Journal of Innovative Image Processing (JIIP) (2020)
Vol.02/ No. 03
Pages: 141-146
https://www.irojournals.com/iroiip/
DOI: https://doi.org/10.36548/jiip.2020.3.004

An Economical Robotic Arm for Playing Chess Using Visual Servoing

Dr. G. Ranganathan,
Head of the department,
Department of Electronics & Communication Engineering,
Gnanamani College of Technology,
Namakkal, India – 6370 18.
Email: profranganthang@gmail.com

Abstract: The proposed paper outlines the design of an economical robotic arm which is used to visualize the chess board and play with the opponent using visual servoing system. We have used the FaBLab RUC's mechanical design prototype proposed and have further used Solidworks software to design the 4 jointed gripper. The proposed methodology involves detecting the squares on the corners of the chessboard and further segmenting the images. This is followed by using convolutional neural networks to train and recognize the image in order to determine the movement of the chess pieces. To trace the manipulator, Kanade-Lucas-Tomasi method is used in the visual servoing system. An Arduino uses Gcode commands to interact with the robotic arm. Game Decisions are taken with the help of chess game engine the pieces on the board are moved accordingly. Thus a didactic robotic arm is developed for decision making and data processing, serving to be a good opponent in playing chess.

Keywords: Chess, Robotic Arm, Arduino, Visual Servoing, Image detection and segmentation

1.Introduction

It was in 1769 that their chess playing machine called "The Turk" was developed by Wolfgang von Kempelen. This machine was designed such that it was operated by a chess master to play against a human opponent. Over the past decades, researchers have been working on Artificial Intelligence to develop an algorithm that could autonomously play chess. David Champernowne and Alan Turing initiated this venture by developing “Turning's paper machine", which is a program that possessed the ability to play a complete game of chess. With the simple aid of paper and pencil, instructions were executed by Turing to process the task. It was only in 1996 did IBM's Deep Blue [3] come into existence a complete computer program which also defeated Gary Kasparov, the World Chess Champion. A number of chess engines have been introduced since then and it has also led to simplification of robot building. This mechanism of autonomous robotics systems for chess game proves to be a good application that serves as a test for implementation of other autonomous robotics systems which need solution for manipulation, perception and interactive between the robot and human [4]. Based on the various perception of the game, many alternatives have been suggested over the years with multiple. Focus on the different chess pieces was made in [1] but is was found that this system could not give a precise positioning of the chess pieces. Similarly, researchers have proposed ways to detect the chess pieces, their movement and their strategy based on the game’s output. In [2] depth and 2D cameras were used to determine the program of the game and further analysis showed that it was possible to compare the previous and current state using the cameras. The proposed work makes use of a fisheye camera is used which is attached to the gripper and serves as the visual monitor for the arm [5]. Over the years a number of detection techniques have been used in order to detect color variation, occupancy of squares and border detectors. Based on the literature review, it has been observed that edge-based methods have gone a considerable way in determining the position of the chess pieces and also work remarkably well in low contrast. The proposed work mainly entitles the use of hybrid Siamese Network [6] which uses comparison layer to determine the changing position of the chess pieces. This was also found to be efficient in quickening the process of database creation by data tagging. Even among very low color contrast, this Siamese network approach resulted in provided excellent results and their robustness against changing lighting condition is very good. Kanade-Lucas-Tomasi [7] feature extraction is used to implement the visual servoing algorithm, based on the observed image [8-9].

ISSN: 2582-4252 (online)
Submitted: 10.05.2020
Accepted: 24.06.2020
Published: 29.06.2020

2. Architecture and Components

The proposed system is made up of an Arduino board which is connected to the computer an can be used to control the two stepper motors with which the robotic arm operates. This system will require a fisheye camera with 170 degree rotation capability [10] that is placed on the arm grip. The robotic arm is powered by the computer by means of the USB.

