Technical Paper of KIARO Team for Participating in RoboCup @Home 2024 SSPL

Team name

KIARO

(Khatam Institute of Advanced Robotics)

Author names

Abdolah Shamisa, Elaheh Rahpeyma

Electrical Engineering Department, Faculty of Engineering, Khatam University, Tehran, IRAN

Corresponding Author: A. Shamisa, a.shamisa@khatam.ac.ir

About KIARO

KIARO (the previous name of our team is KU Robotics) was founded September 2017 in Khatam University (Non-governmental and private University). The initial members of the team included faculty members, students and some graduates of Khatam University. At first, a research and development center including all necessary hardware and software tools to carry out robotics projects was created. Conducting students' robotics lesson projects and providing robotics-related training was one of the primary activities of the team.

In 2017, the first industrial project of the robotics team started with the title "Research to startup a humanoid robot to provide banking services". After initial studies and market research, we purchased a Pepper robot from SoftBank Robotics. Some basic algorithms were performed (for about one year) for Pepper startup as a banking service robot for Pasargad Bank (the best private bank in Iran). In this project, the complete understanding of the hardware-software platform of the Pepper robot and the implementation of various intelligent scenarios were done.

After the completion of the first project, in the second phase of the project, a new project titled "Research and development of humanoid robot software for banking services" was started. We worked with APIs and SDKs of Pepper and implemented different algorithms such as navigation, people detection and recognition, interaction in Persian language, Mask detection, and Object detection and so on in a bank environment.

In 2021, the KIARO team participated in the RoboCup 2021@Home Education competition, which was held online and remotely due to Covid-19, and KIARO team (KU Robotics) won the "Best Technical Video Award, Standard Platform, First Place" award.

After that, the Pepper robot was used as an intelligent agent in the smart industry living lab. The real-time interaction of the robot on the industrial internet of things (IIOT) platform with the industrial process control equipment in the laboratory environment was developed. Also, Human-Robot interaction using gesture and voice commands has been done in the past years and is being developed. the Man-Robot-Machine interaction scenario on IIOT platform is our goal in the next coming year. The KIARO team has also presented articles in national conferences and is currently the only team in Iran country that works on the standard platform (Pepper robot) and has the experience of implementing any intelligent algorithm on this robot.

KIARO Team Experiences

In the more than 5 years, the KIARO has finished many industrial projects. in addition, KIARO has participated in RoboCup @Home Education 2021. In the following, some of main experiences will be briefly explained.

Pepper as an Intelligent Banking service operator

In banking projects, various scenarios were defined by the project employer (Bank administrator), which should be carried out by Pepper. The required tasks was programmed and developed using Python SDKs and NAOqi APIs commands. Due to the specific conditions of the project data security, a specific architecture was considered for the connection between the robot and the server. Network architecture that used is shown in Fig. 1. In this architecture, the Pepper was not connected to Internet directly. A socket programming method on client-server connection was used for local and cloud network. For example, a task to sending an image file to server from Pepper is shown in Fig. 2.

