Chess Recognition Using Computer Vision

May 30, 2017

Ramani Varun (U6004067, contribution 50%)
Sukrit Gupta (U5900600, contribution 50%)

College of Engineering & Computer Science
The Australian National University
Canberra, ACT 0200, Australia

***Abstract*— This report highlights approaches taken to process an image of a chessboard and identify the configuration of the board using computer vision techniques. Although, the use of a chessboard detection for camera calibration is a classic vision problem, existing techniques on piece recognition work under a controlled environment. The procedures are customized for a chosen colored chessboard and a particular set of pieces. The method highlighted in this report supplements existing research by using clustering to segment the chessboard and pieces irrespective of color schemes. For piece recognition, the method introduces a novel approach of using a R-CNN to train a robust classifier to work on different kinds of chessboard pieces. The method performs better on different kinds of pieces as compared to a SIFT based classifier. If extended, this work could be useful in recording moves and training chess AI for predicting the best possible move for a particular chessboard configuration.**

I. INTRODUCTION

The detection and tracking of board games using vision techniques is a familiar problem. Researchers and hobbyists have undertaken this task multiple times over the years on different board games. However, the setup is expensive and work on specialized boards and pieces under a controlled environment. The various assumptions make the gap between chess playing AI and real world relevant because it still depends on human intervention in order to perform. The goal of our project is to bridge this gap and design a system which would work irrespective of the type of chess board, pieces or the angle from which the image is taken. The motivation was derived from letting any user know the best possible move for the current chessboard configuration by taking an image of the board using a smartphone application. The availability of insights like these help in developing fundamental skills and tactics for a chess player. We implement our approach on the assumption that the user is only able to provide one image, from anywhere between a side angle to a stable overhead angle view of the board, to minimize usage restriction and hindrances for the users. We apply various processing methods on a chessboard image taken during the game by a player to recognize the board configuration. The first step is to detect the board irrespective of the angle from which the image was taken. We then identify the board area and then apply appropriate clustering techniques to segment out the board and the pieces separately. This method is effective for boards with different color schemes. Once the pieces are detected, we run them through a neural network based classifier to identify them. The use of R-CNN based approach is found particularly useful for detecting chess pieces from various sets thus making the classifier robust. The remaining parts of the report will highlight previous approaches taken for chess recognition and build on it using appropriate techniques and methodologies that show improvement and provide robustness to the algorithm; with supporting technical explanation of our approaches and discussing experimentation results and outcomes.

II. RELATED WORK

The concept of chess recognition mainly consists of two broad areas; The recognition of the chess board from the image taking in consideration the size, orientation and the varying colours of the tiles on the board, and the second is recognition of the chess pieces which is a combination detection, localisation and identification.

A. Board Recognition

Previous work done on chess recognition in the field of computer vision focuses on the area of board recognition and applies various techniques to find the solutions. One of the approaches used in [1] was to eliminate the errors caused by noise in the image when performing board detection from an image taken at an angle. This was achieved by making an interactive implementation where the user has to select the four corner points of the board manually and then using these points a projective transformation is performed. This approach was justified in [1] by stating that the primary focus was on piece recognition and that this method reduced the errors caused by imperfect board detection. This approach works well if the aim is to just recognize chess pieces but fails short on the premise of making the user choose corner points of the board manually every time the recognition program needs to run.

The de la Escalera paper on camera calibration [6] employs the use of corner detection and line detection in conjunction to detect the chessboard correctly. This approach of first performing Harris corner detection and then making use of the detected points to apply Hough transform has

proved successful in pattern recognition. [5] employs the same algorithmic approach but removes corner detection, citing that tests showed a high proportion of corners detected that were not relevant to the board. We employ a similar approach in our project to perform chess board recognition.

B. Piece Recognition

Commonly cited chess playing robot projects [7] and [8] expect human intervention of manually entering the initial positions of chess pieces. Piece movements are then tracked during the game. This is done because identifying chess pieces is a challenging task as they dont have texture. Matching pieces through SURF descriptors dont furnish acceptable results. Techniques that dont assume starting positions use color segmentation to detect pieces and use shape descriptors for identification [1, 6, 9]. However, this approach relies on the difference between the color of the chessboard and the pieces. Approach in [5] suggests the use of pattern or shape based recognition as it depends on contours. This method does polygonal approximation of shapes and ignores the finer details of the object of interest. Fourier descriptors are used to build a shape representation and the classification is done using k-means. However, it is biased towards a particular chess piece set. The approach in [1] demonstrates the use of scale-invariant feature transform (SIFT) and histogram of oriented gradients (HOG) for feature extraction. A SVM is then used to train a binary one-vs-rest classifier. The piece recognition has near perfect accuracy but like the above mentioned approach they are biased towards a particular set of pieces and dont perform well on other boards. A better approach could be to use a R-CNN to train a network for different kinds of chess pieces. The network would learn a representation for each of the chess pieces and perform consistently across different sets. This method however, relies on huge amounts of training data.

