech Conversion: For our test set we trained the network to achieve an overall correct rate of over 80%, and then we did some project-specific training.

To test the robustness of the system, we added noise to interfere with the recognition on the premise of the original speech, and the experimental results were equally correct. Our initial guess for this robustness is that the input speech spectrogram is transformed into a high-frequency signal when

the Fourier changes, and the high-frequency noise has less weight in the VGG network, so it is not affected.

We then tried using dialects, using the Sichuan version and the Shaanxi version of ”Give me a red apple”, and we did not get very good results with either. We suspect that the change in pitch is very misleading to our recognition, and even if all the phonetic symbols are correct and only one word is incorrect, it will have a serious impact on our overall recognition.

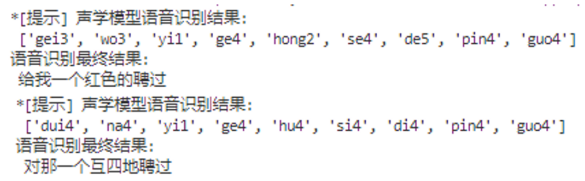


Fig. 12. Attempts on dialects version

We then tried long sentences, hoping to explore whether complex, illogical mixed sentences would have an impact on recognition. And it turned out that the model still maintained a high level of correctness for this type of sentence, with the results shown in the following figure:

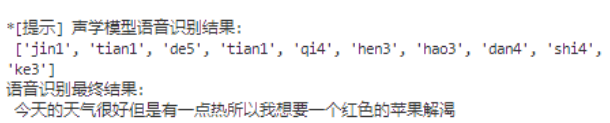


Fig. 13. Attempt on long, complex, illogical mixed sentences

Finally we tested the model on a large scale. For 43 times out of 50 tests, the model was able to perfectly reproduce the utterances we described by speech. In 5 of the remaining cases, although some words were wrong, the key information such as ”apple” and ”red” were correctly recognized. And in the remaining 2 cases, the recognition failed due to poor pronunciation.

2) Keyword Extraction: Our whole task is to reduce the semantics through speech and extract the core actions from the semantics, then keyword extraction is a very important part. For our robotic arm task, the core keywords are grasping the item, a description of the grasped item, the grasping position, and the grasping action.

For our first attempt, the semantic input was ”put the red apple on the table in front of me”, and we were able to recognize perfectly: the grasping object - apple, the descriptor -red, grasping action - put in front, but not the orientation word ”in front”. We guessed that orientation words such as ”in front”, ”behind”, ”left” are usually found at the end of the sentence as position designation, so there is basically no association with the word that follows. And unlike nouns, which can have many combinations with other words, orientation words are weakly associated with other words.

We then proceeded to extract feature words from the long sentences in the previous section to investigate whether the model could accurately locate long sentences with complex

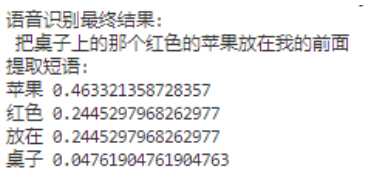


Fig. 14. First attempt of Keyword Extraction

logic. The result is obvious, although the keywords are mixed with interfering words such as ”weather” and ”heat”, ”apple” is still extracted as the first keyword. Unfortunately, ”red” was dropped to the back of the list.

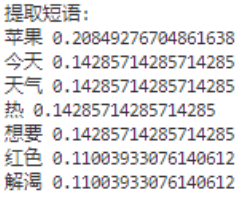


Fig. 15. Keyword Extraction on long sentence

We then replaced the crawled items to test the model, as we wondered if the word ”apple” was too specific to rank high. We entered the sentence ”I love bananas, the table is big, the weather is nice, give me some encouragement, put the red banana on the table in front of me”, and we switched the position of apple and banana, and the two outputs were exactly opposite. This shows that the real reason for the front position of ”apple” is that the phrase it is in is long and is the body of the whole long sentence.

Fig. 16. Keyword Extraction on changing main item

We finally did a large-scale test of the feature extraction function, and for 50 long sentences mixed with various interfering words, the model always extracted information such as ”apple”, ”red”, ”take ” information. And the key item ”apple” always ranks in the top 3.

B. Whole process experiment

The first is the design of the experimental environment. The construction of the whole experimental environment is based on Webots. In the world, we have placed a number of experimental objects, including large and small, animals, daily supplies and food. More importantly, in order to meet the needs of object recognition and grasp position and pose

judgment, RGB-D camera is absolutely necessary. However, because there is no corresponding camera in Webots environment, we decided to add RGB camera and depth camera to the same position in the world environment, and adjust the internal parameters of the two cameras, so as to ensure the pixel alignment of the two cameras. At the same time, in view of the grasping content of mechanical arm in this project, we also added UR5e mechanical arm into the world.

