

Smart Robot Arm Motion Using Computer Vision

Bilal Iscimen¹, Huseyin Atasoy¹, Yakup Kutlu¹, Serdar Yildirim¹, Esen Yildirim¹

¹*Department of Computer Engineering, Iskenderun Technical University,
Iskenderun/Hatay, Turkey
bilaliscimen@gmail.com*

Abstract—In this study computer vision and robot arm are used together to design a smart robot arm system which can identify objects from images automatically and perform given tasks. A serving robot application, in which specific tableware can be identified and lifted from a table, is presented in this work. A new database was created by using images of objects used in serving a meal. This study consists of two phases: First phase includes recognition of the objects through computer vision algorithms and determining the specified objects' coordinates. Second phase is the realization of the robot arm's movement to the given coordinates. Artificial neural network is used for object recognition in this system. 98.30 % overall accuracy of recognition is achieved. Robot arm's joint angles were calculated by using coordinate dictionary for moving the arm to desired coordinates and the robot arm's movement was performed.

Index Terms—Classification, computer vision, robot arm, robot programming.

I. INTRODUCTION

Extracting meaningful information from images is one of the interests of the computer vision field. The primary objective is duplicating the human's vision abilities on electronic environment by applying methods on images for processing, analysing and extracting information. Image understanding can be described as extracting symbolic or numeric information from images by using methods constructed with geometry, physics and statistics [1]–[3].

Computer vision provides basis for applications that use automated image analysis. Computers are preprogrammed in most applications that make use of computer vision to perform a specific task. Recently, learning based methods are also commonly used for that kind of applications [4]–[6]. Controlling processes, navigation, detecting events, modelling objects or environments are examples of computer vision based applications.

One of the applications of computer vision is to determine if any object or activity exists in a given image. The problem gets complicated as the number and type of objects with random location, scale and position increase. Some of the most successfully performed computer vision tasks in well-defined illumination, background and camera angle are, recognizing simple geometric objects, analysing printed or hand-written characters, identifying human faces or

fingerprints. In this study, a smart robot arm system is designed to detect and identify randomly placed, in location and orientation, cutlery and plates on a table.

There are many studies integrate computer vision with robot arm in literature. One of these works presents a learning algorithm which attempts to identify points from given two or more images of an object to grasp the object by robot arm [6]. The algorithm performed with 87.8 % overall accuracy for grasping novel objects. In another study, computer vision was used to control a robot arm [7]. Some coloured bottle stoppers were placed on joints' of the robot arm. Therefore, the joints were recognized via these stoppers using image recognition algorithms. The robot arm was simulated by detected joints in computer and 3D arm control was performed by using stereo cameras. In two other studies robot models were designed to play the game "rock, paper, scissors" against an opponent [8], [9]. In both studies, a fixated camera was used to get images of opponent's hand to determine the played move via computer vision algorithms. In one of the studies, the robot has played a random move [8]. But in the other study robot recognizes the opponent's hand shape rapidly using computer vision algorithm and shapes the robot's fingers such that it can beat the opponent's move [9]. In another work, the movements of a robot arm are controlled according to a human arm's movements using wireless connection and a vision system [10]. Two cameras, having their planes perpendicular to each other, capture the images of the arm's movements through the red coloured wrist. The arm's coordinates are transmitted in binary format through a wireless RF transmitter. The robot arm's movements are synchronized using the received coordinates according to the human arm's position and orientation.

There are some other studies including autonomous object detecting and grasping tasks. One of these studies presents an autonomous robotic framework including a vision system [11]. In their work, the robot arm can perform the task of autonomous object sorting according to the shape, size and colour of the object. In [12], randomly placed coloured objects on a target surface and coloured gripper of the vision based controlled educational robotic arm are detected and the objects are moved to a predefined destination using two on-board cameras. Centre-of-Mass based computation, filtering and colour segmentation algorithm are used in order to locate the target and the position of the robotic arm. In

another work, an educational robotic arm performs the task of detecting a randomly placed object, picking it and moving it to a predefined container using a vision system [13]. A light blue foam-rubber cube is randomly placed on a target area which is surrounded by black lines. A fixed zenithal camera provides an image of target area which includes coloured robot grippers and coloured object. Grippers and the object are detected using computer vision algorithms and the object is moved to the container whose position is predefined and fixed.

