Object Recognition System using Template Matching Based on Signature and Principal Component Analysis

Inad A. Aljarrah  
Jordan University of Science & Technology, Irbid, Jordan  
inad@just.edu.jo

Ahmed S. Ghorab  
University College of Applied Sciences, Gaza, Palestine.  
aghorab@ucas.edu.ps

Ismail M. Khater  
Birzeit University, Birzeit, West Bank, Palestine  
ikhater@birzeit.edu

ABSTRACT

In this paper an object recognition system using template matching is implemented. Since objects are represented by either an external or internal descriptors, a combination of using signature, principal component analysis and color is used. The system efficacy is measured by applying it to recognize an image of a chessboard with a set of objects (pieces). The output of the system includes the pieces names, locations and color. The signature feature is used to distinguish the pieces types based on their external shape but when it falls short, the principal components analysis is used instead. The object color is also obtained. The matching between features is carried out based on Euclidean distance metric. The proposed system is implemented, trained, and tested using Matlab based on a set of collected samples representing chessboard images. The simulation results show the effectiveness of the proposed method in recognizing the pieces locations, types, and color.

KEYWORDS

Template Matching, Object Recognition, Signature, PCA, Euclidean Distance.

1 INTRODUCTION

A picture is worth a thousand words, from human beings perspective; the same idea is true for machines. On the other hand, machines unlike humans need the translation of pictures into a machine understandable format. This is achieved by extracting features from images, and then the features are used to classify and recognize objects. The process is completed by comparing these features with previously stored and known features of different objects. Template matching is considered among the main techniques that are used to accomplish this approach [1, 2, 3].

Object recognition systems are employed in many applications, such as industry and assembly lines, robotics, object clustering, object tracking, face recognition and game playing [2, 4, 5]. All of these applications require a computer vision program able to recognize different types of objects as explained in [2, 3, 4]. Machine vision is also needed for robotics to automatically identify different chess pieces on a chessboard and spot their location. A Chinese chessboard is reconstructed in [6] based on binarized Gabor filter and Hough transform. Template matching algorithms tend to use features extracted from boundary of objects for recognition purposes [7]. In some cases the method works very well, but in other cases where there is either a distortion in the boundary or an occlusion to parts of the object, the process fails. In this work, the signature feature is used as a boundary description for object, but for some conditions when it fails, a Principal Components Analysis (PCA) is used as an alternative to help
build a recognition system that is both computationally efficient and highly accurate. For the purpose of testing the proposed approach, a chessboard that contains chess pieces is used. The input to the system consists of a 2-D chessboard images, each image contains a set of objects (chess pieces); these pieces are arranged over the chessboard in different locations. In a chess game there are two opponents each of the opponents have a set of pieces that differ from the pieces of the other opponent only in color; here we have white and black pieces; the chessboard itself is a board that is divided into 8×8 squares (blocks), so we have 8 rows and 8 columns, the rows are numbered with decimal numbers 1, 2, 3, ..., 8 and the columns are labeled with alphabetical characters A, B, C, ..., H. The squares are colored with two colors as a background for the board; each square is colored with a color that differs from the color of its adjacent, and with the same color of the square on its diagonal. A system that is capable of describing the type, the location, and the color of the pieces in the chessboard have been designed and built.

The rest of this paper is prepared as follows: In section II a definition of the template matching has been stated, in section III the overall system scheme has been presented. The case study of chessboard recognition system will be shown in section IV; in addition, this section shows how the system has been trained. Finally, section V illustrates the experimental results and the discussion.

2 TEMPLATE MATCHING

Template matching is an essential task that frequently occurs in image analysis applications. It is the procedure of finding the location of a sub-image (i.e. template) inside another big image. It is conceptually a simple process. We need to match a template to an image, where the template is a sub-image that contains the shape we are trying to find. [8, 9, 10, 11]. In this work we have compared the features extracted from the boundary of objects with the previously stored and known features for recognition purposes. Template matching is considered among the main techniques that are used to accomplish this approach [1, 2, 3], which then can be called features matching instead of template matching. Accordingly, we measure the Euclidian distance between the object features and the stored features (which may be considered as the templates) and calculate the matching between the object feature and the stored features (templates).

The procedure is then repeated for all extracted objects in the entire image, every object in the image is compared with the stored templates, when a match is found, it is deemed to be recognized.

3 OVERALL SYSTEM SCHEME

The general block diagram that describes the operation of the online system is shown in Figure (1). The input to the system is an image, and the output is a description of the objects found in the input image. In the image pre-processing step the region of interest (ROI) is determined, then, the image converted from RGB to gray scale image then to binary image.

In the features extraction step, all features were extracted for all the objects that have been obtained from the previous stage, and then saved as a .mat file for future using.
The gotten features are then matched with the features of the objects that have been saved before (templates). In order to do that, the Euclidean distances between the saved features and the obtained features have been calculated and used to decide whether the object is recognized or not.