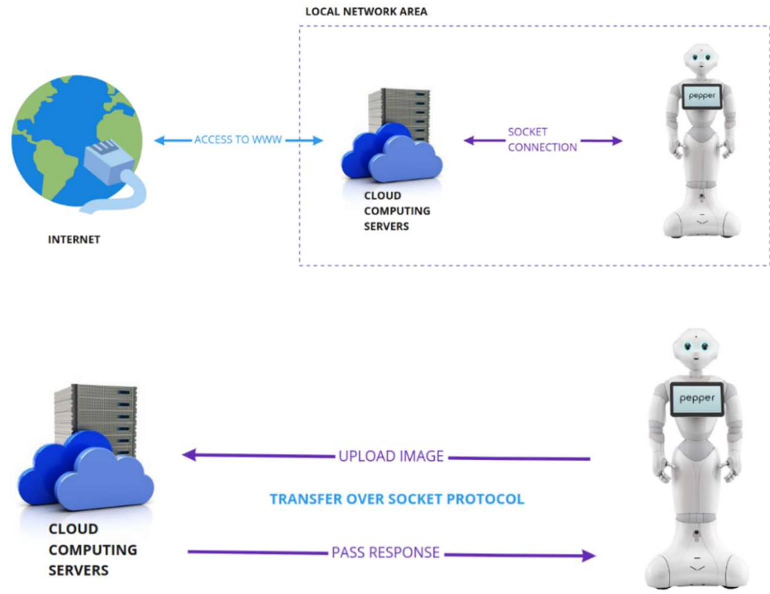
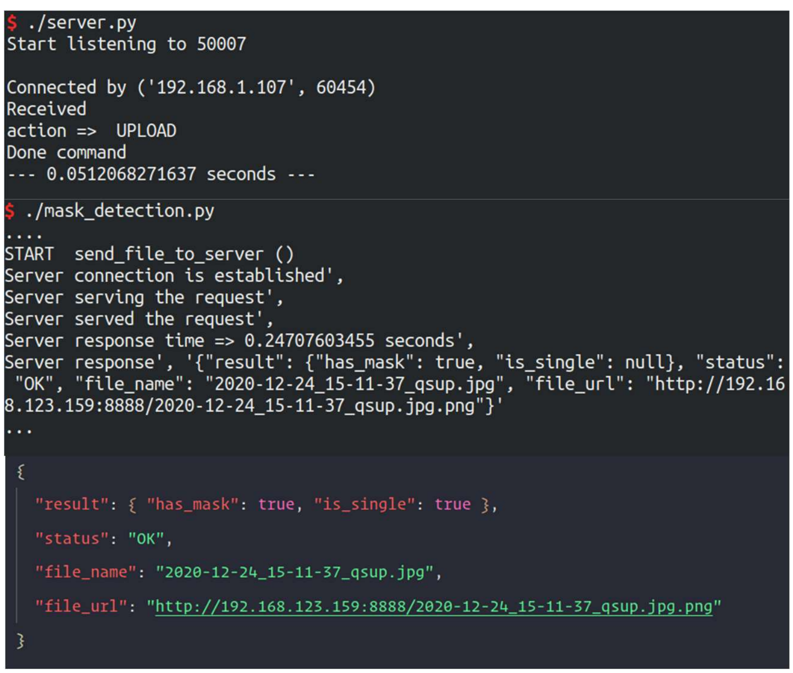


Fig. 1: Network architecture



```
$ ./server.py
Start listening to 50007

Connected by ('192.168.1.107', 60454)
Received
action =>  UPLOAD
Done command
--- 0.0512068271637 seconds ---
```

```
$ ./mask_detection.py
....
START  send_file_to_server ()
Server connection is established',
Server serving the request',
Server served the request',
Server response time => 0.24707603455 seconds',
Server response', '{"result": {"has_mask": true, "is_single": null}, "status":
 "OK", "file_name": "2020-12-24_15-11-37_qsup.jpg", "file_url": "http://192.16
8.123.159:8888/2020-12-24_15-11-37_qsup.jpg.png"}'
...
```

```
{
  "result": { "has_mask": true, "is_single": true },
  "status": "OK",
  "file_name": "2020-12-24_15-11-37_qsup.jpg",
  "file_url": "http://192.168.123.159:8888/2020-12-24_15-11-37_qsup.jpg.png"
}
```

Fig. 2: Socket client-server image transferring

To perform the tasks presented in the scenario of identity recognition and banking services, a two-part design is considered as shown in the Fig. 3. As it is clear in this figure, in this type of design, in addition to the distribution of scenario parts, operational clarity is also achieved.

In general, the process of implementing these two plans is as follows:

1- Mask (face covering) detection

2- Taking a photo and sending it

3- Receiving user information (national code and 10-digit code on the back of the national card)

4- Send information to the server and call the Pod API.

5- Display the image to inform the user of the need to wait to receive the answer

6- Receiving authentication result from POD server.

In order to manage the time of processes and their parallelization, “Redis” has been used.