III. Approaches

The objective of the approach is to detect and recognize chess pieces for images taken from a side angle. Fig. 1. shows the block diagram for the chess recognition system. The structure has been divided into two components - board recognition and piece recognition, throughout the report for better analysis.

A. Board Recognition and Segmentation

After capturing an image of a set chessboard from a side angle, the first step is to pre-process the image by applying image filtering and resizing operations. A precursor to line detection using **Hough Transform** is to perform edge detection using the **Canny** method. The hough peaks are used to determine the lines which in our case go through the corners of the board squares. The intersection of these lines are used to determine the corner points for each of the square. Once we find the corner points, a minimal bounding box for all these points are found. The corners of this bounding box are considered as the endpoints of the chessboard.

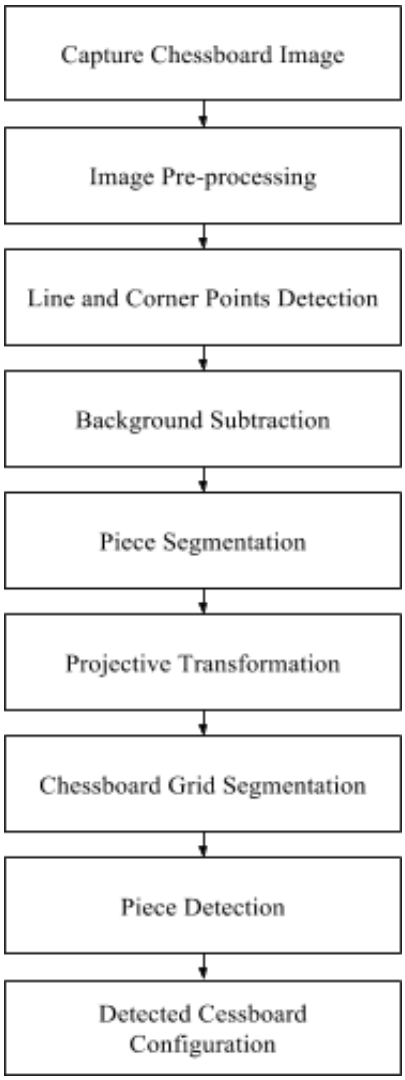

Fig. 1. The figure shows the approach sequence taken in project

In order to perform projective transform on the image, we require two set of coordinates - real world and image coordinates. The correspondence between the two points is used to determine the projective transformation matrix. The world coordinates are determined from a 640x640 pixel chessboard image whereas the image coordinates are the corners obtained from the previous step.

The projective transformation matrix is calculated as follows. Let P_i and $(u, v, 1)_1^T$ be the corresponding corner points in homogeneous coordinates on input image and rectified image where $i \in 1, 2, 3, 4$. Then projective transformation matrix can be found up to a scale factor by solving the system of equations $Ph = 0$, where

$$P := \begin{bmatrix} P_1^T & 0 & -u_1 P_1^T \\ 0 & P_1^T & -v_1 P_1^T \\ . & . & . \\ . & . & . \\ P_4^T & 0 & -u_4 P_4^T \\ 0 & P_4^T & -v_4 P_4^T \end{bmatrix}$$

and h is H reshaped into a column vector. By taking the first column vector of the third returned matrix after applying SVD to P we can find the unknown h.

The obtained transformation matrix is used to project the image and obtain the top view of the board. This step elaborated in the next section, is useful to determine configuration of the board by finding the relative position of the detected pieces.

Following the projective transformation we take a step back with the aim to segment out the board and the pieces. [5] takes an approach of segmenting the pieces out by subtracting the original image with the red and green channels. This step is effective due to their assumption of using a chessboard with red and green square tiles [5]. Because we aim to make our method more generic to work across chessboards, we employ **k-means clustering** to segment out the board and the pieces. For k-means to work properly we use our knowledge of the board boundary to subtract the background out. We then feed the pixels of interest into the algorithm by setting number of clusters as 2. Since the board occupies the majority of the image, the histogram information can be used to map the identified cluster center to the object of interest.