In this study, a smart robot arm system is designed to detect and identify cutlery and plates and grasp the objects without colouring the objects. An image of objects is taken through a camera. All objects in the image are identified using image processing methods and all detected objects' coordinates are determined on the computer and sent to the robot arm. Afterwards, the robot arm joints' angles are calculated according to received coordinates and the robot arm moves the objects and lifts them in the order they are detected.

II. MATERIALS AND METHODS

Proposed system consists of two phases: recognizing the objects and constructing the movement. In the first phase; a database of cutlery and plate images is constructed, pre-processing, feature extraction, classification and determining the coordinates of the detected objects steps are achieved. In the second phase, the robot arm receives the coordinates and moves towards the object. Figure 1 and Fig. 2 show the steps of first and second phases respectively. Details of these steps are described in the following subsections.

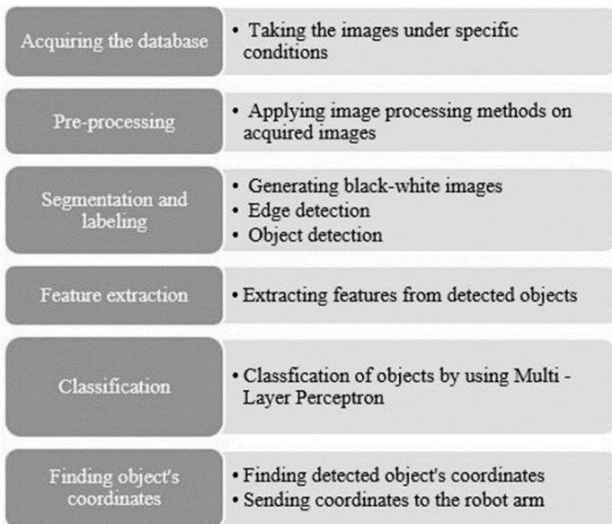


Fig. 1. Steps of the first phase: Application of computer vision algorithms.

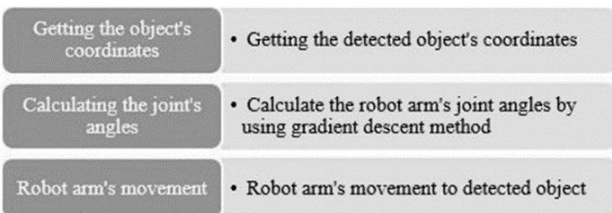


Fig. 2. Steps of the second phase: Movement of the robot arm.

A. Acquiring the Database

Two databases are acquired for training and test purposes.

1) Training Database

This database includes separate images for each object. The distribution of the images according to the objects is given in Table I. In each image, one object is located on a dark background floor with different positions and the images are taken from different distances. Sample images from the train database are given in Fig. 3.

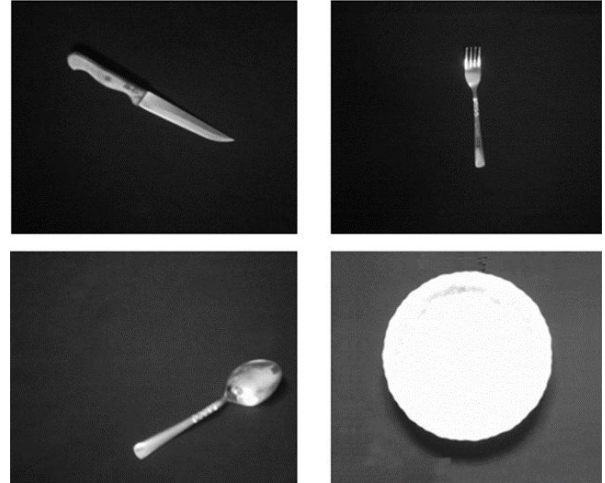


Fig. 3. Sample images from train database.

TABLE I. NUMBER OF UTENSILS IN TRAINING DATABASE.

Object	Count of objects
Knife	38
Fork	65
Spoon	69
Fruit knife	75
Oval plate	48
Total number of objects	295

2) Test Database

For the test purposes we constructed a database that includes 153 images, including randomly selected utensils that placed on a dark background each having random positions. Sample images from test database are given in Fig. 4. Total number of utensils in test images are shown on Table II.

