SSRG International Journal of Computer Science and Engineering
ISSN: 2348 – 8387 /doi:10.14445/23488387/IJCSE-V8I1P105
Volume 8 Issue 1, 24-29, January 2021

Researching Robot Arm Control System Based On Computer Vision Application And Artificial Intelligence Technology

Hoang Thi Phuong

[#]*University of Economics - Technology for Industries, Viet Nam*

Abstract *— One of the most popular robots in the manufacturing world is the robotic arm. In most cases, robotic arms are programmed and used to perform specific tasks, most common for manufacturing, fabrication, and industrial applications. This article presents a robotic arm control system by recognizing hand gestures from the operator. The system is based on three main steps: locate the hand gesture on the received image, determine the outline of the hand gesture, and recognize this gesture using neural networks and deep learning technology. The use of a region of interest extraction and contour detection reduces computation volume, thereby speeding up hand gesture recognition, making it possible for the robot arm to perform real-time operations. The experimental results show the positive effect of the proposed method.*

Keywords *— Artificial intelligence, Deep learning, Robot arm, Computer vision, Edge detection.*

I. INTRODUCTION

Computer vision refers to the entire process of simulating human vision in a non-biological machine. This includes primary photography, subject detection and recognition, temporary scene recognition between scenes, and developing a high level of understanding of what is happening at the appropriate intervals. This technology has long become ubiquitous in science fiction, and as such, it is often taken as a matter of fact. A system that provides reliable, accurate and real-time computer vision is a challenging issue that has yet to be fully developed. As these systems mature, there will be many applications that rely on computer vision as a major component. Typical examples are self-driving cars, autonomous robots, drones, intelligent medical imaging devices that aid surgery, and surgical implants for restoring human vision. Although computer vision holds great promise in the future, it carries an inherent complexity and is always a challenge for computer systems. Part of the complexity is because computer vision is not a single task. Instead, it is a series of non-simple tasks that require the use of complex algorithms and sufficient computational power to operate in real-time. At a high level, computer vision's side tasks are object detection and segmentation, image classification, object tracking, image labeling with meaningful descriptions (e.g., annotations image), and finally, understand the whole scene's meaning.

While there are still significant obstacles in the evolutionary path of computer vision to the 'human-level,' Deep Learning systems have made significant progress in dealing with a number of related side tasks. Concerned. The reason for this success is in part based on the additional responsibility assigned to deep learning systems. It's plausible to say that the biggest difference with deep learning systems is that they no longer need to be programmed to look for specific features. Instead of searching for specific features using carefully programmed algorithms, the neural networks inside deep learning systems are trained. For example, if the car in the image is misclassified as a motorcycle, don't tweak the parameters or rewrite the algorithm. Instead, you keep training until the system gets it right. With the increased computing power provided by modern deep learning systems, there is steady and remarkable progress towards the point where a computer will recognize and react to everything it sees. See.

This article refers to a robotic arm system that integrates a motion control system with control through hand gestures. These systems are integrated with the function of coordinate analysis, real-time processing to increase the system's efficiency. The method chosen and deployed is to capture and detect interest regions in the frame, done using the Point Feature Matching technique. In addition, we also incorporated noise reduction during image acquisition using and compared four image filtering techniques: Canny, Sobel, Prewitt, and Roberts. The final step is to perform image classification by Artificial Intelligence technology, including a Convolutional Neural Network (CNN). This method ensures identifying all images in the frame, satisfying the assumptions made in robot simulation. In addition, a structure has been developed that allows the robotic arm to maintain or change the formation of defined trajectories and perform individual manipulation tasks. This paper is divided into 5 parts, in which part 2 presents the problem discussed and the solutions implemented. This problem is solved by the algorithm introduced in part 3 and statistical methods to verify the reliability. The results after applying the proposal are presented in section 4 and conclusions in section 5.