The cluster containing the chess piece pixels is used for further processing. A simple thresholding operation is performed to binarize the image and obtain the blobs. Morphological operators are used to close the holes and separate blobs too close to each other because they might be adjacent pieces. Once the blobs are identified, a minimal bounding box is drawn around each of the them. These bounding boxes are used to crop out pieces of interest required for the next step.

B. Piece Recognition

Existing method as per [1] evaluates piece recognition performance by using two separate descriptors - scale-invariant feature transform (SIFT) [2] and histogram of oriented gradients (HOG) [3]. SVMs are then used to train a one-vs-rest classifier for each class [1]. But use of such descriptors keep the detection limited to a particular chess pieces. But we aim for the system to work on different kinds of chess pieces. For this purpose, we use a Regions with Convolutional Neural Network (R-CNN) [4] to train an object detector.

This method unlike the previous methods does not use a sliding window technique and rather work on regions where the object are likely to be present. Rather than training a network from scratch, we use transfer learning by starting with a pre-trained network. This approach saves a lot of time and effort and works on the principle of tuning the network using newly fed images. We used the pretrained network, alexnet for our system. The layers in the Convolutional Neural Network (CNN) comprise of : an input layer, 2D convolutional layer, rectified linear unit (ReLU), max pooling layer, softmax layer and finally a classification layer. The middle three layers are repeated units which determine filter weights. The training is done using Stochastic Gradient Descent with Momentum (SGDM) methods with the initial learning rate set to 0.001. After the network is trained a linear SVM is used to detect the score of each of the classes. The class with the highest score is the label assigned to the detected chess piece. Table 1. shows size of the training set by type of chess piece.

TABLE I
CHESS PIECE TRAINING SET

Type of Chess Piece (class)	Size of training set(white&black each)
King	100
Queen	100
Empty Chess Squares	100
Pawn	100
Bishop	100
Knight	100
Rook	100

IV. IMPLEMENTATION AND RESULTS

In our approach we assumed that the camera angle for the input image is flexible ranging from a side profile to an overhead angle. In an optimal case, the overhead angle is preferred as there is no need for perspective transformation and the edges of the board are not hindered by the chess pieces. To support images not taken from an overhead angle we implemented various methods mentioned in the section above to rectify the image by performing projective transformation using homography matrix obtained from the detected corners. The user input image is first preprocessed (rescaled and filtered) for the purpose of faster computation and reduction of noise. We run the pre-processed image

Fig. 2. Input image of the chessboard

through a canny edge detector which extracts useful structural information from our image and reduces the amount of data needed to be processed for the line detecting Hough transform method.

Next we perform our Hough line transform using the above canny edge detected output image and extract our required lines from a two-dimensional array called the Hough accumulator. We overlay these lines on top of our pre-processed image and extract the intersection points and their

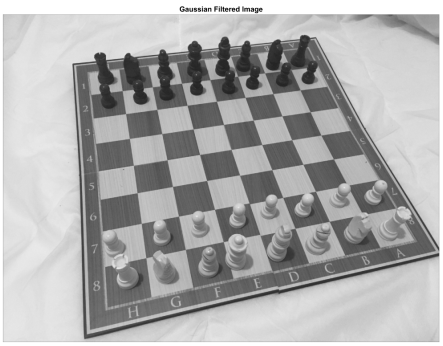

Fig. 3. Image after applying Gaussian filter

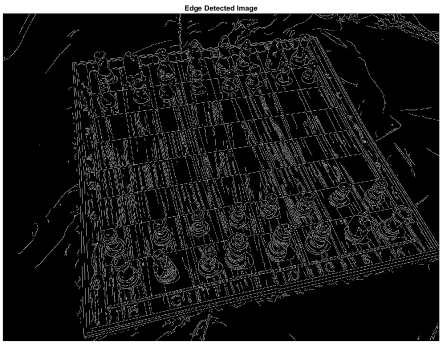

Fig. 4. Image after applying canny edge detector

pixel coordinates.

These intersection points as shown in Fig. 5. represent the corner points of our chessboard tiles and the minimum bounding box of these corners represent the boundary of the chess board as illustrated in Fig. 6.

