ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

Object Recognition Using Moment Descriptors and Genetic Algorithm

[I]Mahmood ul Hassan, [II]Muteb Alruwaili, [III]Jawad Usman Arshed, [IV]Khalid Mahmood Ali Asghar
[I,II]College of Science & Arts Tabarjal, Al Jouf University, Tabarjal, Saudi Arabia
[III]Karakoram International University, Gilgit, Baltistan, Pakistan
[IV]College of Science & Arts Mahayil Asir, King Khalid University, Saudi Arabia

Abstract

In many image analysis and computer vision applications, object recognition is the ultimate goal. This work presents study and experimentation for object recognition when isolated objects are under discussion. The circumstances of similarity transformations, presence of noise, and occlusion have been included as the part of the study. For simplicity, instead of objects, outline of the objects have been used for the whole process of the recognition. Hu's moments and their extended counterparts have been used as features of the objects. Genetic Algorithm is applied on these moments to get optimum descriptors. These optimum descriptors are then used in process of recognition. Various similarity measures have been used and compared for recognition. The test objects are matched with the model objects in database and the object with the least similarity measure is taken as the recognized object. A detailed experimental study has been made under different conditions and circumstances including transformation, noise, and occlusion. Databases of different sizes have been used to have a look at various experimentations. Some interesting observations have been made which may be useful for research and practicing community.

Keywords

Moment, Genetic Algorithm, Object Recognition, descriptors.

I. Introduction

Object recognition [5-9] is one of the important areas of research. Moment invariants [1-5, 8-12] have been frequently used as features for image processing, remote sensing, shape recognition and classification. Moments can provide characteristics of an object that uniquely represent its shape. Several techniques have been developed that derive invariant features from moments for object recognition and representation. These techniques are distinguished by their moment definition, such as the type of data exploited and the method for deriving invariant values from the image moments. It was Hu [1] that first set out the mathematical foundation for two-dimensional moment invariants and demonstrated their applications to shape recognition. They were first applied to aircraft shapes and were shown to be quick and reliable by [4]. These moment invariant values are invariant with respect to translation, scale and rotation of the shape. The moments which have the property of invariant image recognition as well as image reconstruction was introduced by [3]

Hu [1] defined seven of these shape descriptor values computed from central moments through order three that are independent to object translation, scale and orientation. Translation invariance is achieved by computing moments that are normalized with respect to the centre of gravity so that the center of mass of the distribution is at the origin (central moments). Size invariant moments are derived from algebraic invariants but these can be shown to be the result of simple size normalization. From the second and third order values of the normalized central moments, a set of seven invariant moments can be computed which are independent of rotation.

The idea of Hu's moments was extended in 1993. Flusser and Suk [8] proposed additional features that are invariant under general affine transformations and may be used for recognition of affine-deformed objects. Addition of these four moments to the Hu's seven moments make a class of moment-based features invariant to image rotation, translation, scaling, other changes. This paper has used these 11 features with the combination of Genetic Algorithm, which optimizes these features for best recognition rate. The experiments are done with different combinations of these features, for the recognition of objects captured by a non-ideal imaging system which may transform, make noise or can have occlusion in the images. An extensive experimental study has been made using various similarity measures in the process of recognition. These measures include Euclidean Measure and Percentage error. Although the whole study has been made for bitmap images, but it can be easily extended to gray level images.

The outline of the remainder of the paper is as follows. Moments are explained in Section 2. The concept of similarity measures are explained in Section 3. Genetic Algorithm is explained in Section 4. The Algorithm for object recognition problem has been concocted in Section 5. Detailed experimental study and analysis is made in Section 6 whereas Section 7 deals with interesting observations during the experimental study. Finally, Section 8 concludes the paper as well as touches some future work.

II. Moments

Traditionally, moment invariants are computed based on the information provided by both the shape boundary and its interior region [1]. The moments used to construct the moment invariants are defined in the continuous but for practical implementation they are computed in the discrete form. Following paragraphs provide a brief description of moments [1,9-10].