II. RELATED WORK

It is quite common to help computers recognize and understand human body language, thereby controlling robot components, this technique mentioned and used by authors in the article [1]. The author converts the RGB image to YCrCb format to recognize the predefined hand gestures. The serial signals received from the computer after the direct video processing are sent to the Atmega2560 microcontroller. A Zigbee module is mounted on the robot arm and the microcontroller to send serial and wireless signals. Depending on the signal received, the corresponding actions are performed by a robotic arm. Saraiva [2] presents in his paper a method to control a robot, using hand gestures, in which a CNN-like neural network recognizes gestures from images taken with a camera mounted. At a fixed position. The author says that this approach is powerful in real-time commands recognition and execution. In Rotary's study [3], the author has performed a comparative analysis of the use of hand gestures as human-machine interaction. The author says the use of hand gestures provides an attractive and natural alternative to computer-human interaction. In his article, the author has studied more than two hundred and fifty related publications. Wen [4] proposes a collaborative surgical robot system, guided by hand gestures and powered by an augmented reality system.

Mobile surgical robot system performing minor surgery. Natural hand gestures are an intuitive and powerful method to interact with both surgical systems and robots. Gesture recognition is also used to navigate quadrupedal robots, such as in [5], in which hand image segmentation is performed, coming from a fixed position camera. The author uses Threshold Segmentation, Continuously Adaptive Mean-Shift, and Restricted Boltzmann Machines to classify gestures in real-time, thereby giving control commands. For quadruped robots. In the study conducted by Parada [6], the authors used hand gesture recognition technology in automobile control, and the authors created an interface system that allows the use of the devices. Automatically without distraction and thereby reducing the number of traffic accidents related to distraction while driving. This system works with an infrared camera mounted on the car that recognizes the operator's hand gesture and provides a consistent response. In the article [7], the author Gupta also uses hand gesture recognition techniques for intuitive car interfaces. In this paper, the authors point out that hand gestures performed manually on the wheel steering wheel or close to it lead to low physical distraction.

Computer vision provides the basis for applications that use automatic image analysis. Computers are preprogrammed in most applications that use the computer's vision to perform a particular task [10-15]. There are many studies integrating computer vision with robotic arms in the literature. One of these works presents a learning algorithm that tries to determine points from two or more given images of an object to grasp the object with a robotic arm [15]. The algorithm performs with 87.8% overall accuracy to capture new objects. In another study, computer vision was used to control a robotic arm [16]. In two other studies, robot models were designed to play the game "rock paper scissors" against opponents [17], [18]. In another work, robotic arm movements are controlled according to human arm movements using wireless connections and vision systems [19]. There are a number of other studies that cover autonomous subject detection and capture tasks. One of these studies presented an automated robotic framework consisting of a visual system [20-25]. In its work, the robotic arm can perform the task of self-classifying an object according to the shape, size, and color.

This paper has developed a method to move a robotic arm with hand gestures in real-time from the above studies. This method will be detailed in the next section.

III. THE ALGORITHM CONTROLS THE ROBOTIC ARM THROUGH THE IMAGE

In this section, the structure of the system is introduced. The first step is to collect and store hand gestures, which are then analyzed by the algorithm and classified by artificial neural networks.

A. System Structure

The system uses a webcam to record and send images to the computer system. The recognition method will then be based on a sequence of steps that will enable real-time tracking. These steps start with algorithms to establish a region of interest, detect edges, and compare features identified in categories. Then, the system will send a signal to control the robot arm to perform certain actions based on this gesture from the detected gesture. A number of algorithms are also developed to support system reliability. The proposed method can be observed in (Figure 1), while the algorithm used is shown in Algorithm 1.

