Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 02, February-2021

Deep Learning based 6DOF Robot Arm Control with Computer Vision

Bien Xuan Duong
Advanced Technology Center
Le Quy Don Technical University
Hanoi, Vietnam

Anh Ngoc Mai, Tuan Anh Phan, Vinh Huu Ta,
Bao Quoc Le, Khanh Ba Nguyen, Vuong Ngoc Vu,
Phung Minh Nguyen, Phimmason Keothongbin,
Le Quy Don Technical University
Hanoi, Vietnam

***Abstract* — This article presents the control technique for the six degrees of freedom (6DOF) robot arm to pick up and drop objects based on computer vision and deep learning (DL) techniques. Control commands are performed by the human voice. Information in voice control commands is processed through the voice recognition module and converted to text. Based on the Machine Learning model, characteristic information from the complete text is extracted and encoded into the object recognition module (computer vision) input signal. Model DL2 is built-in computer vision engineering to recognize objects on demand from voice. The outputs of the DL1, DL2, and ML networks are the information needed for converting into robot arm control signals through the DL3 network. The structure of the model DL1 and ML is not clearly mentioned in the scope of this article. A 6DOF robot model is built to perform tests to assess the feasibility of the control method. The research results have important signification for the construction of robotic control systems that use intelligent, flexible, and responsive algorithms in real-time.**

Keywords- Deep Learning, Computer Vision, Robot arm

I. INTRODUCTION

The robot control problem is designed for different tasks such as controlling position, trajectory, or performing tasks of picking and dropping objects. There are many position or trajectory control systems developed such as PID [1], Sliding Mode Control [2], Backstepping [3], Robust control [4], Fuzzy logic [5], to other Intelligent control algorithms such as Adaptive control [6], Neural Network [7], Machine Learning (ML) [8], Reinforcement Learning [9], Deep Learning (DL) [10]. However, modern robots performing pick and drop tasks need to be smarter and more flexible. In addition to identifying the single object and its position, it is also necessary to identify the object that needs to act between many different objects. Computer vision technique developed and incorporated into intelligent control systems is a viable solution to meet the practical needs mentioned above. Furthermore, robotic control techniques performed by voice are also integrated to increase the flexibility of the control system and respond to changing the task of robot in real-time.

The voice control system for robots based on an artificial intelligence algorithm has been reviewed in [11]. An intelligent ear design for the robot to determine the direction of the emitted sound is performed in [12]. The technique of voice recognition is based on the deep learning algorithm proposed in [13]. An autonomous robot is controlled by voice through the Google Assistant application tool on the basis of IoT technology shown in [14].

The problem of object recognition using the artificial intelligence network is mentioned in [15-24]. A Neural Network (NN) model is proposed in [15] to increase the object recognition efficiency on the basis of the data library VOC2007 and ILSVRC2012. A Loss function adjustment structure is proposed in [16] to optimize the overlap of Boundary Boxes. The structure of the YOLO algorithm is described in [17], [18]. The 3DOF robot arm controlled using computer vision based on NN is shown in [19]. Pick-up robots using computer vision is mentioned in [20], [21]. The problem of using data libraries and deep learning methods for the object recognition problem is summarized quite clearly in [22]. The robotic arm control system using the combination of high-frequency brain waves and computer vision techniques is described in [23], [24, 25].

This article focuses on building the control system of the 6DOF robot arm performing object pick-up task based on computer vision technique using deep learning model 1. Deep learning model 2 is used to solve the inverse kinematics problem with the input being the position of the identified object in the workspace and the output being the value of the corresponding joint variables. The joint variable values are converted to control signals for the drive motors ensuring the robot performs motion and object pick-up. DL1, DL2 models are built and tested in PYTHON language with support libraries. The 6DOF robot model is practically fabricated to experiment with object recognition and grasp problems. The paper consists of 3 parts. Part 1 is the Introduction. The material and research method are the contents of Part 2. The last part is the results of the simulation and experiment.