Finally, in the last step, the system classifies the objects and puts a description according to the object types. For more convenience, a case study of chessboard recognition system has been adopted to implement this methodology.

4 A CASE STUDY: CHESSBOARD RECOGNITION SYSTEM

4.1 An Overview

In order to distinguish chessboard objects, good features or a set of features are needed. The signature feature was one of the most powerful features used for chessboard recognition system, it can be used alone to recognize the chessmen (chess pieces).

When the signature comes to grief, the PCA (Principal Component Analysis) [12, 13, 14] is used; in this case PCA is considered as an additional feature that supports the signature feature when it fails in recognizing the object with a high percentage of accuracy; this is because the signature of the object is an external descriptor which will be badly affected by any noise, so PCA is adopted which will describe the object as a whole image (i.e. internal descriptor), so both descriptors have been combined.

The location is determined by dividing the chessboard itself into 64 blocks as shown in Figure (8), so we have 8 rows and 8 columns, the rows are numbered with decimal numbers 1, 2, 3, ..., 8 and the columns are labeled with alphabetical characters A, B, C, ..., H. the location of the object is defined by its column and its row, for instance (A1, H3, G4, ...).

The color of the chess piece is determined using a simple method of calculating the percentage of the black pixels in the object as discussed in section (4.4).

4.2 Extracting the Signature Feature

The signature is a one-dimensional function, that can be extracted by several methods, we have used a plot of the distance from the centroid to the boundary of the object as a function of angle[15]; as shown in Figure (2); this reduces the dimensionality of the boundary from 2-D to an easy 1-D signature function.

The signature feature is size variant and relies on scaling, so; the signature values have been normalized; the normalization is done by dividing the signature values by its maximum value. There is no need to find all the points of the signature; just a point every 15 degrees was obtained;
yielding 23 points due to ignoring 0 and 360 degrees points. These points were saved for each object as a vector feature (template) that will be used in the next steps for recognition and matching. The signature is also rotation variant; that means it would look different from different angles except if it is normalized for that by starting from the robust point (angle), so we have started from angle 0 to angle 360 counter clockwise. For each block of the chessboard, it is checked whether it contains a piece or not (i.e. noise), then for each piece, the external border is gotten, then this border is passed to a method to calculate the signature. For instance, the signature of the white rook is shown in Figure (3).

![Figure 3: The signature for white rook object (only 23 angles have been calculated).](image)

### 4.3 Principal Component Analysis – PCA

PCA involves a mathematical process that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components [12, 13, 14]. The first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible. PCA is the simplest of the true eigenvector-based multivariate analyses [3, 12]. Often, its operation can be believed as revealing the internal structure of the data in a way which best explains the variance in the data. If a multivariate dataset is visualized as a set of coordinates in a high-dimensional data space (1 axis per variable), PCA supplies the user with a lower-dimensional picture, a "shadow" of this object when viewed from its most informative viewpoint.

PCA has been used to determine the most discriminating features between images of components. To establish PCA, the system have been trained for 192 sample components, these components are collected from all the samples, then, the extracted Eigen values have been stored, the mean image of the training database, and the matrix of centered image vectors; as features in a (.mat) file; this was performed to prevent re-computing these features every time the system is run.

### 4.4 Object Color

Simply, since the chessboard has only two colors namely black and white the color of the object is determined by counting the number of the black pixels and the number of the white pixels. This process is feasible due to the fact that the chessboard will be converted to binary image as shown in Figures (7 and 8). The number of black pixels (B) and white pixels (W) is obtained as follows:

\[
B = \sum_{r,c \in R} \text{black pixels} \\
W = \sum_{r,c \in R} \text{white pixels}
\]

Where r and c represent the rows and columns of image R respectively.
The percentage of the black pixels (P) in the object is calculated using the following equation:

\[ P = \frac{B}{B+W} \% \]  

(3)

This percentage (threshold) was gotten for all possible objects in the training set, and the following thresholds were concluded by observing the cutting edges between the black and the white objects.

When \( P \geq 0.24 \), the object will be black, when \( 0.19 \geq P \geq 0.7 \) the object will be white, when \( P = 0 \) the image will be empty, and otherwise the object will be noise.

5 EXPERIMENTAL RESULTS & DISCUSSION

The suggested algorithm has been applied on different samples of the chessboards, for the purpose of explanation; the result of one sample image is shown in different stages of the algorithm.

The complete flow chart of the system is described in Figure (4); First of all, the RGB image was read, then, the ROI (i.e. the board) is obtained, after that, it was converted to gray level as shown in Figure (6).

![Figure 5: The input RGB image.](image1)

Next, the gray level image was converted to a binary image with a threshold of 25%; an opening operation was done. The result image is as shown in Figure (7).

![Figure 6: The gray level of the input image.](image2)

Next, the gray level image was converted to a binary image with a threshold of 25%; an opening operation was done. The result image is as shown in Figure (7).