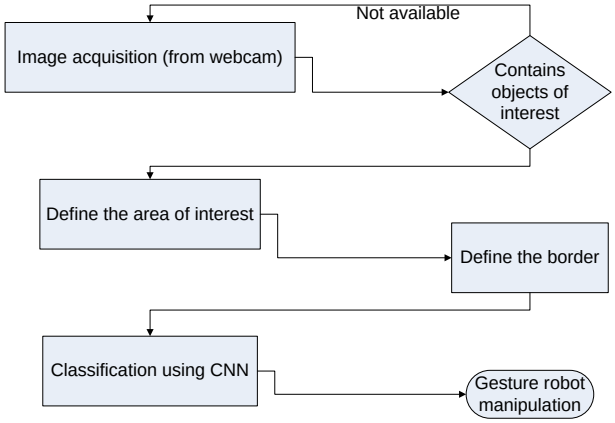

Figure 1: Schematic structure of the system

Algorithm 1: Pseudocode pseudo-code of the method

Data*: Images from webcam*

Result*: The robot works according to the operator's*

gesture

***Begin**:*

***Read** image reference*

image reference;

feature point = feature point extraction (reference image)

***Calling** CNN artificial neural networks;*

list of gestures = List ('g1', 'g2', 'g3', 'g4', 'g5', 'g6', 'g7', 'g8');

***While loop**: executes until the user ends*

current image = Picture from webcam

current image feature point = extract feature point (current image)

***If** compare feature point (feature point, present image feature point) match*

region of interest = define region of interest ();

the borderline of interest = define contour (area of interest, 'method');

prediction = prediction (region of interest contour, CNN neural network);

If the gesture list contains predictions

Robot manipulation (prediction);

End If

End If

End of the Loop

B. Identify Areas of Interest on the Photo

In this part of the article, we introduce the method used to recognize hand gestures from images received from the webcam. This method is programmed in testing using OpenCV and Node.js. We first process the image to create a binary mask of the hand to compute the hand contour. Then segment the image based on the hand skin color using the step manipulation. This will convert the frames from the default BGR format in OpenCV to the HLS color space (Hue, Lightness, Saturation). The Hue channel encodes the actual color information. This way, we have to find out the appropriate range of Hue values for the skin and then adjust the values for Saturation and Lightness. Then use the OpenCV functions to find the contours of the binary mask of the hand. An example of this algorithm is depicted in Figure 2.

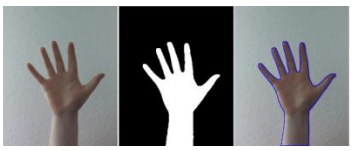

Figure 2: Algorithm of determining region of interest, binary mask, and contour of hand gestures

For this paper's purposes, we use four types of contour detection techniques: Sobel, Roberts, Prewitt, and Canny [8]. These algorithms' filters are shown in (Figures 3, 4, 5) and are all integrated into OpenCV. Sobel used two masks of size [3 x 3] where one mask rotates the other at a 90-degree angle, as shown in figure 3 [9]. These masks are designed to find vertical and horizontal boundaries best. When doing the convolution between the image and the masks, we get vertical and horizontal gradients Gx, Gy. The Sobel operator looks like Figure 3.

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Figure 3: Filter to define contour with Sobel operator

The Prewitt method is almost identical to Sobel. This is the oldest, most classical method. The Prewitt operator is shown in Figure 4.

−1	−1	−1
0	0	0
1	1	1

−1	0	1
−1	0	1
−1	0	1

Figure 4: Filter to define contour by operator Prewitt

For the Roberts operator, similar to Sobel, we compute the horizontal and vertical boundaries separately using two masks (Figure 5), then sum them up to make the image's real border. However, since Roberts' mask is quite small, the result is a lot of noise.

0	0	0
0	−1	0
0	0	1

0	0	0
0	0	−1
0	1	0

Figure 5: Filters for defining contours using Roberts operator

Besides the operators introduced above, in this article, we were also using the Canny method. This method uses two levels of high and low thresholds. Initially, we use the high threshold to find the starting point of the boundary, and then we determine the direction of the boundary's development based on consecutive pixels with a value greater than the low threshold. We only discard points with values less than the low threshold. Weak borders will be selected if they are linked with strong borders. To execute the algorithm, 4 main stages are performed on a segmented image:

- Step 1: First, use the Gaussian filter to smooth out the image.
- Step 2: Then compute the gradient of the edge of the smoothed image.