2.1. Robotic Arm Design

The proposed methodology has implemented modifications in the rapid prototype of the robot structure devised by FabLab RUC in Denmark. This design is an open source which makes use of CNC diagram for certain software elements and laser cutting which are not included in this research. The prescribed design uses Nema 42 stepper motor and is cut in 12mm of plywood. It was found that the project budget was too pricey and hence the proposed methodology uses a Nema 23 motor which is smaller and more economical to use. Similarly all other components have also been minimized by two-thirds of a factor from the original design. The plywood thickness has also been reduced to 9mm from 12m [11-12]. Every stepper motor is given an end-stop switch at the joint which serves as the reference position.

2.2 Gripper Design

An Ultimaker 3D printer is used to print the four-finger gripper that is built with the help of solidworks software.

2.3 Arduino

To communicate between the stepper driver and the computer, an Arduino mega 2560 is being used. This Arduino is used to control the drivers by means of digital pins and is triggered by starting the motor.

2.4 Stepper Drivers

The Nema 23 motor is powered by means of the TB6560 driver that rotates on 3 fixed axes. The parallel port enables interfacing the driver. Step pins, direction and activation is used for each individual motor. Using the end-stop switches of the Arduino, task is performed thereby proving the limit pins of the 3 axis to become void.

2.5 G-Code Commands

G-code commands are used to take control over the robotic arm by means of communication through the Arduino. The G-commands used are as follows:

- M114: Used to transfer the current position of the motor.
- M17: The stepper motors are enabled, although the position of the motors are not known
- M18: The stepper motors are disabled and the robot can be manually moved.
- G1: This command is used to start a pathway between the required coordinate and the present location of the motor. However, if any change is observed in the command, then the direction of the movement will also be changed accordingly. If the position of the arm is unknown, then movement is not possible, foreseeing the possibility of collision of the joints. Similarly the current position of the motor is also sent by the Arduino.
- G28: With positions fixated on point zero, all motors are said to move towards the end-stop switches. When this is triggered, signals are sent by the Arduino via the serial port.

ISSN: 2582-4252 (online)
Submitted: 10.05.2020
Accepted: 24.06.2020
Published: 29.06.2020

3. Proposed Work

3.1 Change Detection

The chess board corners are detected by means of edge detection and the obtained image is further segmented such that it is possible to detect the pieces despite low contrast of the background. Every move the game is tracked using the change detection. At the outset only color classification was tested for different models and based on the output, it was found that the convolutional neural network gave the best results. Accordingly, the pieces and their position can be classified as follows:

- Black pieces on green background
- Black pieces on white background
- White pieces on green background
- White pieces on white background

where the color of backgrounds used were white and green.