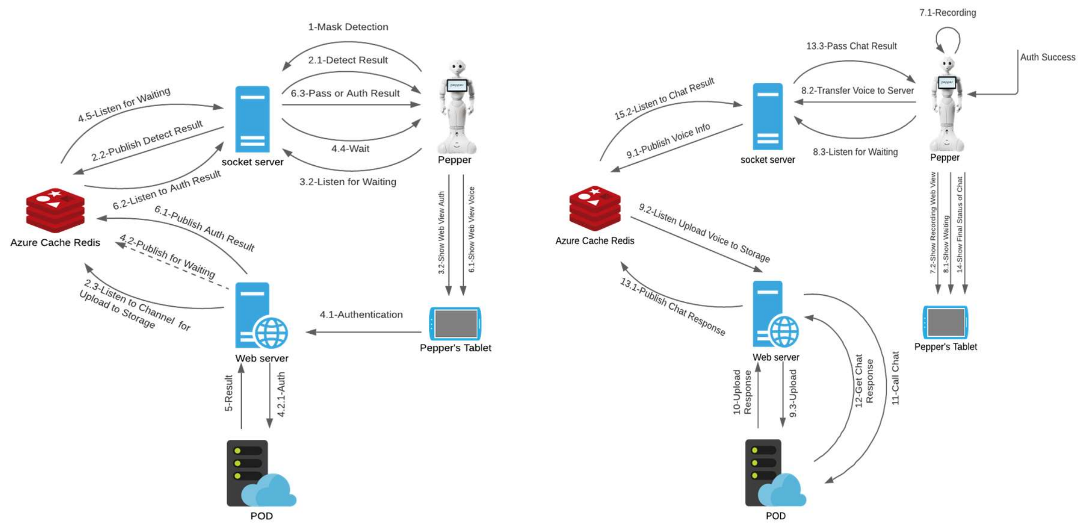


Fig. 3: The two-part plan for banking services scenario

Task implementation process and approach in RoboCup @Home Education 2021

Carry the Luggage

Object recognition

In the planned object recognition tasks, the first part is to recognize the luggage and then estimates its location and distance. This should be done in such a way that the robot can approach and then stands at an acceptable distance from the human and prepares to grab the bag from the person. In the second part, the robot has to determine the arrangement of the chairs and correctly identify the empty seat, then guide the person towards the sitting place.

In these scenarios, the Yolo3 real-time object detection and recognition algorithm; which is one of the most effective object detection algorithms, is used. A data set of paper bags was trained with TensorFlow 2.3 and Keras 2.4 and then converted to Yolo3 format for real-time object identification to achieve the highest performance on the goal object recognition.

The presence of the chairs in the environment is checked by suitable algorithms for detecting empty seats in the video stream received from Pepper's bottom camera using gstram (Fig. 4).

If there are chairs in the frame, then they are sorted by location in the video frame from left to right, and then the presence of humans is checked. The entire chair detection flow was applied in the approach's find my mates component. If there are people in the frame, the overlap of each person and the chair is checked, and the position of an empty chair is provided to the customer depending on the results. In addition, for identifying the location and distance of a paper bag from Pepper, after detecting the bag object in the video stream from Pepper's tablet camera, its location and size are identified, and the distance is computed depending on the size of the bag.

Fig. 4: detecting empty seats in the video stream received from Pepper's bottom camera

Take the bag

To take a bag from a person who is standing and take the bag in the front of its body, Pepper must identify the bag, calculate the distance from the bag, and then begin moving forward. Of course, distance estimation will occur any time Pepper reaches the last target; which is determined by the last distance information. But what is important is Pepper should stop at a specific distance from the bag. After that, it is needed to define a trajectory for its hands, and by way of is mentioned in object recognition, the information of the bag, its size, and the distance is accessible. So, for several standard bags, the size is extracted, and depend on them, four different trajectories of grasping the bag have been defined (Fig. 5). Since a timeline is put on the Choregraphe to monitor joint movements, it was easier to use this choice, because, apart from the increased implementation speed, the moving behavior was smoother.