- Step 3: The next step is to remove the non-maxima.
- Step 4: The final step is to remove any values that are less than the threshold level.

C. Robot Arm

The robotic arm has shown above (figure 6) is used, consisting of four degrees of freedom, each with a magnitude of 180 degrees. To control the robot arm, eight hand gestures are used; each hand gesture transmits a certain action to the arm, transmitted from the computer to the actuators after being detected, over the serial protocol. G1 gestures are used to move the first joint from 0 to 180 degrees, which involves rotating the stand, i.e., the arm will rotate 180 degrees to the left of its position. Gesture G2 includes undoing the previous movement so that the first joint will rotate from 180 to 0 degrees. G3 gesture transmits a command to make the second joint move from 0 to 160 degrees; G4 gesture will make it move 160 to 0 degrees, which means G3 and G4 are responsible for the arm's high angles. However, the G5 will be responsible for moving the triple coupling from 0 to 120 degrees, and the G6 switches from 120 to 0 degrees, so the G5 and G6 are responsible for controlling the angle of the arm opening. The G7 and G8 gestures are responsible for controlling the fourth and the last coupling of the arm, making the fourth joint go from 0 to 150 degrees, and vice versa. Thus, the last two gestures G7 and G8, affect the hand part of the robot arm.

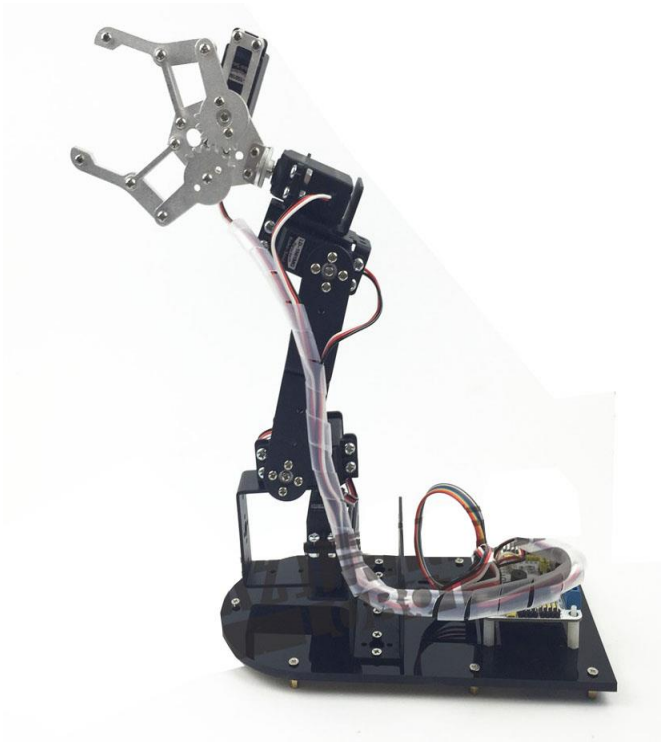

Figure 6: Four degrees of the freedom robot arm, each with a magnitude of 180 degrees

D. Classifier

The final step of the system is to train the classifiers responsible for performing gesture recognition. This paper uses the CNN neural network algorithm because these algorithms have proven superiority in complex problems. Therefore, deep learning technology with artificial neural networks is becoming more and more popular, especially in computer vision. Convolutional Neural Network (CNN) is a variant of the multilayer perceptual network. It is inspired by the biological processing of data visually and comprises many components, with each component a different function. Each class is composed of weighted neurons, and these weights can change during learning. Each neuron receives an input signal, is weighted multiplied, and then filtered through a non-linear function. The entire neural network still has a single goal: from the pixels of the raw image at the input through the computation stage, we will have the score of this image corresponding to the classification classes to the class's score. That. In this article, we used a trained CNN to process and classify images. The CNN architecture used is AlexNet [8]. AlexNet receives an input of 227 x 227 pixels per channel. The first convolution layer uses an 11 x 11 x 3 filter, in the second layer, it is 5 x 5 x 3, and in the third layer, 3 x 3 x 3. In addition, the third, fourth, and fifth layers are connected without using pooling. Finally, the network has two fully connected layers with 2048 neurons per layer and an output layer with 1000 neurons, which is also the number of classification layers.