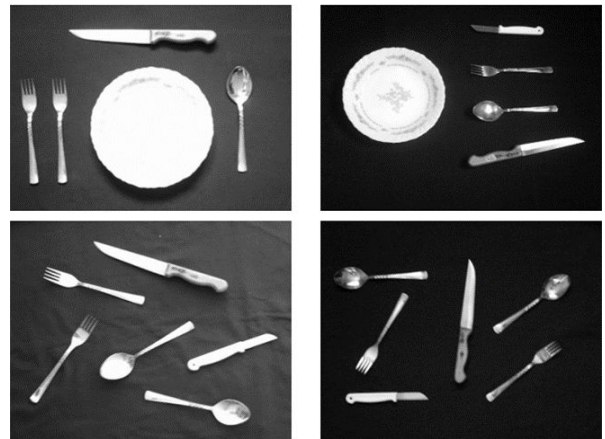


Fig. 4. Sample images from test database.

TABLE II. NUMBER OF UTENSILS IN TEST DATABASE.

Object	Count of objects
Knife	101
Fork	208
Spoon	199
Fruit knife	161
Oval plate	93
Total number of objects	762

B. Object Detection and Feature Extraction

Image processing methods are applied on acquired images and objects are detected. The following steps are performed for this task:

- The taken image was resized.
- The coloured input image was converted to a grayscale image.
- Sobel Filter was used for edge detection.
- Image was filtered by a row matrix shaped morphometric structure element in order to fix edge disorders and make the edge apparent.
- Overflowing or missing pixel issues were fixed by erosion and dilation processes.
- Inner sides of edges were filled in order to detect the whole apparent area of object.

11 features were extracted for each object using MATLAB. The extracted features are area, major axis length, minor axis length, eccentricity, orientation, convex area, filled area, Euler number, equivalent diameter, extent and solidity of the detected image. All features are divided by the perimeter value of the object for normalization purposes.

C. Image Classification

Artificial Neural Networks (ANN) are used for classification [14]. ANN includes units that correspond to neurons of the biological neural network. There are input and output layers in an ANN with adjustable weights and each neuron unit of these layers produces an output value which is calculated via a function of the sum of its inputs [14], [15]. The output value of each neuron is calculated as

$$y_i = f\left(\sum x_i w_i\right), \quad (1)$$

where y_i represents the output, $f(\cdot)$ refers to activation function, w_i refers to weight and x_i refers to input of the i th unit.

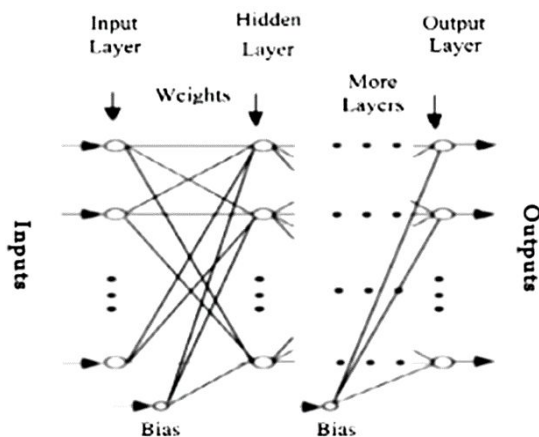


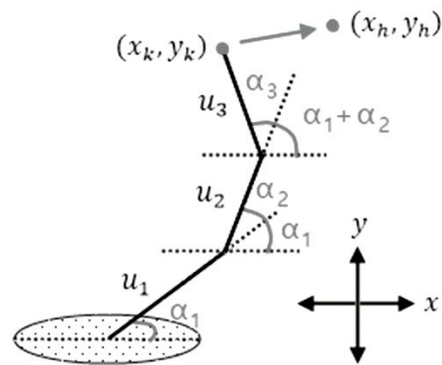
Fig. 5. Architecture of the MLP network.

Multi-Layer Perceptron (MLP) is one of the mostly used structures of ANNs. MLP consist of various number of hidden layers with different number of units besides input and output layers. The first layer receives the inputs from outside and transmits to hidden layers. Hidden layers process the data in their turns and transmit to the output layer. Figure 5 shows the basic architecture of a MLP network [14].

D. Joint's Angle Calculation and Robot Arm's Movement

After the classification process, gravitational centres of forks, knives and spoons and plates were determined as targets of the robot arm. Angles of the joints were calculated on two 2-dimensional planes; x-y and x-z.