To obtain the projective transformed output image we make use of our real world and image coordinates (homography matrix) and perform the transformation to obtain a top view image of the chess board.

After successfully transforming the image we move onto recognising the board and segmenting it by applying color based clustering on our image. This segmentation is important to differentiate the board from the chess pieces. We apply k-means clustering for this purpose. Using the boundary information extracted from the previous task we subtract the background and only consider the pixel values related to the chess board.

Fig. 5. Image showing intersecting points for Hough lines

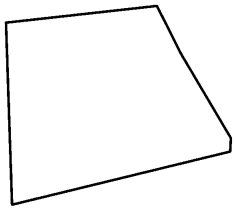

Fig. 6. Detected boundary of the chessboard

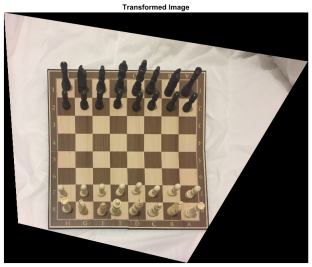

Fig. 7. Projective transformed image of the chessboard

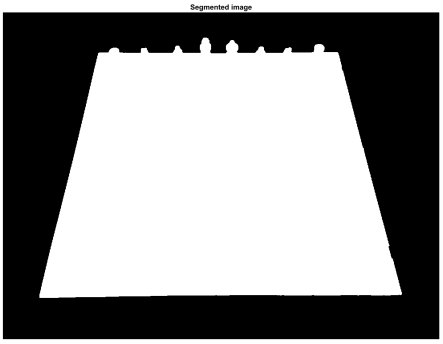

Fig. 8. Segmentation between the board and background

We feed only the required pixel values into our k-means clustering algorithm and set the cluster size to two as we ideally require one pixel cluster representing the chess board and the other cluster representing the pieces. Our results for these clusters are shown in Fig. 9 and Fig. 10 respectively.

We further process cluster two containing the pixels for chess pieces by applying threshold operation to binarize the image. The blobs obtained as seen in Fig. 11 are further optimized by performing morphological operations to separate them from other relatively nearby blobs. The morphological operation results as seen in Fig. 12 help in constructing minimal bounding box around these blobs, representing the detection