II. MATERIAL AND METHOD

A. Control robot arm by voice recognition and computer vision

Commands by voice (in Vietnamese) via Microphone will be processed and recognized through the DL1 model. DL1 output information is a complete text, including the direction of motion and object features that become the input of the ML model. Object information from the ML output is the target for the DL2 model to identify through the camera and available learning data. The rest of the ML output combined with recognized object information (position, object type, color, shape) gathered as input to the DL3. The outputs of

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 02, February-2021

DL3 are the value of the joint variables that control the robot to perform the task as required. The overall model of the control system can be illustrated in Fig. 1.

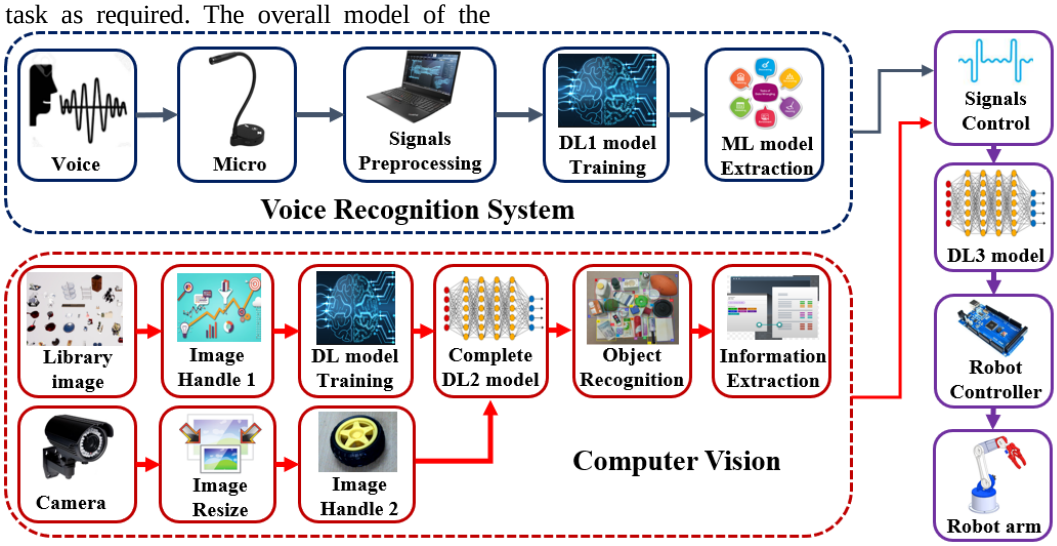


Fig. 1. The overall model of the control system

B. Voice recognition by model DL1

The robotic arm receives voice commands from the operator using the voice recognition module that converts from human voice containing control information to text in the program. The robot control information contained in the voice includes information such as the direction of the movement of the robot (turn to the left or the right), what action the robot needs to perform (the action of grabbing or dropping), identifying the object (wheels, trays, boxes, ...), distinguishing features of objects (color, shape, size, ...).

The input voice and the output control signal must be defined to solve the robot control target. In essence, the voice recognition module is a natural language processing problem, a DL1 model built in order for the network to learn how to convert information from voice to text. The steps to perform the voice control system are determined as follows:

Step 1: Preprocessing the input voice

Step 2: Build a DL1 model to recognize the voice

Step 3: Process the text from the output of the DL1 model

Step 4. Build ML model to extract information and encode information

Step 5: Calculate the motion robot control signal by DL3 model

C. Object recognition based on computer vision technique with DL2 model

After the system recognizes the voice command, the robot arm begins to move in the workspace. At the same time, the camera performs the continuous imaging task to detect and recognize the required object according to the output information of the ML model. When an object is recognized, information about the position of the object is converted to a control signal that moves the gripper to the position of the object to perform a programmed action.

The input to the computer vision module is image data taken from the camera. Image quality must be of sufficient resolution, sharp enough to distinguish objects. The frame of camera rate should be fast enough to ensure the module's real-time object detection. The output of the computer vision module is an object with information that answers questions such as: is there an object in the sight? What kind of object? What color is it? What is the position of the object? Is the object lower or higher than the frame range? Is the subject going left or right of the frame range?