In this study a coordinate dictionary was created by generating x and y coordinates using (2) and (3) with respect to joint angles.

Fig. 6. Bone lengths (u) and joint angles (α) on the x-y plane.

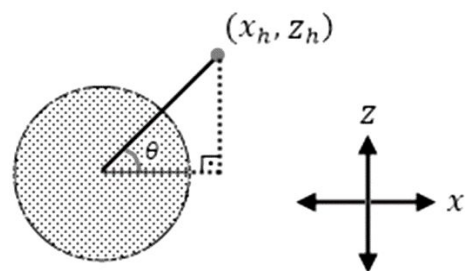
$$x_k = \sum_i u_i \cos\left(\sum_{j=1}^i \alpha_j\right), \quad (2)$$

$$y_k = \sum_i u_i \sin\left(\sum_{j=1}^i \alpha_j\right). \quad (3)$$

x_k and y_k values were sampled for all possible triple combinations of three angles that take values between $[0, f]$, $[-f, 0]$ and $[-f/2, f/2]$, respectively, with a step size of 0.05. As a result, 250047 (x_k, y_k) pairs were obtained. Then a coordinate dictionary that keeps the (x_k, y_k) pairs as keys and the angles as corresponding values was created.

When a coordinate pair is searched in the dictionary, the pair that has the lowest Euclidian distance to the searched pair is considered as the best match and corresponding angles are used to construct the joint angle.

The algorithm explained above was used to determine only the angles on the x-y plane. The last angle was calculated on the x-z plane (Fig. 7) using (4).

Fig. 7. The last needed angle (θ) on the x-z plane.

$$\theta = \tan^{-1}(z_h / x_h). \quad (4)$$

Determined four angles make the robot arm able to reach certain targets in 3-dimensional space.

III. EXPERIMENTAL SETUP AND RESULTS

Generated system is shown on Fig. 8 and Fig. 9. Object recognition is tested on both training and test datasets using MLP. 10-fold cross validation schema is used for performance evaluation. In 10-fold cross validation, the dataset is randomly divided in 10 disjoint sets and nine sets are used for training purposes and the remaining is used for testing. This procedure is repeated until each set is used for testing. Performances of classification tasks are given in terms of recall (5), precision (6) and specificity (7) of each object and also average accuracy (8). These terms are calculated according to confusion matrix and are formulated as:

$$\text{recall} = \frac{TP}{TP + FN} \times 100\%, \quad (5)$$

$$\text{precision} = \frac{TP}{TP + FP} \times 100\%, \quad (6)$$

$$\text{specificity} = \frac{TN}{TN + FP} \times 100\%, \quad (7)$$

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%, \quad (8)$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, FN is the number of false negatives. The terms positive and negative refer to the classifier's prediction, and the terms true and false refer to whether that prediction corresponds to the real label of samples.

The classification results are given in Table III and Table IV. Average accuracies of 98.62 % and 93.83 % were obtained for training and test datasets respectively. As can be seen from Table III, all objects were identified with recall values of higher than 97 % with specificity values higher than 99 % for the training dataset as expected.

Table IV shows the systems performance for the test dataset. Note that the system is trained with single objects and the test dataset includes the combination of these objects in various position and locations. The results are still higher than 96 % except for the knife. The recall for the knife is 62.38 % because the object recognition system confuses the knife with the fruit knife.

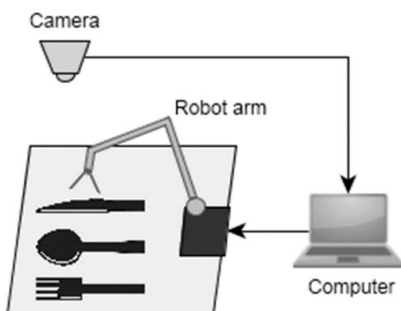


Fig. 8. Illustration of the system.

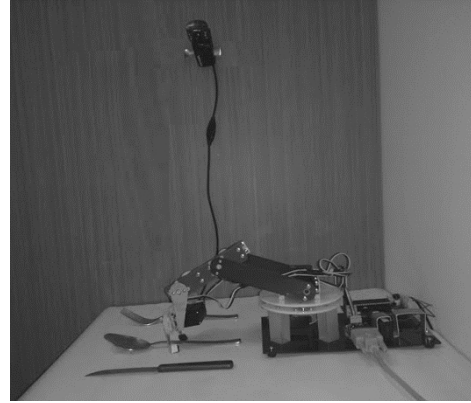


Fig. 9. Application of the system.