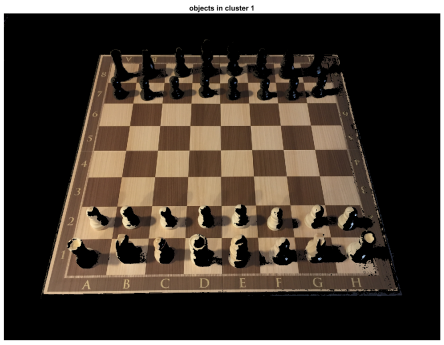

Fig. 9. Clustered pixels belonging to the chess board

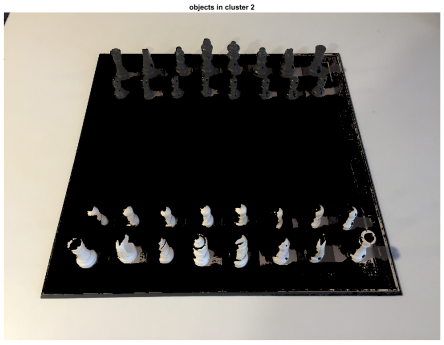

Fig. 10. Cluster pixels belonging to the chess pieces

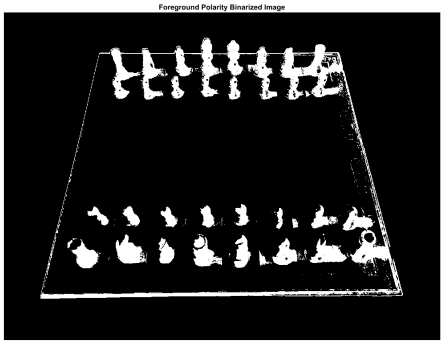

Fig. 11. Image obtained from cluster 2 after being binarized

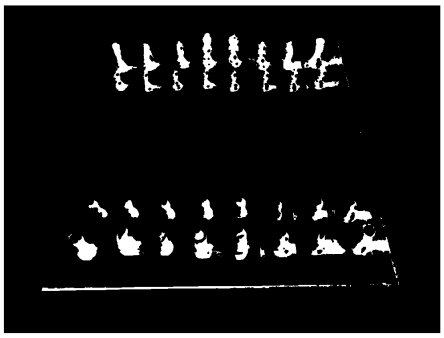

Fig. 12. Binarized image after applying morphological operations

of chess pieces. As the result illustrated in Fig. 13 shows

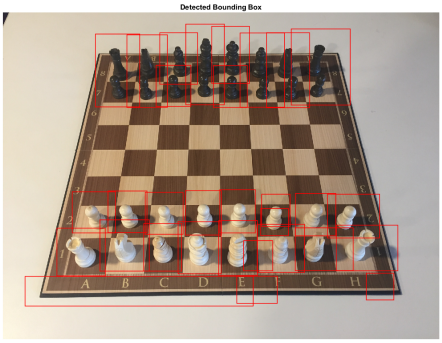

Fig. 13. Detected bounding box around chess pieces

some of the bounding boxes contain two chess pieces which reduces the accuracy of our classifier used to identify the chess pieces . To rectify this issue we took an ideal case scenario where the input image was taken from an overhead angle, reducing the problem of occlusion, and ran it through the same process, the blobs detected were cropped around their bounding box to later use them as test images for the classifier (Fig. 14). We fed our cropped images through the

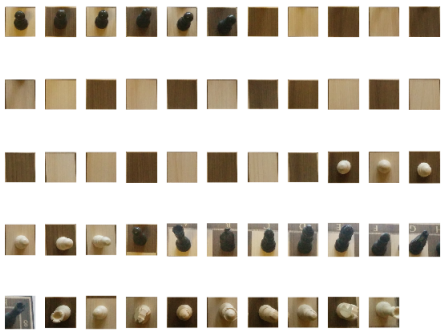

Fig. 14. Cropped images of detected blobs

trained Convolutional Neural Network, the classifier based on the score of each class in relation to the image, assigns the label to each of our test images. Using the positions of our cropped images relative to the chessboard image we overlay the labels that our classifier has assigned onto the input image.

V. Discussions

As shown in the previous section, the approach performed well for a chessboard where the color of the pieces and the chessboard squares were very similar. The adverse lighting conditions increased the similarity in color and but did not affect the performance of clustering based segmentation. Although the system performs well for different boards and lighting conditions, the images segmented sometimes

TABLE II

LAYERS OF CONVOLUTIONAL NEURAL NETWORK

1	input	image input
2	conv1	Convolution
3	relu1	ReLU
4	norm1	Cross Channel Normalization
5	pool1	Max Pooling
6	conv2	Convolution
7	relu2	ReLU
8	norm2	Cross Channel Normalization
9	pool2	Max Pooling
10	conv3	Convolution
11	relu3	ReLU
12	conv4	Convolution
13	relu4	ReLU
14	conv5	Convolution
15	relu5	ReLU
16	pool5	Max Pooling
17	fc6	Fully Connected
18	relu6	ReLU
19	fc7	Fully Connected
20	relu7	ReLU
21	fc8	Fully Connected
22	prob	Softmax
23	classification Layer	Classification Output

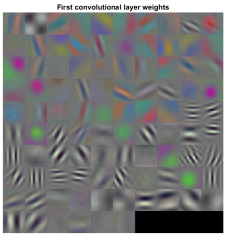

Fig. 15. Computed weights from the first layer of CNN

contained chess pieces in pairs. This can be improved by detecting edges using operators like Sobel and then separating the detected blobs using subtraction. Another issue faced was occlusion when the pieces were hidden behind one another. The occluded pieces could be recognized based on already detected pieces and the probability of them occurring under that observation. Overall by segmenting the board out of the image and using clustering, the approach fares better than the controlled techniques used in [1] and [5].

To evaluate the performance of our classifier, we used a SIFT based classifier from a different chessboard and ran