The second part is the summary of experimental data. Besides the most important part of speech recognition, this project also includes many other parts, such as object recognition and grasping posture. So we're counting the success rates of all the other parts as well as the overall success rates. Specifically, total SR represents the success rate of the overall experimental process, while training SR and testing SR also represent the success rate of the overall experimental process, but the biggest difference between them is that the statistical experiment of training SR is directly input with text labels. That is, the speech recognition part is omitted and the success rate of the implementation of other parts is directly considered, while testing SR statistics experiments with normal speech signals as input. In addition, we also specially consider the tp&spec SR that only recognize the position and posture of objects and grasp by robotic arms, and others SR that only recognize objects. Most importantly, in order to justify the project we designed, we didn't waste too much time. We also specially counted the processing time of the whole experiment.

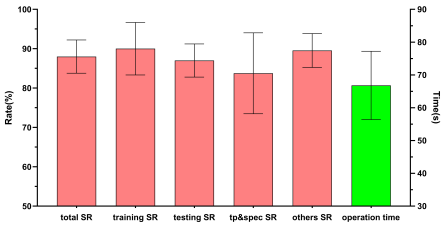


Fig. 17. The result of mean and SD of total SR (mean=88.0%, SD=4.2%), training SR (mean=90.0%, SD=6.7%), testing SR (mean=87.0%, SD=4.2%), tp&spec SR (mean=86.0%, SD=8.4%), others SR (89.0%, SD=5.2%) and operation time (mean=66.8s, SD=10.4s). The y value of the first five is represented by the left axis, the unit is % and the y value of operation time is represented by the right axis, the unit is second.

The final stage is the sorting of experimental data, as shown in the Fig. 17. First, the success rate of our overall project is close to 90%, which means that our project has a certain stability and success rate. Second, through the comparison of testing SR and training SR, it can be found that our speech recognition project is mature enough, that is, there is no significant difference between the use of speech recognition and the direct input of expected objects in the experimental results. This means that the speech recognition part of us is already sophisticated enough to extract the keywords in the speech signal. Thirdly, both object pose recognition and grasp by mechanical arm and object recognition have enough amazing success rate in nature. Although the success rate of grasp and grasp by mechanical arm has great fluctuation and the lowest average value, the overall success rate will not be significantly affected due to the sufficient clamping force and friction degree of mechanical claw. Fourthly, the process time of our whole project is restricted to about 65s, but in fact, the operation time here is calculated based on the corresponding time of RGBD image generation and mechanical arm. Therefore, after the whole project is truly integrated, this time will be further shortened, which reflects the superiority of this project in grasping.

VI. Conclusion

A. Speech recognition

Our speech recognition system can achieve a certain degree of accuracy and robustness for noisy speech input and long speech input, but it does not get good results for dialects that change intonation, which is attributed to the strong correspondence between pinyin tones and text, and once the pinyin tones are wrong, recognition will basically fail, and the results of our final large-scale tests are as follows:

TABLE II
Speech recognition results

Quantity	Result
43	Perfect recognition of voice translation to complete all semantics
5	Mis-translated some unimportant speech but successfully translated core semantics, e.g., grasping objects and descriptions of grasping objects
2	Did not complete accurate translation because of bad pronunciation

B. Keyword Extraction

Since we did not use crawling as a training context, the priority of the keyword depends only on the proportion of the word in the whole sentence. We found that nouns and adjectives are more easily extracted because they have better compatibility with other words and can form various phrases, but orientation words are not easily extracted. And the closer the word is to the middle of the sentence and not located at the ends of the sentence, the higher the extraction priority is. And the higher the proportion of the content of the main clause to which the word belongs to the whole sentence, the higher the priority is.

C. Target detection

First of all, our target detection system has a large library of objects, that is, it can quickly identify almost all objects in daily life, and it will not be limited by strange objects. Additionally, the target detection system can reach great accuracy in our simulation environment. For most of the objects captured with the camera in our experiments, the model can identify them and give out their locations. However, some problems do exist in the process of detection. Examples are that if the object sits on the ground with its side facing the camera, the model is not able to identify it due to limited features. And if the object is tilted in the picture due to wrong poses of

camera, there is a chance for the model to misrecognize the features. For future improvements, data from different angles of a certain object can be added for training the model.

D. Grasp planning

GQCNN can generate grasp points for various complex objects with high success rate. We conducted over 20 object grasp tests with GQCNN, and it was able to generate effective grasp points for 91.3% of the objects. Sometimes, the grasp points generated by GQCNN are located far from the object's center of gravity, requiring high gripping forces and friction coefficients to ensure a secure grasp and prevent slippage on the gripper. Second, GQCNN also has extremely high robustness. Specifically, during the experiment, the shooting of our camera will have wide-angle distortion, but this will not affect the final result.