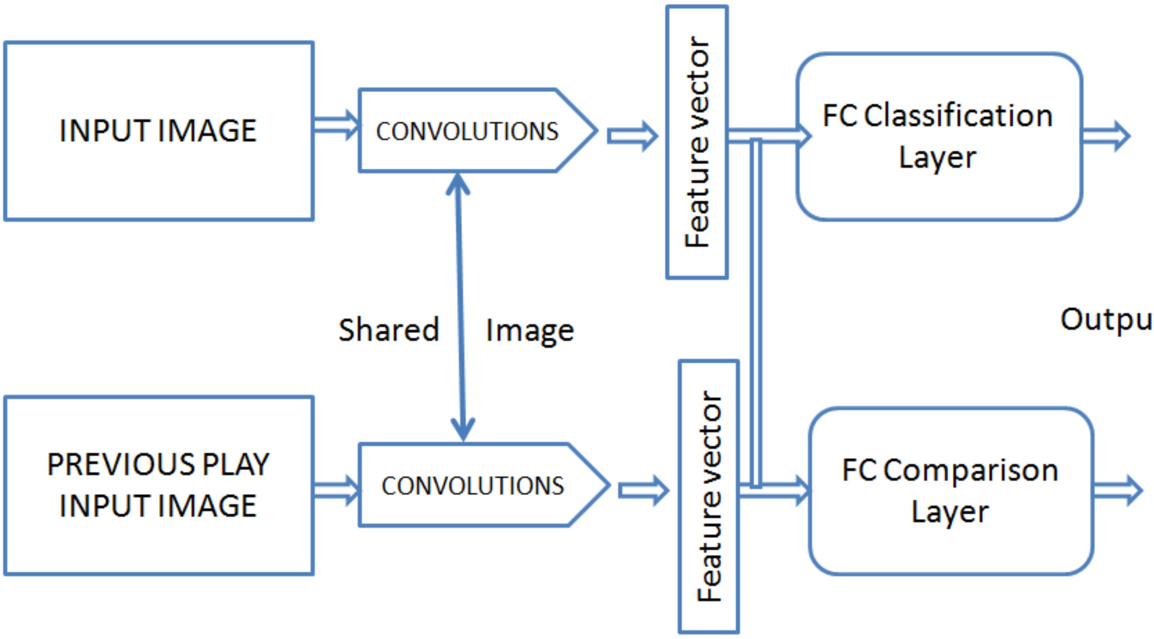

Fig.1. Siamese Network Structure

Further, to find the movement made, the image of the previous move is used to compare with that of the current one in the Siamese network. For the purpose of color recognition, feature vector is used and further by adding network layers detection of movement is also made. Since the color contrast of white board with white pieces is difficult to identify, the Siamese network is used in the proposed model. It is also to be noted that the use of it is

ISSN: 2582-4252 (online)
Submitted: 10.05.2020
Accepted: 24.06.2020
Published: 29.06.2020

not enough to simply compare the output of classification using class classification. Keras [13] designed the model wherein 1 output is used for detecting change and 6 outputs are used for class classification as shown in Fig.1.

3.2 Database Collection and Training

The software developed has a number of features such as predicting the next move with the help of the current model, exporting the file using npz extension, labeling the data panel, color labels and board corners. Thus a complete database of images is created. To have a more realistic data, the pieces of the board are misplaced, the lighting conditions are changed and the robot camera position is also moved by some inches for each image. In order the train the system, the corners, images and labels of the npz files are imported. With the reference of the corners, the images are trimmed such that they have a padding which is 0.25 times that of the square size. In this manner, the surrounding squares of the image will enable recognition of background color as shown in Fig.2. The images are built such that they use 3 color channels to the size of 25x25 pixels. A ratio of 9:2.5 is used for separating the training and test data. In order to use the data better, the images are flipped and rotated. A total of 8 images are generated based on the original database for the individual images.

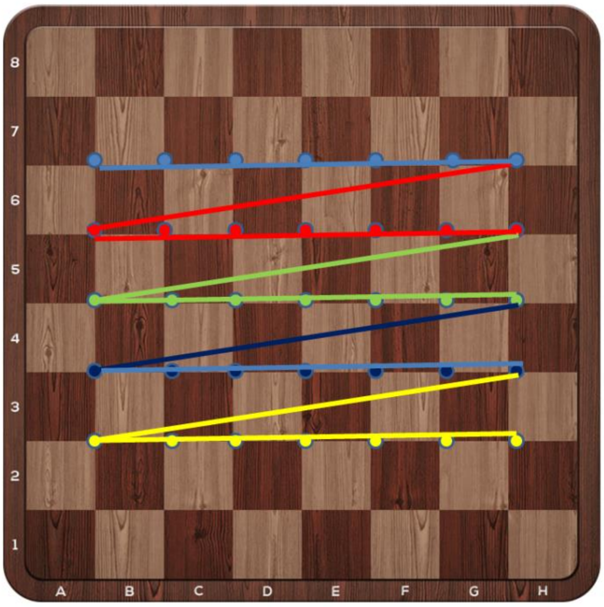

Fig.2 Edge detection on a chessboard

3.3 Validation

Movement validation and game decisions are made with the help of Stockfish chess engine. The initial board position is detected using color prediction and based on the changes made using tracking. In order to detect the change in movement, a comparison is made between 1 and 0 which are the output values.