A DL2 network model is built to learn and recognize surrounding objects in real-time to solve this problem. DL2 model is built on the basis of YOLO method [17], [18], [26], [27]. Image data after being taken with the camera will be resized, divided into small cells (Fig. 2)

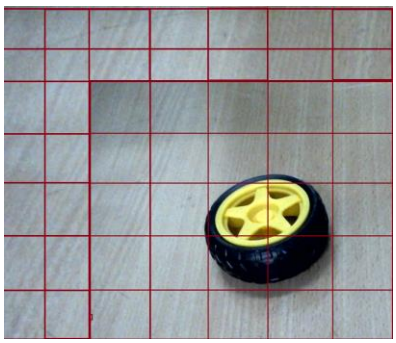


Fig. 2. Create the split grid of frames

The picture is divided into $n \times n$ squares ($n = 1,2,3...$). At each of these squares, 2 boundary boxes must be predicted (Fig. 3), each boundary box contains information such as Whether the square contains the object or not? Predict what type of object name (class) belongs to, Predict the center of the object and the center of the box surrounding the object. Each square is a vector including m values [26]

$$m = 5n_{box} + n_{Class} \quad (1)$$

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 02, February-2021

Where n_{box} is the predicted number of boundary boxes for each square, n_{Class} is the number of object layers in the model (Fig. 4).

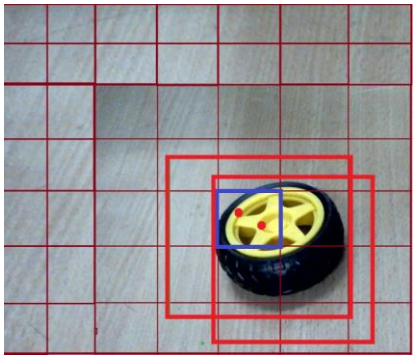


Fig. 3. Two boundary boxes contain the object information

Fig. 4. The predictive information structure of each square

More specifically for model DL2, Input data is the images from gallery (to network training) and camera images (for identification). The output is a *Tensor* containing prediction information for the $n \times n$ squares. The network structure consists of *Convolutional Layers* (size 3×3) combined with *Maxpooling* (size 2×2). The last layer is the *Conv Layer* with a filter size of 1×1. The activation function for the last class is a linear function.

To determine the training error, the DL3 model uses the squared error function between the training results and the predicted results (the real results of the model). According to the output structure of each box boundary, the error consists of 3 parts: Confidence Loss, Classification Loss, and Localization Loss. An Intersection Over Union (IOU) function is used to evaluate the predicted accuracy of the boundary box based on the labeled boundary box parameters (sample data) [28]. The larger the IOU, the better the evaluated model (good trained).

The computed data for the DL3 network model is spatial coordinate set and the corresponding set of rotation angle parameters that were collected and fed into the training DL network repeatedly until the model could give control signals for the robot accurately, to meet the requirements of the problem. After training and assessing responsiveness, the DL3 model is used as a predictive model of robot rotation angle values with recognized object positions in the robot workspace.

III. Simulation and Experiment Results

A. Experimental system

The 3D robot arm model is shown in Fig. 5. The kinematics diagram of the 6DOF robot arm is described in Fig. 6. The fixed coordinate system is $(OXYZ)_0$. Local coordinate systems $(OXYZ)_i, (i = 1, 6)$ are placed on the links accordingly. The joint variable i is denoted by q_i. The real 6-DOF robotic arm is presented in Fig. 7.

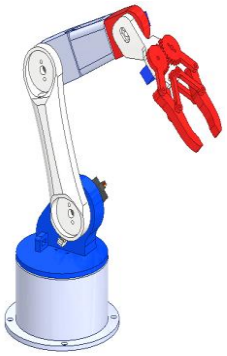


Fig. 5. 3D robot arm model

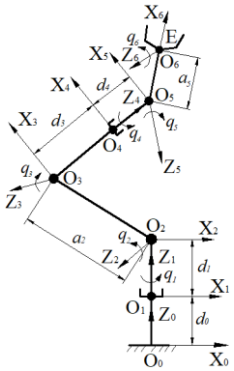


Fig. 6. Kinematics model

Fig. 7. Real robot arm

Kinematics parameters of the 6-DOF robot arm are determined according to DH rule [29] and are given in Tab 1. Homogeneous transformation matrices $\mathbf{H}_i, (i = 1, 6)$ on the links are determined [29]. The position and direction of the end-effector link relative to the fixed coordinate system are represented by homogeneous transformation matrix $\mathbf{D}_6$. This matrix is calculated as follows

$$\mathbf{D}_6 = \mathbf{H}_1\mathbf{H}_2\mathbf{H}_3\mathbf{H}_4\mathbf{H}_5\mathbf{H}_6 \quad (2)$$