GQCNN encounters difficulties in solving grasp poses for objects with specific shapes, such as smooth spheres or circular cans. GQCNN cannot adjust grasp points based on gripper size or gripper type. Additionally, since GQCNN does not take gripper configuration into account, it generates many unreasonable grasp points for our specific gripper setup. For example, in the grasping experiment with a small sheep toy, GQCNN generated grasp points at the location of the sheep's legs. However, due to the relative size difference between our gripper and the sheep toy, it was unable to grasp the sheep's legs, resulting in an error.

Fig. 18. Errors occur when there is a mismatch in size between the fixture and the object being grasped.

E. Manipulator grasping

1) Work completed: In this part, our main work content are: First, select and set parameters of mechanical arm and claw. Second, obtain the target position and claw position, and get the joint angle by DH inverse kinematics solution. Third, send parameters into Webots controller and edit manipulator controller. The last one is change the environment and objects, then obtain data through repeated experiments.

2) Problems and future improvement: The first problem is in the calculation method of DH, it is relatively basic and does not fully consider the real factors such as joint interference. So for our next move, we could add calibration and compensation in an analytical way to avoid such circumstances. The second problem is also vital that there is no path planning for manipulator control, For example, in the process of grasping, there may be interference of the robot arm's own joints, as well as interference between the robot arm and objects in the environment. Therefore, in order to solve such problems, we will use ROS-MoveIt! as an external controller to integrate movement solutions and path planning.

ACKNOWLEDGMENTS

REFERENCES

[1] Andrew Zisserman Andrew Zisserman. End-to-end acoustic modeling using convolutional neural networks for hmm-based automatic speech recognition. 2019. doi: 10.48550/arXiv.1409.1556.

[2] Ravi Balasubramanian, Ling Xu, Peter D. Brook, Joshua R. Smith, and Yoky Matsuoka. Physical human interactive guidance: Identifying grasping principles from human-planned grasps. *IEEE Transactions on Robotics*, 28(4):899–910, 2012. doi: 10.1109/TRO.2012.2189498. URL http://dx.doi.org/10.1109/TRO.2012.2189498.

[3] Jeannette Bohg, Antonio Morales, Tamim Asfour, and Danica Kragic. Data-driven grasp synthesis—a survey. *IEEE Transactions on Robotics*, 30(2):289–309, 2014. doi: 10.1109/TRO.2013.2289018.

[4] Y. Zhu C. Wang, D. Xu and R. Mart´ın-Mart´ın. "densefusion: 6d object pose estimation by iterative dense fusion". *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, page 3343–3352, 2019.

[5] F. Li X. Ding D. Zhang, J. Hu and A. K. Sangaiah. "small object detection via precise region-based fully convolutional networks". *Computers, Materials and Continua*, pages 1503–1517, 2021.

[6] Ronan Collobert Dimitri Palaz, Mathew Magimai-Doss. End-to-end acoustic modeling using convolutional neural networks for hmm-based automatic speech recognition. pages 15–32, 2019. doi: 10.1016/j.specom.2019.01.004.

[7] Daniel Kappler, Jeannette Bohg, and Stefan Schaal. Leveraging big data for grasp planning. In *2015 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA)*, IEEE International Conference on Robotics and Automation ICRA, pages 4304–4311. IEEE, 2015. ISBN 978-1-4799-6923-4. IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, MAY 26-30, 2015.

[8] Ian Lenz, Honglak Lee, and Ashutosh Saxena. Deep learning for detecting robotic grasps. *INTERNATIONAL JOURNAL OF ROBOTICS RESEARCH*, 34(4-5, SI): 705–724, APR 2015. doi: 10.1177/0278364914549607.

[9] H. Wang X. Huang N. Li, X. Ye and S. F. Tao. “an improved yolov5-based method for sar image ship detection in complex scenes”. *Signal Processing*, page 1–16, 2021.

[10] Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. pages 3406–3413, 2016. doi: 10.1109/ICRA.2016.7487517.

[11] Florian T. Pokorny and Danica Kragic. Classical grasp quality evaluation: New algorithms and theory. pages 3493–3500, 2013. doi: 10.1109/IROS.2013.6696854.

[12] Domenico Prattichizzo and Jeffrey C. Trinkle. Grasping. pages 955–988, 2016. doi: 10.1007/978-3-319-32552-1_38. URL https://doi.org/10.1007/978-3-319-32552-1_38.

[13] Joseph Redmon and Anelia Angelova. Real-time grasp detection using convolutional neural networks. In *2015 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA)*, IEEE International Conference on Robotics and Automation ICRA, pages 1316–1322. IEEE, 2015. ISBN 978-1-4799-6923-4. IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, MAY 26-30, 2015.

[14] V. Lepetit S. Hinterstoisser and S. Ilic. “model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes”. *Asian conference on computer vision*, page 548–562, 2012.

[15] V. Narayanan Y. Xiang, T. Schmidt and D. Fox. “posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes”. 2017. doi: 1711.00199.

[16] J. Yu and S. Luo. “a yolov5-based method for unauthorized building detection”. *Computer Engineering and Applications*, page 236–244, 2021.