ISSN: 2582-4252 (online)
Submitted: 10.05.2020
Accepted: 24.06.2020
Published: 29.06.2020

4. Results and Discussion

It is observed that between the epochs of 35 and 150, there is a fall in the test data from the actual performance of the data and finally overlap with the actual curve. Based on the observation, it also shows that the success rate at which the positions of the chess pieces are detected is set at about 506/516. Fig.3 and Fig.4 represents the graph the difference in the test data from the actual data.

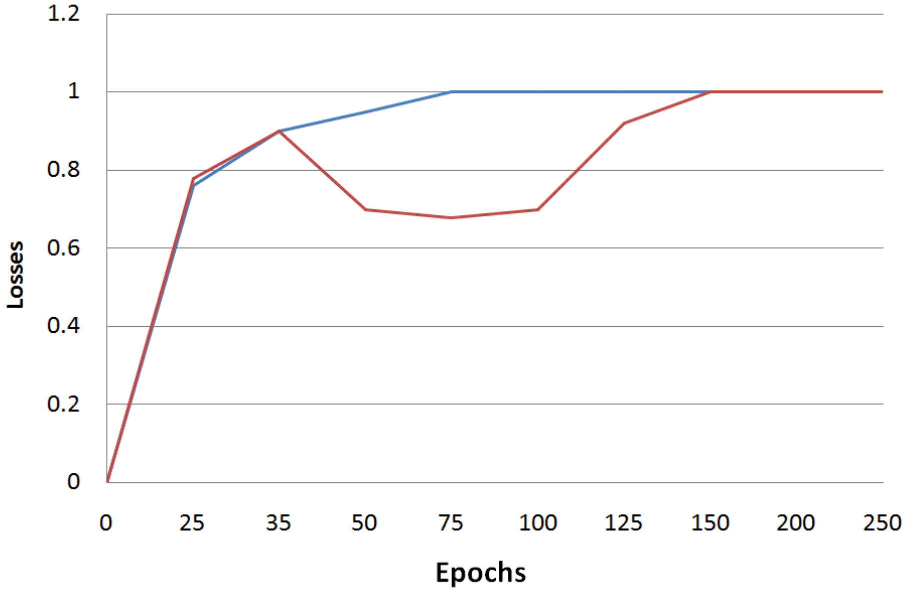

Fig. 3. Losses Vs Epochs

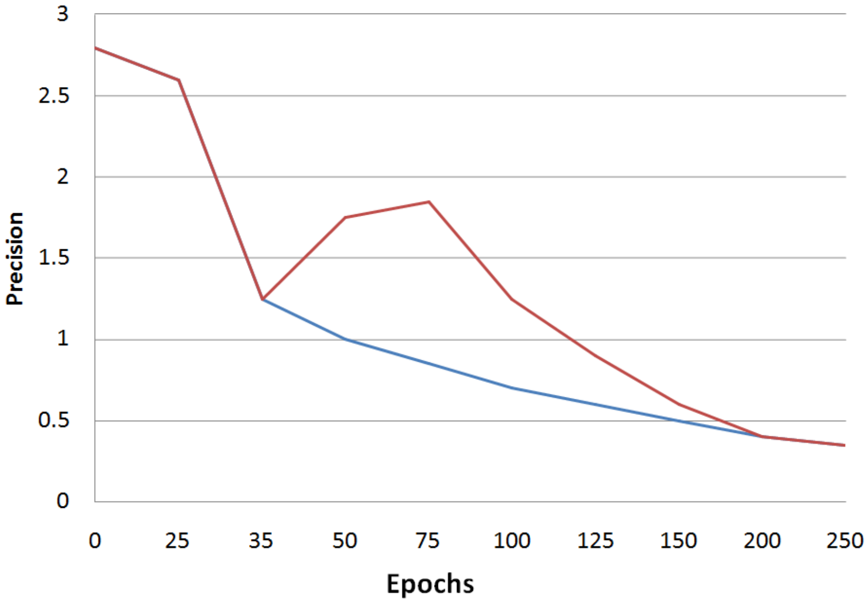

Fig. 4. Precision Vs Epochs

ISSN: 2582-4252 (online)
Submitted: 10.05.2020
Accepted: 24.06.2020
Published: 29.06.2020