TABLE III. PERFORMANCE EVALUATION OF TRAINING DATA.

Item	Recall	Specificity	Precision
Knife	97.37 %	99.30 %	94.87 %
Fork	98.46 %	99.62 %	98.46 %
Spoon	98.55 %	99.61 %	98.55 %
Fruit Knife	97.33 %	100 %	100 %
Oval Plate	100 %	99.64 %	97.96 %
Average	98.62 %	99.70 %	98.31 %

TABLE IV. PERFORMANCE EVALUATION OF TEST DATA.

Item	Recall	Specificity	Precision
Knife	62.38 %	99.24 %	92.65 %
Fork	99.04 %	99.64 %	99.04 %
Spoon	99.50 %	100 %	100 %
Fruit Knife	96.27 %	93.68 %	80.31 %
Oval Plate	100 %	99.70 %	97.89 %
Average	93.83 %	98.43 %	94.35 %

After object recognition, centroids of detected spoons, forks, and knives and edge coordinates of plates are determined.

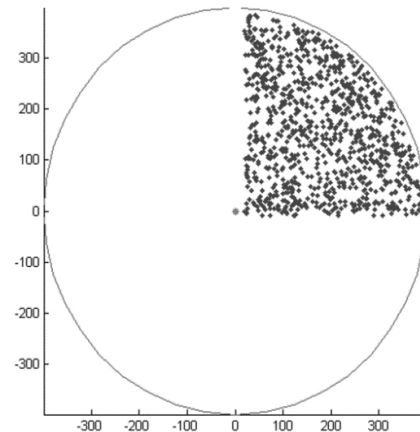


Fig. 10. Randomly generated 1000 points for comparison of the methods.

In the previous study [16], gradient descent algorithm was used to calculate angles required on the x-y plane [16]. Gradient descent algorithm converges to the minimum of a function, step by step. It was used to minimize the error function that represents the difference between the target and the current position on the x-y plane. It is converged to the minimum point of the error function by following the opposite direction of the gradient. The converging iterations are ended when the absolute difference between the last

value and the previous one is reached a predefined sensitivity value.

In this study, robot arm's joint angles were determined using the coordinate dictionary method. Performances of the gradient descent and the coordinate dictionary algorithms are compared in Table V in terms of Euclidian distance error and time consumed while finding the best solution for the objects. Comparison was performed using 1000 points that were generated randomly (Fig. 10) in a region bounded by lines: $x = 20$, $y = -10$ and the circle $x^2 + y^2 = 400^2$. Values are given in millimetres.

TABLE V. TEST RESULTS OF THE GRADIENT DESCENT ALGORITHM (GDA) AND THE COORDINATE DICTIONARY METHOD (CDM) FOR RANDOMLY GENERATED 1000 POINTS.

	Time (seconds)		Euclidian distance error (millimeters)	
	GDA	CDM	GDA	CDM
Min	0	0.005411	0.001113	0.009403
Max	1.45	0.006539	15.199337	1.407663
Mean	0.030202	0.005579	0.038011	0.523210
Std. dev.	0.054243	0.000163	0.510101	0.258507

The results for gradient descent algorithm based [16] and coordinate dictionary based joint angle calculations are given in Table V. Results show that the joint angles are calculated in 5.579 milliseconds with 0.523 millimeters Euclidian distance error which is an ignorable error for movement of the robot arm. Results also show that the coordinate dictionary method is much faster than the gradient descent method in which the joint angles are calculated in 30.202 milliseconds. The standard deviation of the distance error is 0.259 millimeters, which is almost half of standard deviation value when the gradient descent algorithm is used, which means that it produces more stable results than the method used in [16].

IV. CONCLUSIONS

In this study, a smart robot arm system is designed. The system can detect and identify cutlery and plates and lift them from a table. Average recall values of 98.62 % and 93.83 % are obtained for training and test sets in the classification of the objects. In the previous study [16], the smart robot arm system was performed average accuracy of 90 % using kNN classifier with the same features. Performance of the system is increased by the use of MLP for classification. This results shows MLP is better model to classify the objects with extracted features.