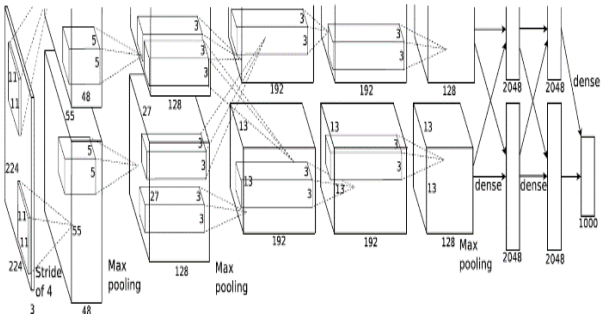

Figure 7: Architecture of the AlexNet artificial neural network

For this work, transfer learning techniques were applied to speed up the training process, using the Alexnet network structure, changing the output layer to eight neurons according to the classified gesture types. Hence we don't have to train all the weights of the network layers. If so, this would be an expensive process, as AlexNet is a weighted model that uses a subset of 1000 ImageNet categories (this is a very popular database of images today). A total of 800 images were used, of which 60 percent (480) of these were used during CNN training, while the remaining 40 percent (320) were used for testing and testing. Check the classifier accuracy. The images used for CNN training and testing are adapted from the control works using common hand gestures shown in Figure 8.

Figure 8: Some hand gestures

E. Classification Agreement

To analyze the quality of the classification, it is necessary to classify this object many times. Like a classification tool, we have a confusion matrix that provides a basis for describing a classification's accuracy and describing errors, helping to refine the classification. From a confusion matrix, several measures can be drawn to calculate the classification's accuracy, and here we use the Kappa index to solve this problem. The confusion matrix is formed by an array of squares arranged in rows and columns representing the number of sample units of a type, the classification derived from the algorithm, and the exact classification data. Typically underneath the columns is the reference data, which is compared with the classification data shown on the rows. The indices obtained from the confusion matrix are overall accuracy, class accuracy, and Kappa number, among others. The total precision is calculated by dividing the sum of the error matrix's main diagonal x_{ii} by the total number of samples collected n. According to equation (1).

$$T = \frac{\sum_{i=1}^{a} x_{ii}}{n} \quad (1)$$

The precision distribution across individual layers is not shown in the total accuracy. However, the accuracy of an individual class can be obtained by dividing the total number of correctly classified samples in that category by the total number of samples. In this paper, the Kappa measure is used to describe the agreement's magnitude, based on the appropriate number of responses. The kappa is the measure of the interobserver agreement and the degree of agreement beyond what can happen. This is a discrete multivariate technique used to evaluate subject accuracy and use all the confusion matrix elements in its computation. The Kappa coefficient (K) is the measure of the actual agreement (denoted by the diagonal elements of the confusion matrix) minus the opportunity agreement (expressed as the sum of the products of the row and column, and no include unrecognized items). The Kappa coefficient can be calculated from Equation 2:

$$K = \frac{n\sum_{i=1}^{a} x_{ii} - \sum_{i=1}^{a} x_{+i}x_{ii}}{n^2 - \sum_{i=1}^{a} x_{+i}x_{ii}} \quad (2)$$

This deal measure has a maximum value of 1, where this value 1 represents the total deal and values close to zero, indicating no deal or a deal that was correctly created. By accident. The Kappa's final value is less than 0, negative, indicating that the deal was found less than expected. Therefore, it shows disagreement. Interpretation of the Kappa values is shown in Table 1.