5. Conclusion

The movement detection and color classification of the chess pieces is performed using a machine learning model implement using a chess-playing robot. When there is low color constrast, a CNN is called for to recognise the chess piece in action. This output is used for tracking the change and classifying the class. The robot is also equipped with a fisheye camera that serves as a visual input in order to grip the piece and move it based on Kanade-Lucas_tomasi object manipulation algorithm. The drawback is that when repeatability problem occurs, the robot might retrace its previous positions instead of looking for an alternative move.

References

[1] Bechlioulis, C. P., Heshmati-Alamdari, S., Karras, G. C., & Kyriakopoulos, K. J. (2019). Robust image-based visual servoing with prescribed performance under field of view constraints. IEEE Transactions on Robotics, 35(4), 1063-1070.

[2] Toshev, A., Sadeghi, F., & Levine, S. (2020). U.S. Patent Application No. 16/622,181.

[3] Zake, Z., Caro, S., Roos, A. S., Chaumette, F., & Pedemonte, N. (2019, June). Stability analysis of pose-based visual servoing control of cable-driven parallel robots. In International Conference on Cable-Driven Parallel Robots (pp. 73-84). Springer, Cham.

[4] Ourak, M., Tamadazte, B., Lehmann, O., & Andreff, N. (2019). Direct visual servoing using wavelet coefficients. IEEE/ASME Transactions on Mechatronics, 24(3), 1129-1140.

[5] Xu, T., Guan, Y., Liu, J., & Wu, X. (2019). Image-based visual servoing of helical microswimmers for planar path following. IEEE Transactions on Automation Science and Engineering, 17(1), 325-333.

[6] Tian, N., Tanwani, A. K., Chen, J., Ma, M., Zhang, R., Huang, B., ... & Sojoudi, S. (2019, May). A fog robotic system for dynamic visual servoing. In 2019 International Conference on Robotics and Automation (ICRA) (pp. 1982-1988). IEEE.

[7] Ruth Anita Shirley D, Ranjani K, Gokulalakshmi Arunachalam, Janeera D.A., "Distributed Gardening System Using Object Recognition and Visual Servoing" In International Conference on Inventive Communication and Computational Technologies [ICICCT 2020], Springer, India, 2020.

[8] Suma, V. (2019). Computer Vision for Human-Machine Interaction-Review. Journal of trends in Computer Science and Smart technology (TCSST), 1(02), 131-139.

[9] Bindhu, V. (2019). Biomedical Image Analysis using Semantic Segmentation. Journal of Innovative Image Processing (JIIP), 1(02), 91-101.

[10] Chen, J. I. Z., & Chang, J. T. (2020). Applying a 6-axis Mechanical Arm Combine with Computer Vision to the Research of Object Recognition in Plane Inspection. Journal of Artificial Intelligence, 2(02), 77-99.

[11] D. A. Janeera and Sasipriya.S. "A Brain Computer Interface Based Patient Observation and Indoor Locating System with Capsule Network Algorithm" In International Conference on Image Processing and Capsule Networks (ICIPCN 2020), Springer, Thailand, 2020.

[12] Zhang, X., Fang, Y., Zhang, X., Jiang, J., & Chen, X. (2019). A novel geometric hierarchical approach for dynamic visual servoing of quadrotors. IEEE Transactions on Industrial Electronics, 67(5), 3840-3849.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 580–587.

ISSN: 2582-4252 (online)
Submitted: 10.05.2020
Accepted: 24.06.2020
Published: 29.06.2020