TABLE I. DH PARAMETERS

Parameters	q_i	d_i	a_i	α_i
Link 1	q_1	$d_0 + d_1$	0	$p/2$.
Link 2	q_2	0	a_2	0
Link 3	q_3	d_3	0	$p/2$
Link 4	q_4	d_4	0	$-p/2$
Link 5	q_5	0	a_5	$-p/2$
Link 6	q_6	0	0	0

Denote $\mathbf{q} = [q_1 \quad q_2 \quad q_3 \quad q_4 \quad q_5 \quad q_6]^T$ is the generalized coordinate vector and $\mathbf{x} = \begin{bmatrix} x_E & y_E & z_E \end{bmatrix}^T$ is the position vector of the end-effector point. The kinematics equations are determined as follows

$$\mathbf{x} = f(\mathbf{q}) \quad (3)$$

The geometry parameters of the robot are

$$d_0 = 57mm, d_1 = 36mm, a_2 = 120mm, d_3 = 90mm,$$
$$d_4 = 30mm, a_5 = 38mm.$$

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 02, February-2021

The joints variable limit is $-90^0 \pounds q_i \pounds 90^0$. The drive motors are Servo MG995, Arduino Nano circuit, Logitech B525-720p Camera, Dell Precision M680 Laptop, Razer Seiren Mini Microphone. The actual experimental system is shown in Fig. 8.

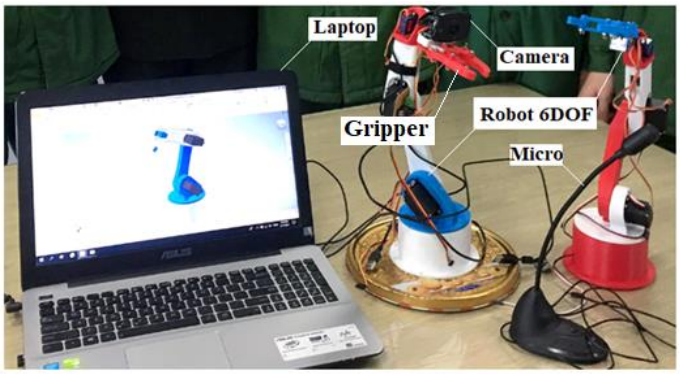


Fig. 8. The actual experimental system

B. Object recognition results

The voices command to perform motion control of the robot (in Vietnamese) is "**Quay bên phải, lấy bánh xe màu vàng**" (This sentence has 8 syllables, each word corresponds to 1 syllable) (This command in English is: "Turn right, grab the yellow wheel" - A total of 7 syllables). After recognizing the voice, the robot starts to move in the requested direction. The next action of the robot arm depends on the analysis results of the object recognition module (yellow wheel). To conduct the yellow wheel identification experiment in 3 objects including a yellow wheel, a green tray, and a red box. The image is divided into 7×7 cells. The identification algorithm uses 2 boundary boxes ($n_{box} = 2$) predictions for each cell. The number of objects to be classified is 3 ($n_{class} = 3$). The number of each object image in the library for training the DL2 model is 140 pieces. Each square is a vector with 13 parameters.

The input is the resized image ($448, 448, 3$ - RGB image). Experiencing the structure of the model with 6 *Maxpooling* layers with the size 2×2, the size of the image decreased by 64 times. Input image size is 448×448, output image size is 7×7. The last layer has 13 nodes so the output of the network will be $7 \times 7 \times 13$, predicting the required photo information. The structure of YOLO is described in Tab. 2. YOLO programming code in PYTHON is shown in Fig. 9.