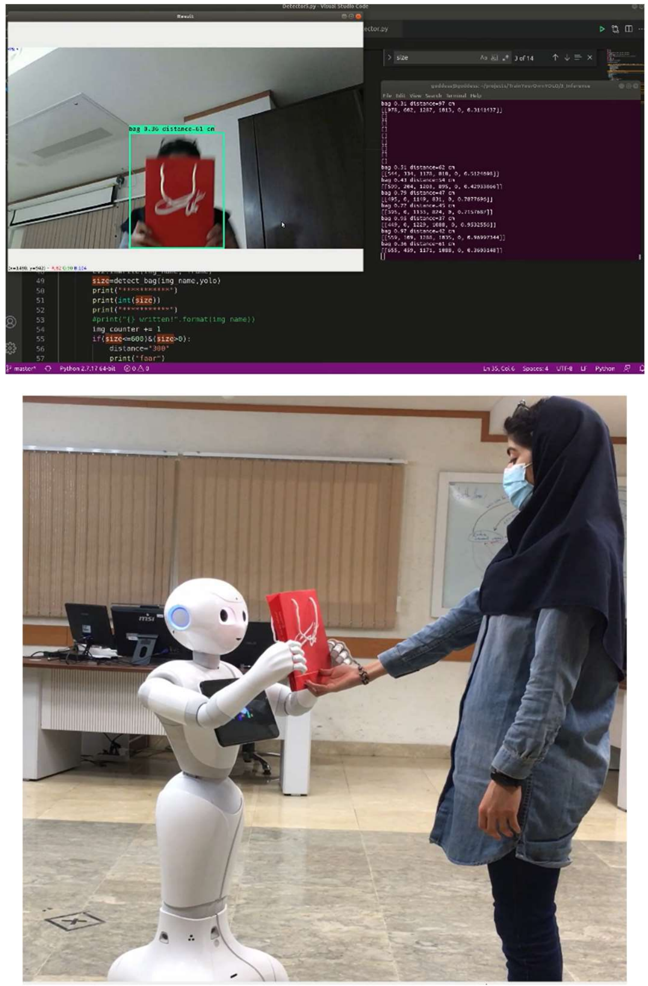


Fig. 5: Bag recognizing and take it from the person

Mapping and navigation

In this section, a first and important part was building a map. Two ways of building a map are using ROS or using the ALNavigation API, where each one of these ways has its challenges. The most unpleasant difficulty when using the API was that Pepper didn't support radius mapping of more than three meters, and if the specified radius was more than three meters, the connection was lost and the session was disconnected. As a result, utilizing APIs for mapping was only appropriate for explorations of less than three meters. There were various packages for utilizing ROS to generate a better map all packages use “slam_gmapping” to map the environment, but because the most appropriate ROS distribution for Pepper is indigo, it was important to modify packages to make Pepper mapping. One of the issues of using ROS for mapping was that the laser range was insufficient, therefore it took a long time to generate a map. After generating a map of the environment, it was needed to localize the Pepper in the map. For moving the robot, the joy node runs to make the joy code of Pepper reachable (with moving the robot all arrows around Pepper modify its position on the map). Finally, the map construction and localizing were completed. The navigation stack was used to autonomously move the Pepper robot from one point to another.

Find My Mates

People recognition and face detection

Since the results of robot APIs for saving people's faces and recognition were unreliable, a new technique for recognizing the face of the person and communicating with the robot was needed. However, one of the challenges of using a new technique was the lack of good results, from the use of Peppers cameras to training and recognition. So, using the Pepper's tablet camera was the only way to avoid using an extra camera, while still getting good results. As a result, the tablet's camera was used for both training and detection. For the training part, the Python Face Recognition Library was used to encode the face that was found and saving it as a record with the name of the individual, who belongs to the face info. A trained KNN model is loaded for the recognition part to predict and measure the distance between recorded data and new face data. If the prediction is less than a certain threshold, the face was not recognized, and the train part should be repeated. When Pepper detects that someone is ready to communicate with it, an ID is sent to the callback function and based on that ID and face identification, the recognize part is executed. If the individual's face information is in the reported data, Pepper greets the person and tells the name of it; however, if an individual is a new person, the training flow will repeat.

Receptionist

Speech recognition and interaction

Pepper speech recognition API (ALSpeechRecognition) is appropriate for greeting, but it was inefficient in two areas: first, there was a need to recognize the individual's name, and second, it was necessary to take the order that was picked from the menu. These reasons make us to mixed Pepper speech recognition API with Google speech recognition. For recording the voice to do processing, two methods were used and tested: first, start and stop recording based on the RMS level of environmental sounds from an acceptable threshold, which was specified based on background noises, and second, using an API to connect event that fired from changes of environment sound level, the value which received was 1 or 0. In the second method of recording, each load voice and the length of speech causes the event to send value 1, but in a sentence, each pause causes the event to send value 0. But after a while, if there was a sound again, 1 was received due to this mechanism. The recording will continue until a certain time has passed, after which it will be paused. Since we give break time after every 0 value, we can ensure that we don't miss the last part of the individual's speaking when we use the second method. For example, when an individual starts to speak, the recording process will begin immediately after it, and even if the individual stops speaking, we can ensure that we didn't miss the last part of the individual's speaking because we give break time after every 0 value.

Using a humanoid robot to recognize people's face masks

Last years, when the world was dealing with the covid-19 disease, according to medical experts, one of the most important factors to prevent the spread of this virus was the use of masks. In this project, the Pepper humanoid robot was used to detect the presence of masks on people's faces in real time. The working method was that first the presence of a human in front of the robot was detected, then the location of his face was determined. Next, facial components were recognized using image processing algorithms, including the use of Haar image cascade classification and OKAO's proposed algorithm. To detect the presence of a mask on the face, the presence of the mouth and nose on the face, as well as the dimensions of the mouth and nose were measured in relation to the dimensions of the face, and then the robot performs the corresponding operations based on this comparison. The results of the implementation of the algorithms proposed in this project on the paper robot show that the speed of its steps and the accuracy of the detection were acceptable for the real-time implementation of face mask detection by the robot.