![Figure 7: The binary image.](image3)
Then, the board was sliced into 64 blocks as shown in Figure (8).

![Figure 8: Slicing the board into 64 blocks.](image)

For each slice, we have checked whether it contains a piece or not (noise or empty). After preparing each slice alone, each object was processed to obtain its signature (section 4.2), and by calculating the Euclidean distance between the computed signature and all the stored signatures (templates), the minimum Euclidean distance is obtained which indicates the best match.

If the calculated Euclidean distance does not give a correct indication in case of noisy objects, then the PCA is used as an alternative in the matching process, this alternative is carried out because the signature of the object is an external descriptor which will be badly affected by noise; while PCA will describe the object as a whole image (i.e. internal descriptor), so both descriptors have been combined.

As described before; to employ PCA, the system has been trained using 192 objects, and the extracted Eigen images have been stored as features. The normalized training set is shown in Figure (9).

![Figure 9: The training set of the PCA.](image)

Samples of the Eigen images are shown in Figure (10). And finally the resulted mean image is shown in Figure (11).

![Figure 10: Samples of the Eigen images of the PCA.](image)

![Figure 11: The mean image of the PCA.](image)

The input block image has been projected against the Eigen faces and the system reconstructed the image as shown in Figure (13); this image is then used to recognize the object of the input image using the Euclidean distance between the reconstructed image and the saved known objects.
The color of the object is determined by equation (3) as described in section (4.4) and the results shows that when \( P \geq 0.24 \), the object will be black, when \( 0.19 \geq P \geq 0.7 \) the object will be white, when \( P = 0 \) the image will be empty, and otherwise the object will be noise.

The block location is determined based on the index of the block in the blocks matrix that we have used. Figure (14) shows an example of the results.

A testing set has been collected; which contains 107 samples; each sample is one of the expected objects listed in Table(1), the system output results show that the system recognized 103 samples correctly with an overall percentage of 96.3%.

The results in Table (1) show the type of the error for each sample. The only type error occurred in the case of white-bishop which is recognized as a white-pawn; this is due to the similarity of their shapes.

The other three errors resulted from the color recognition method, like in the case that the white-queen was recognized as a black-queen. As shown in Figure (8); the white-queen has a lot of lines which will increase the percentage of the black pixels (reaches 25%); so it has been recognized as a black-queen.

Finally, the signature technique comes to grief in 45 samples with a percentage of 42.1% of the overall samples.

<table>
<thead>
<tr>
<th>Object (B-Black, W-White)</th>
<th>Correctly recognized samples / total samples</th>
<th>Percent %</th>
<th>Type of error</th>
</tr>
</thead>
<tbody>
<tr>
<td>BKing</td>
<td>7/7</td>
<td>100</td>
<td>-</td>
</tr>
<tr>
<td>WKing</td>
<td>6/6</td>
<td>100</td>
<td>-</td>
</tr>
<tr>
<td>BKing</td>
<td>11/11</td>
<td>100</td>
<td>-</td>
</tr>
<tr>
<td>WKing</td>
<td>11/11</td>
<td>100</td>
<td>-</td>
</tr>
<tr>
<td>BBishop</td>
<td>9/9</td>
<td>100</td>
<td>-</td>
</tr>
<tr>
<td>WBishop</td>
<td>9/10</td>
<td>90</td>
<td>type</td>
</tr>
<tr>
<td>BQueen</td>
<td>7/7</td>
<td>100</td>
<td>-</td>
</tr>
<tr>
<td>WQueen</td>
<td>6/9</td>
<td>66.7</td>
<td>All in color</td>
</tr>
<tr>
<td>BPawn</td>
<td>12/12</td>
<td>100</td>
<td>-</td>
</tr>
<tr>
<td>WPawn</td>
<td>10/10</td>
<td>100</td>
<td>-</td>
</tr>
<tr>
<td>BRook</td>
<td>7/7</td>
<td>100</td>
<td>-</td>
</tr>
<tr>
<td>WRook</td>
<td>8/8</td>
<td>100</td>
<td>-</td>
</tr>
<tr>
<td>Overall</td>
<td>103/107</td>
<td>96.3%</td>
<td>3 in color and 1 in type</td>
</tr>
</tbody>
</table>

6 CONCLUSIONS

An object recognition system using template matching to recognize a set of objects from an image is implemented. The system takes an image as an input
and generates a description of that image as an output. The system efficiency was measured by applying it to recognize an image of a chessboard with a set of objects (pieces). The output of the system includes the pieces names, locations and colors. The system can recognize multiple objects based on the features extracted. The simulation results give 96.3% accuracy. As a future work, different issues can be considered as an improvement to this work. Chessboard computer assisted systems are meant to deal with real-world real-time processing of chessboard games; the proposed system in this paper uses still images, it can be improved to deal with real-time chess games. Other improvements may include using rotation invariant features and adding more features.

7 REFERENCES