TABLE II. THE STRUCTURE OF YOLO

Name layer	Filters	Output dimension
Input(448x448x3)		
Conv 1	7x7x192	448x448x192
Max pool 1	2x2x192	224x224x192
Conv 2	3x3x256	224x224x256
Max pool 2	2x2x256	112x112x256
Conv 3	3x3x512	112x112x512
Max pool 3	2x2x512	56x56x512
Conv 4	3x3x1024	56x56x1024
Max pool 4	2x2x1024	28x28x1024
Conv 5	3x3x1024	28x28x1024
Max pool 5	2x2x1024	14x14x1024
Conv 6	3x3x1024	14x14x1024
Max pool 6	2x2x1024	7x7x1024
Conv 7	1x1x13	7x7x13

```
model=Sequential()
model.add(Conv2D(192,kernel=(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(256,kernel=(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(512,kernel=(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(1024,kernel=(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(1024,kernel=(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(1024,kernel=(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Conv2D(13,kernel=(1,1),activation='relu'))

model.compile(loss =Loss_total, optimizer=opt,metrics=['accuracy'])
model.fit(data,index_data,epochs=2000,batch_size=1200)
```

Fig. 9. YOLO programming code

The training was performed for 8.27 hours on GeForce RTX 3080 GPU. The limit for training times is 3000. The results of the training after 84 epochs are shown in Fig. 10 and Fig 11.

```
2086, iou_loss = 0.005280, total_loss = 138.737366
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 37 Avg (IOU: 0.267952), count: 1, class_loss = 641.517517, iou_loss = 0.051575, total_loss = 641.569092
 total_bbox = 472, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 30 Avg (IOU: 0.445260), count: 4, class_loss = 138.206223, iou_loss = 0.326462, total_loss = 138.532684
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 37 Avg (IOU: 0.000000), count: 1, class_loss = 644.164612, iou_loss = 0.000000, total_loss = 644.164612
 total_bbox = 476, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 30 Avg (IOU: 0.144940), count: 4, class_loss = 139.925140, iou_loss = 0.003479, total_loss = 139.928619
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 37 Avg (IOU: 0.000000), count: 1, class_loss = 642.781128, iou_loss = 0.000000, total_loss = 642.781128
 total_bbox = 480, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 30 Avg (IOU: 0.222676), count: 4, class_loss = 139.950134, iou_loss = 0.005371, total_loss = 139.955505
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 37 Avg (IOU: 0.000000), count: 1, class_loss = 644.298706, iou_loss = 0.000000, total_loss = 644.298706
 total_bbox = 484, rewritten_bbox = 0.000000 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 30 Avg (IOU: 0.380493), count: 4, class_loss = 139.007950, iou_loss = 0.131760, total_loss = 139.139709
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 37 Avg (IOU: 0.000000), count: 1, class_loss = 641.923523, iou_loss = 0.000000, total_loss = 641.923523
 total_bbox = 488, rewritten_bbox = 0.000000 %

 Tensor Cores are disabled until the first 3000 iterations are reached.

 7: 391.519226, 391.749146 avg loss, 0.000000 rate, 1.332000 seconds, 420 images, 8.270124 hours left
```

Fig. 10. Model DL2 training process

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 02, February-2021

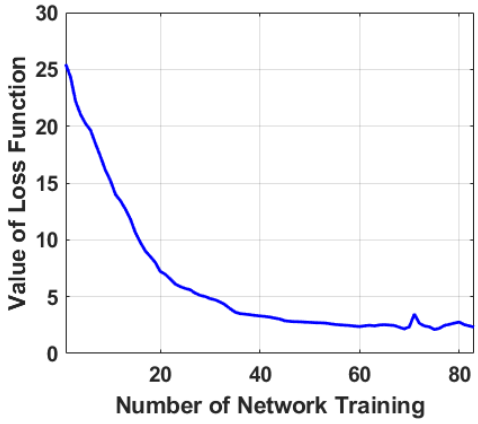


Fig. 11. Value of Loss Function

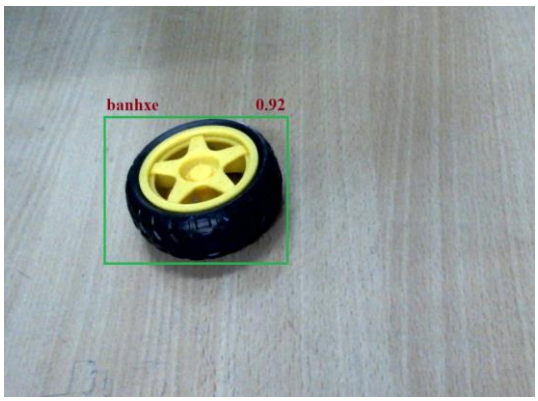


Fig. 12. The object is recognized with the trained DL2 model

The model distinguished the object with an accuracy of 92%.