Table 1: Table of Kappa values

Kappa value	Level agreement
< 0	No agreement
0-0.19	Bad deal
0.20-0.39	Fair agreement
0.40-0.59	Moderate agreement
0.60-0.79	Significant agreement
0.80-1.00	Nearly perfect deal

IV. EXPERIMENTAL RESULTS

In this section, the accuracy of the proposed method is assessed using a standard validation technique, in which the accuracy of CNN is measured using the method presented in Equation 1. Algorithm Performance. CNN's for each of the proposed methods are presented in (Table 2). The learning and training CNN neural network is done on Nvidia Geforce RTX 2080 graphics card, with 3072 CUDA core processor. The same parameters are used on CNN in all edge detection methods. The MaxEpochs parameter (one Epoch corresponding to a complete data completion) is set to 15. The Mini-Batch Size parameter corresponding to the number of observations made at each iteration is set to 80.

Table 2: The accuracy of the training method and time

Methods	Average accuracy	Training time in seconds
Traditional	99.60%	161.30
Prewitt	95.60%	45.59
Roberts	94.00%	39.45
Sobel	94.00%	43.58
Canny	98.10%	44.26

In (Table 2), it can be verified that the original (colored) images give the best accuracy of 99.60% in this experiment. However, the CNN network's training time is about four times higher than the Canny operator, in second place with 98.10% accuracy, followed by the Prewitt, Roberts, and Sobel edge extraction methods, with the shortest processing

time for the Roberts method. In (Table 3), we present our analytical results consistent with the Kappa index, where the K value for each method represents an almost perfect consensus for all the methods used. It can be seen that the Canny method, although slightly lower in accuracy and agreement than the traditional method, has a much lower training time.

Table 3. The value of the Kappa index

Methods	Classification Agreement K
Traditional	0.9915
Prewitt	0.9523
Roberts	0.9571
Sobel	0.9523
Canny	0.9843

V. CONCLUSIONS

This article presents a robotic arm control system from hand gestures, which can replace traditional control via command. In addition, the proposed method accuracy related to hand gesture recognition was shown to reduce about four times the CNN training time by reducing the volume of data through the filter application to receive. Know the contour, with relatively little reduction in accuracy compared to identifying through the original image. The algorithm also offers real-time video processing solutions, making it more efficient to evaluate and convert operator actions into the robot arm's actions. In the future, we plan to increase the number of gestures and images. In addition, we also plan to test other methods of contour determination to improve the method's accuracy.

ACKNOWLEDGMENT

This study was supported by the University of Economics - Technology for Industries, Viet Nam; http://www.uneti.edu.vn/.

REFERENCES

[1] A. A. Saraiva, D.B.S Santos, F.C.F Marques Junior, J. V. M. Sousa, N.M. Fonseca Ferreira, and A. Valente,Navigation of quadruped multi robots by gesture recognition using restricted Boltzmann machines, 21st International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, Panama City, 09(2018) 309-317.

[2] A. Saraiva, R. Melo, V. Filipe, J. Sousa, N.M Fonseca Ferreira, and A. Valente,Mobile multi-robot manipulation by image recognition, International Journal of Systems Applications, Engineering Development, 12(04)(2018) 63-68.

[3] S. S. Rautaray and A. Agrawal,Vision-based hand gesture recognition for human-computer interaction: a survey, Artificial Intelligence Review, 43(1)(2015) 1–54.

[4] R. Wen, W.-L. Tay, B. P. Nguyen, C.-B. Chng, and C.-K. Chui, "Hand gesture guided robot-assisted surgery based on a direct augmented reality interface, Computer methods and programs in biomedicine, 116(2)(2014) 68–80.

[5] G. Choudhary and C. R. BV,Real-time robotic arm control using hand gestures," in High-Performance Computing and Applications (ICHPCA), 2014 International Conference on. IEEE, (2014) 1–3.

[6] F. Parada-Loira, E. Gonz´alez-Agulla, and J. L. Alba-Castro,Hand gestures to control infotainment equipment in cars, Intelligent Vehicles Symposium Proceedings, 2014 IEEE. IEEE, (2014) 1–6.