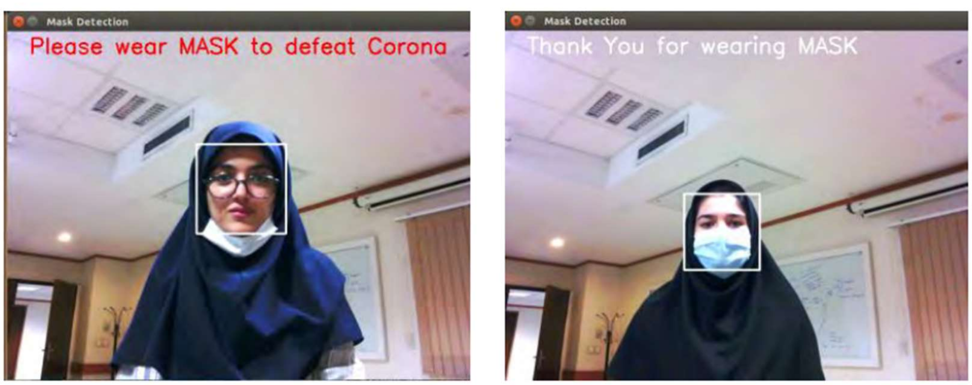


Fig. 6: Mask detection on people's face

A Hand Gesture Human-Robot Interaction for Pepper Robot using ROS

In the current project, the Man-Machine-Pepper interaction on the IIOT platform in a smart industry living lab is in progress. For this goal, a Human-Robot interaction using hand gestures and voice commands should be developed. One of the main purposes of robot application planning/programming is the improvement of interaction between humans and robots. In this project, a method is proposed for the real-time locomotion of a Pepper humanoid robot using hand gesture commands. Five numbers of hand gestures have been trained, enabling robot locomotion in various directions. On the robot side, the hand gestures are converted into a given locomotion command by developed software. The approach is incorporated in an open communication architecture based on the Robot Operating System (ROS). A multi-node processing architecture is designed for implementing the proposed method. Many real situations have been tested in different environmental conditions and the results showed that the performance is acceptable based on two selected criteria.

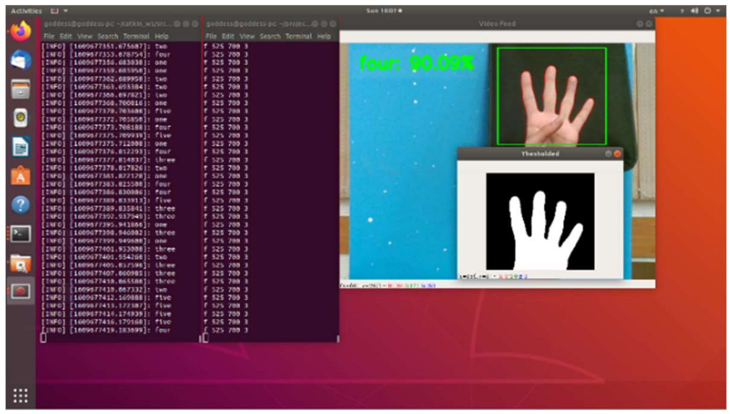


Fig. 7: Hand Gesture using Pepper Cameras

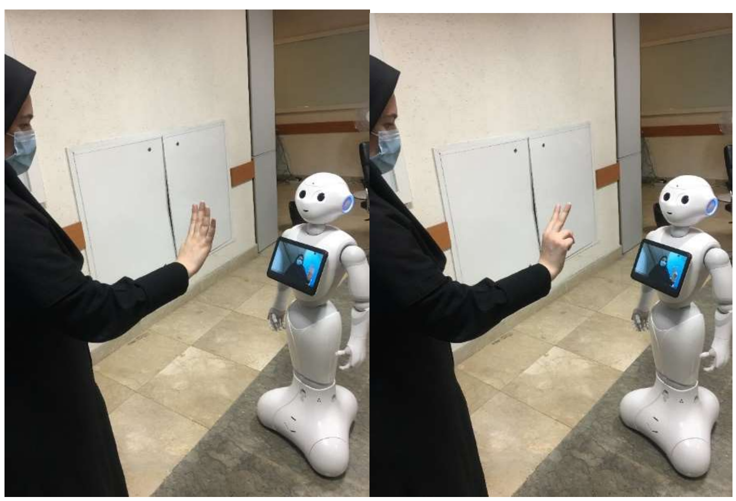


Fig. 8: Human-Robot interaction using hand gesture commands

Conclusion

The KIARO team has several years of technical experience using the Pepper Robot hardware-software platform. Carrying out several industrial projects under a commercial contract with companies, as well as participating in the RoboCup @Home education 2021 competitions, will enable the KIARO team to participate in the international RoboCup @Home SSPL 2024 competitions worthily.