C. Control result

Network parameter DL3 controlling the robot is shown in Fig. 13 with 5 outputs corresponding to 5 rotation angles of the robot joints. The network consists of 9 hidden layers with the *Relu* activation function. The number of nodes per layer is presented in Fig. 14.

```
model=Sequential()
model.add(Dense(256,activation='relu'))
model.add(Dense(256,activation='relu'))
model.add(Dense(256,activation='relu'))
model.add(Dense(340,activation='relu'))
model.add(Dense(340,activation='relu'))
model.add(Dense(350,activation='relu'))
model.add(Dense(350,activation='relu'))
model.add(Dense(350,activation='relu'))
model.add(Dense(200,activation='relu'))
model.add(Dense(5))

model.compile(loss = 'mean_absolute_error', optimizer='Adam',metrics=['accuracy'])
model.fit(X,Y,epochs=2000,batch_size=40)
```

Fig. 13. Network parameter DL3 for control the robot

Check on test data with input as the position vector of the end-effector point in the workspace is $\mathbf{x} = \begin{bmatrix} 0 & 20 & 0 \end{bmatrix}^T (mm)$, the output of the test data corresponds to the joint variable value. The $\mathbf{q}$ value obtained from the model is $\mathbf{q} = \begin{bmatrix} 90 & 50 & 105 & 90 & 79 \end{bmatrix}^T$ (deg). Thus, the accuracy is 98.67% on the test data set.

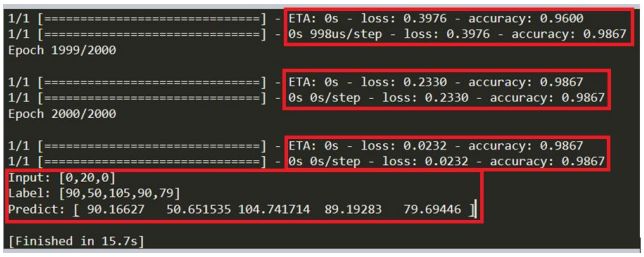


Fig. 14. Training results and prediction on the test dataset

The joint variable values to control the robot arm to the position with the object (yellow wheel) are shown in Fig. 15. Some positions of the robot arm during command execution are shown in Fig. 16.

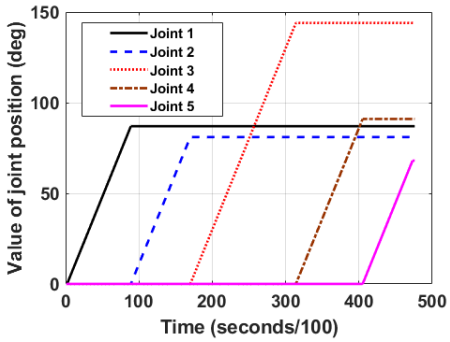
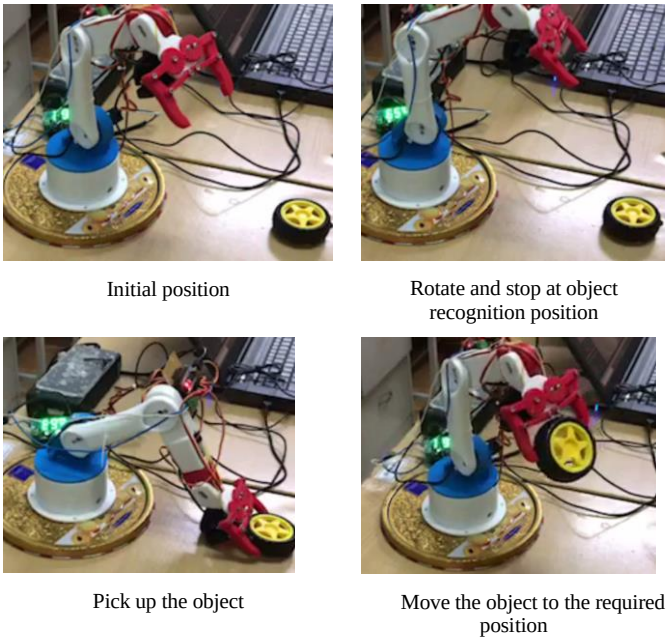