The robot arm joints' angles were calculated with an average Euclidian distance error of 0.523 millimeters in an average time of 5.579 milliseconds. This is a very fast response with an acceptably small distance error for the robot arm.

Methods for better object recognition and classification and better coordinate value estimation in a less response time might be searched for future work. Besides, this study can be re-performed using a robot arm that has more fingers (three or five fingers). Additionally, instead of detecting all the objects in the image automatically and lifting all of them, the algorithm might be changed such that only a predefined desired object is searched for and lifted for a more effective usage of the robot arm.

REFERENCES

- [1] A. D. Kulkarni, *Computer vision and fuzzy-neural systems*. Prentice Hall PTR, 2001, ch. 2 and ch. 6.
- [2] R. Jain, R. Kasturi, B. G. Schunck, *Machine Vision*, McGraw-Hill 1995, ch. 14.
- [3] D. A. Forsyth, J. Ponce, *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference, 2002, ch. 15.
- [4] G. Bradski, A. Kaehler, V. Pisarevsky, "Learning-based computer vision with Intel's open source computer vision library", *Intel Technology Journal*, vol. 9, 2005.
- [5] C. H. Lampert, H. Nickisch, S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer", *IEEE Computer Vision and Pattern Recognition*, pp. 951–958, 2009. [Online]. Available: <http://dx.doi.org/10.1109/cvpr.2009.5206594>
- [6] A. Saxena, J. Driemeyer, A. Y. Ng, "Robotic grasping of novel objects using vision", *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008. [Online]. Available: <http://dx.doi.org/10.1177/0278364907087172>
- [7] R. Szabo, A. Gontean, "Full 3D robotic arm control with stereo cameras made in LabVIEW", *Federated Conf. Computer Science and Information Systems (FedCSIS)*, 2013, pp. 37–42.
- [8] Y. Hasuda, S. Tshibashi, H. Kozuka, H. Okano, J. Ishikawa, "A robot designed to play the game Rock, Paper, Scissors", *IEEE Industrial Electronics*, pp. 2065–2070, 2007. [Online]. Available: <http://dx.doi.org/10.1109/isie.2007.4374926>
- [9] Ishikawa Watanabe Lab., University of Tokyo. [Online] Available: www.k2.t.u-tokyo.ac.jp/fusion/Janken/index-e.html
- [10] A. Shaikh, G. Khaladkar, R. Jage, T. Pathak, J. Taili, "Robotic arm movements wirelessly synchronized with human arm movements using real time image processing", *IEEE India Educators' Conf. (TIIEC), Texas Instruments*, 2013, pp. 277–284.
- [11] S. Manzoor, R. U. Islam, A. Khalid, A. Samad, J. Iqbal, "An open-source multi-DOF articulated robotic educational platform for autonomous object manipulation", *Robotics and Computer-Integrated Manufacturing*, vol. 30, no. 3, pp. 351–362, 2014. [Online]. Available: <http://dx.doi.org/10.1016/j.rcim.2013.11.003>
- [12] N. Rai, B. Rai, 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 Int. Conf.*, pp. 1–9, 2014. [Online]. Available: <http://dx.doi.org/10.1109/et2ecn.2014.7044931>
- [13] T. P. Cabre, M. T. Cairol, D. F. Calafell, M. T. Ribes, J. P. Roca, "Project-based learning example: controlling an educational robotic arm with computer vision", *Tecnologias del Aprendizaje, IEEE Revista Iberoamericana de*, vol. 8, no. 3, pp. 135–142, 2013. [Online]. Available: <http://dx.doi.org/10.1109/RITA.2013.2273114>
- [14] Y. Kutlu, M. Kuntalp, D. Kuntalp, "Optimizing the performance of an MLP classifier for the automatic detection of epileptic spikes", *Expert Systems with Applications*, vol. 36, no. 4, 2009. [Online]. Available: <http://dx.doi.org/10.1016/j.eswa.2008.09.052>
- [15] C. M. Bishop, *Neural networks for pattern recognition*. Clarendon Press, 1995, ch. 4.
- [16] B. Iscimen, H. Atasoy, Y. Kutlu, S. Yildirim, E. Yildirim, "Bilgisayar gormesi ve gradyan inis algoritmasi kullanilarak robot kol uygulamasini", *Akilli Sistemlerde Yenilikler ve Uygulamalari (ASYU) Sempozyumu*, 2014, pp. 136–140. (in Turkish).