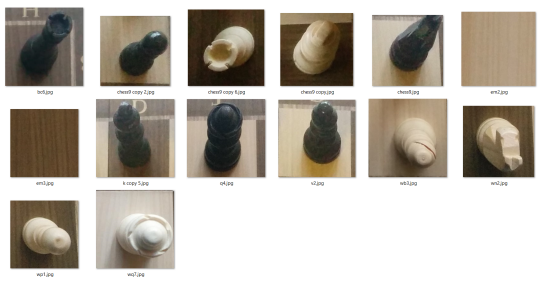

Fig. 16. Sample of training images used

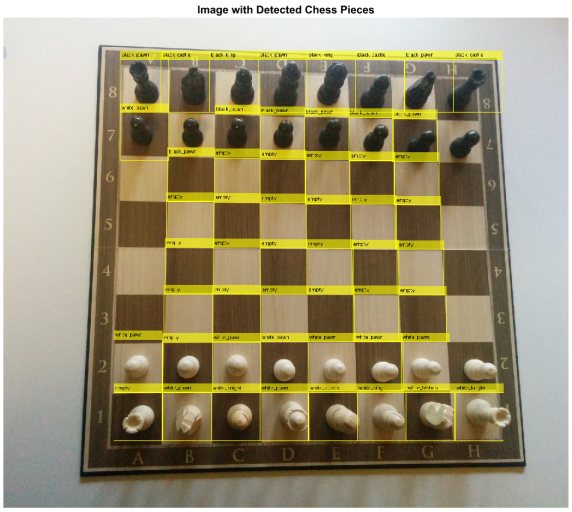

Fig. 17. Recognized and labeled chess pieces

TABLE III

TABLE SHOWS COMPARISON OF ACCURACY RESULTS

Method Applied	**Accuracy on their board**	**Accuracy on our board**
SIFT & HOG descriptors	84.6%	61.2%
Fourier Descriptor	90%	Board not recognized
K-means and CNN	67.3%	69%

it on our detected pieces. We did the same with our CNN and ran it on both the chessboards. Our CNN based approach fared uniformly for both images, whereas the SIFT approach showed relatively fewer true positives for our chessboard image. This achieves our aim to build a generic classifier using deep learning. To improve the accuracy of the classifier, training should be done with more images under various conditions like lighting and orientation. Additionally, two separate classifiers can be trained for projected pieces and original image pieces to obtain better results. Errors due to shadows and reflections can be removed by applying appropriate image enhancement techniques [10].

VI. CONCLUSIONS

In this report, we have presented a system that performs chess recognition irrespective of the kind of board and pieces. The use of clustering based segmentation removes the bias towards using a single type of board and operating under a certain lighting condition. The use of R-CNN enables us to perform detection across different chess sets thus going hand in hand with our aim to build a generic system. Moving forward this system can be extended to identify board configuration and record chess moves. This information when given to a chess engine will help finding the next best move and identify similarity to moves made by professional players. and give visual feedback on possible chess moves.

REFERENCES

[1] J. Ding, ChessVision: Chess Board and Piece Recognition [Online].Available:https://web.stanford.edu/class/cs231a/prev_projects_2016/CS_231A_Final_Report.pdf

[2] H. Soyel and H. Demirel, Improved SIFT matching for pose robust facial expression recognition, In *Automatic Face & Gesture Recognition and Workshops* (FG 2011), 2011 IEEE International Conference on (pp. 585-590). IEEE, 2011, March.

[3] N. Dalal and B. Triggs, Histograms of oriented gradients for human detection, In *Computer Vision and Pattern Recognition*, 2005. CVPR 2005. IEEE Computer Society Conference on , (Vol. 1, pp. 886-893). IEEE, 2005, June.

[4] S. Lawrence, C.L. Giles, A.C. Tsoi and A.D. Back, Face recognition: A convolutional neural-network approach, In *IEEE transactions on neural networks*, 8(1), pp.98-113, 2005, June.

[5] C. Danner and M. Kafafy. (2015). Visual Chess Recognition [Online]. Available: https://web.stanford.edu/class/ee368/ProjectSpring1415/Reports/DannerKafafy.pdf

[6] A. De la Escalera and J. M. Armingol, Automatic chessboard detection for intrinsic and extrinsic camera parameter calibration,In *Sensors*, vol. 10, no. 3, pp. 20272044, 2010.

[7] J. Goncalves, J. Lima, and P. Leitao. Chess robot system: A multi-disciplinary experience in automation. *Proceedings of the 9th Spanish-Portuguese Congress on Electrical Engineering*, 2005

[8] N. Banerjee, D. Saha, A. Singh, and G. Sanyal, A simple autonomous robotic manipulator for playing chess against any opponent in real time.

[9] C. Koray and E. Sumer, A Computer Vision System for Chess Game Tracking, presented at the 21st Computer Vision Winter Workshop, Rimske Toplice, Slovenia, 2016

[10] Jinshan Tang, E. Peli and S. Acton, Image enhancement using a contrast measure in the compressed domain, *IEEE Signal Processing Letters*, vol. 10, no. 10, pp. 289-292, 2003.