Fig. 15. The joints values are received by the voice command



Initial position

Rotate and stop at object recognition position

Pick up the object

Move the object to the required position

Fig. 16. Some positions of the robot arm

IV. CONCLUSION

The 6DOF robotic arm control system has been built and tested by an experimental model based on computer vision and voice recognition techniques. Accordingly, the voice recognition module through building the DL1 network model with a data library in Vietnamese has been completed. DL2 network model is built to recognize objects through data from the available photo library and images transferred from the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 02, February-2021

Camera. The calculation of control signals transmitted to the robot's drive motors is performed by the DL3 model based on information about the motion direction and position of the recognized object in the workspace. The robotic arm control system based on computer vision is completely independent of the Internet connection. The results of experiments on the actual model show the correctness and reliability of computer vision techniques with a recognition accuracy of 92%. However, this study has not considered the evaluation of the robot arm's motion accuracy due to fabrication errors, the processing rate of the control system in real-time. The research results have an important meaning in the research and development of advanced intelligent algorithms combining speech recognition, computer vision, and the integration of IoT systems.

REFERENCES

[1] S. Zhen, Z. Zhao, X. Liu, F. Chen, H. Zhao, Y. Chen, “A Novel Practical Robust Control Inheriting PID for SCARA Robot”, IEEE Access, vol. 8, pp. 227409-227419, 2020.

[2] D. Nicolis, F. Allevi, P. Rocco, “Operational Space Model Predictive Sliding Mode Control for Redundant Manipulators”, IEEE Transaction on Robotics, pp. 1-8, 2020.

[3] C. Pezzato, R. Ferrari, C. H. Corbato, “A Novel Adaptive Controller for Robot Manipulators, Based on Active Inference”, IEEE Robotics and Automation Letters, vol. 5 (2), pp. 2973-2980, 2020.

[4] M. T. Ziabari, A. R. Sahab, V. Afsar, “Stability in A Flexible Manipulator Using Optimal Nonlinear Controller”, Journal of Basic and Applied Scientific Research, vol. 3(2), pp. 323-329, 2013.

[5] T. Zebin, M. S. Alam, “Dynamic modeling and fuzzy logic control of a two-link flexible manipulator using genetic optimization techniques”, Journal of Computers, vol. 7(3), 578-585, 2012.

[6] C. Hwang, W. Yu, “Tracking and Cooperative Designs of Robot Manipulators Using Adaptive Fixed-Time Fault-Tolerant Constraint Control”, IEEE Access, vol. 8, pp. 56415-56428, 2020.

[7] M. Hwang, B. Thananjeyan, S. Paradis, D. Seita, J. Ichnowski, D. Fer, T. Low, K. Goldberg, “Efficiently Calibrating Cable-Driven Surgical Robots with RGBD Fiducial Sensing and Recurrent Neural Networks”, IEEE Robotics and Automation Letters, vol. 5(4), pp. 5937 - 5944, 2020.

[8] H. Huang, C. Chuang, “Artificial Bee Colony Optimization Algorithm Incorporated with Fuzzy Theory for Real-Time Machine Learning Control of Articulated Robotic Manipulators”, IEEE Access, vol. 8, pp. 192481-192492, 2020.

[9] R. Liu, Q. Zhang, Y. Chen, J. Wang, L. Yang, “A Biologically Constrained Cerebellar Model with Reinforcement Learning for Robotic Limb Control”, IEEE Access, vol. 8, pp. 222199-222210, 2020.