[7] S. Gupta, P. Molchanov, X. Yang, K. Kim, S. Tyree, and J. Kautz, Towards selecting robust hand gestures for automotive interfaces, in Intelligent Vehicles Symposium (IV), 2016 IEEE. IEEE, (2016) 1350–1357.

[8] Alex Krizhevsky and Sutskever, Ilya and Hinton, Geoffrey E .,ImageNet Classification with Deep Convolutional Neural Networks, Advances in Neural Information Processing Systems 25(2012) 1097-1105.

[9] Nguyen Vinh An,Comparison of Edge Detection Techniques, Vietnam National University Journal of Science: Natural Sciences and Technology, 31(2) (2015) 1-7.

[10] A. D. Kulkarni,Computer vision and fuzzy-neural systems, Prentice Hall PTR, (2001), ch. 2 and ch. 6.

[11] R. Jain, R. Kasturi, and B. G. Schunck.,Machine Vision, McGraw-Hill (1995), ch. 14.

[12] D. A. Forsyth and J. Ponce, "Computer vision: a modern approach," Prentice Hall Professional Technical Reference, 2002, ch. 15.

[13] G. Bradski, A. Kaehler and V. Pisarevsky, Learning-based computer vision with Intel's open-source computer vision library,Intel Technology Journal, 9,(2005).

[14] C. H. Lampert, H. Nickisch, and S. Harmeling,Learning to detect unseen object classes by between-class attribute transfer, IEEE Computer Vision and Pattern Recognition, (2009) 951-958,.

[15] A. Saxena, J. Driemeyer, and A. Y. Ng,Robotic grasping of novel objects using vision, The International Journal of Robotics Research, 27(2)(2008) 157-173.

[16] R. Szabó and A. Gontean,Full 3D Robotic Arm Control with Stereo Cameras Made in LabVIEW, Federated Conference on Computer Science and Information Systems (FedCSIS), (2013) 37-42.

[17] Y. Hasuda, S. Tshibashi, H. Kozuka, H. Okano and J. Ishikawa,A robot designed to play the game Rock, Paper, Scissors,IEEE Industrial Electronics, (2007) 2065-2070.

[18] Ishikawa Watanabe Lab., University of Tokyo www.k2.t.u tokyo.ac.jp/fusion/Janken/index-e.html.

[19] A. Shaikh, G. Khaladkar, R. Jage, T. Pathak and J. Taili,Robotic arm movements wirelessly synchronized with human arm movements using real-time image processing, IEEE India Educators., Conference (TIIEC), Texas Instruments, (2013) 277-284.

[20] S. Manzoor, R. U. Islam, A. Khalid, A. Samad, and J. Iqbal, "An opensource multi-DOF articulated robotic educational platform for autonomous object manipulation, Robotics and Computer-Integrated Manufacturing, 30(3)(2014) 351-362,.

[21] N. Rai, B. Rai, and P. Rai.,Computer vision approach for controlling educational robotic arm based on object properties, IEEE Emerging Technology Trends in Electronics, Communication, and Networking (ET2ECN), 2nd International Conference, (2014) 1-9.

[22] T. P. Cabré, M. T. Cairol, D. F. Calafell, M. T. Ribes and J. P. Roca, "Project-Based Learning Example: Controlling an Educational Robotic Arm With Computer Vision, Tecnologias del Aprendizaje, IEEE Revista Iberoamericana de, 8(3)(2013) 135-142.

[23] Y. Kutlu, M. Kuntalp and D. Kuntalp,Optimizing the performance of an MLP classifier for the automatic detection of epileptic spikes, Expert Systems with Applications,36(4) (2009).

[24] C. M. Bishop,Neural networks for pattern recognition," Clarendon Press, (1995), ch. 4.

[25] B. Iscimen, H. Atasoy, Y. Kutlu, S. Yildirim, E. Yildirim,Bilgisayar Gormesi Ve Gradyan Inis Algoritmasi Kullanilarak Robot Kol Uygulamasi, Akilli Sistemlerde Yenilikler ve Uygulamalari (ASYU) Sempozyumu, (2014) 136-140.