[10] J. Luo, E. Solowjow, C. Wen, J. A. Ojea, A. M. Agogino, “Deep Reinforcement Learning for Robotic Assembly of Mixed Deformable and Rigid Objects”, International Conference on Intelligent Robots and Systems (IROS), pp. 2062-2069, Madrid, Spain, October 1-5, 2018.

[11] D. P. Mital, G. W. Leng, “A Voice-activated Robot with Artificial Intelligence, Robotics and Autonomous Systems”, vol. 4, pp. 339-344, 1989.

[12] S. Hwang, Y. Park, Y. S. Park, “Sound direction estimation using an artificial ear for robots”, Robotics and Autonomous Systems, vol. 59, pp. 208-217, 2011.

[13] M. Buyukyilmaz, A. O. Cibikdiken, “Voice Gender Recognition Using Deep Learning”, Advances in Computer Science Research, vol. 58, pp. 409-411, 2017.

[14] S. Sachdeva, J. Macwana, C. Patela, N. Doshia, “Voice-Controlled Autonomous Vehicle Using IoT”, 3rd International Workshop on Recent Advances on the Internet of Things: Technology and Application Approaches (IoT-T&A 2019), vol. 160, pp. 712-717, November 4-7, Coimbra, Portugal, 2019.

[15] D. Erhan, C. Szegedy, A. Toshev, D. Anguelov, Google Inc, "Scalable Object Detection using Deep Neural Networks", IEEE Conference on Computer Vision and Pattern Recognition, pp. 2155-2162, 2014. Doi: 10.1109/CVPR.2014.276.

[16] H. Schulz, S. Behnke, "Structured Prediction for Object Detection in Deep Neural Networks", 24th International Conference on Artificial Neural Networks (ICANN), Hamburg, Germany, September 2014, pp. 1-8, 2014.

[17] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", IEEE Conference on Computer Vision and Pattern Recognition, pp. 779-788, 2016. DOI: DOI 10.1109/CVPR.2016.91.

[18] J. Redmon, A. Farhadi, "YOLO9000: Better, Faster, Stronger", IEEE Conference on Computer Vision and Pattern Recognition, pp. 6517-6525, 2017. DOI: 10.1109/CVPR.2017.690.

[19] B. İşçimen, H. Atasoy, Y. Kutlu, S. Yıldırım, E. Yıldırım, "Smart Robot Arm Motion Using Computer Vision", Elektronika Ir Elektrotechnika, 2015. DOI: 10.5755/j01.eee.21.6.13749.

[20] A. Saxena, J. Driemeyer, and A. Y. Ng, “Robotic grasping of novel objects using vision”, *The International Journal of Robotics Research*, vol. 27, no 2, pp. 157-173, 2008.

[21] J. Wei, H. Liu, G. Yan, F. Sun, "Robotic grasping recognition using multi-modal deep extreme learning machine", Multidim Syst Sign Process, pp. 1-17, 2016. DOI: 10.1007/s11045-016-0389-0.

[22] A. R. Pathaka, M. Pandeya, S. Rautaray, "Application of Deep Learning for Object Detection", International Conference on Computational Intelligence and Data Science, pp. 1706–1717, 2018.

[23] X. Chen, B. Zhao, Y. Wang, X. Gao, "Combination of high-frequency SSVEP-based BCI and computer vision for controlling a robotic arm", Journal of Neural Engineering, pp. 1-12, 2018.

[24] X. Chen, X. Huang, Y. Wang, X. Gao, "Combination of augmented Reality-Based Brain-Computer Interface and Computer Vision for High-Level Control of Robotic Arm", Transactions on Neural Systems and Rehabilitation Engineering, 2020. DOI: 10.1109/TNSRE.2020.3038209.

[25] X. Chen, B. Zhao, Y. Wang, and X. Gao, “Combination of high frequency SSVEP-based BCI and computer vision for controlling a robotic arm,” *J. Neural Eng.*, vol. 16, no. 2, p. 026012, 2019.

[26] Website: https://deepsystems.ai/ (Access in Feb 27, 2021).

[27] Website: https://pjreddie.com/darknet/yolo/ (Access in Feb 27, 2021).

[28] Website: https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/.

[29] M. W. Spong, S. Hutchinson, M. Vidyasagar, Robot modeling and Control, First edition, New York, USA, 2001.