Moment invariants [8-16] have been frequently used as features for image processing, remote sensing, shape recognition and classification. Moments can provide characteristics of an object that uniquely represent its shape. Several techniques have been developed that derive invariant features from moments for object recognition and representation. These techniques are distinguished by their moment definition, such as the type of data exploited and the method for deriving invariant values from the image moments.

The discrete version of the Cartesian moment for an image consisting of pixels P_{xy} is:

$$m_{pq} = \sum_{x=1}^{M}\sum_{y=1}^{N} x^p y^p P_{xy} \quad . \qquad (1)$$

m_{pq} are the two dimensional Cartesian moments where M AND N are the image dimensions and the monomial product $x^p y^q$ is the basis function. The zero order moment m_{00} is defined as the total mass (or power) of the image. If this is applied to a binary (i.e. a silhouette) $M \times N$ image of an object, then this is literally a pixel count of the number of pixels comprising the object.

$$m_{00} = \sum_{x=1}^{M}\sum_{y=1}^{N} P_{xy} \qquad (2)$$

The two first order moments are used to find the Centre Of Mass (COM) of the image. If this is applied to a binary image and the results are normalized with respect to the total mass (m_{00}), then the result is the centre co-ordinates of the object. Accordingly, the centre co-ordinates $\bar{x}$, $\bar{y}$ are given by:

I.

$$\bar{x} = \frac{m_{10}}{m_{00}}, \ \bar{y} = \frac{m_{01}}{m_{00}} \qquad (3)$$

The COM describes a unique position within the field of view which can then be used to compute the centralized moments of an image.

The definition of a discrete centralized moment as described by Hu [1] is:

$$\mu_{pq} = \sum_{x=1}^{M}\sum_{y=1}^{N} (x-\bar{x})^p (y-\bar{y})^q P_{xy} \qquad (4)$$

This is essentially a translated Cartesian moment, which means that the centralized moments are invariant under translation. To enable invariance to scale, two dimensional scale-normalized centralized moment are used [11] and are given by:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \ , \qquad (5)$$

where

$$\gamma = \frac{p+q}{2} + 1 \ , \quad \forall (p+q) \geq 2 \ .$$

HU [1] recognized that rotation invariance is the most difficult to achieve and proposed two different methods for computing rotationally invariant moments. The method used for rotation invariance through principal axis is based on the observation that moments may be computed relative to a unique set of principal axes of distribution and will therefore be invariant to the orientation of the distribution. The principal axis moments are obtained by rotating the axis of the central moments until μ_{11} is zero [5, 9-10].

In special case, HU [1, 9] defined seven values computed from central moments through order three, that are invariant to object scale, position, and orientation. In 1993, Flusser and Suk [8] proposed additional features that are invariant under general affine transformations. This leads, altogether, to 11 moments which may represent the features of the objects. We will denote these moments as $\phi_1, \phi_2, ..., \phi_{11}$.

III. Similarity Measures

Given two sets of descriptors, how do we measure their degree of similarity? An appropriate classification is necessary if unknown shapes are to be compared to a library of known shapes. If two shapes, A and B, produce a set of values represented by a(i) and b(i) then the distance between them can be given as c(i) = a(i) – b(i). If a(i) and b(i) are identical then c(i) will be zero. If they are different then the magnitudes of the components in c(i) will give a reasonable measure of the difference. It proves more convenient to have one value to represent this rather than the set of values that make up c(i). The easiest way is to treat c(i) as a vector in a multi-dimensional space, in which case its length, which represents the distance between the objects, is given by the square root of the sum of the squares of the elements of c(i).

This paper implements different simple classifiers that calculate different similarity measures of the corresponding descriptors of the input shape and each of the shapes contained in the database as shown in Figure 1.

The similarity measures, attempted for experimental studies, are as follows:

$$\sqrt{\sum_{i=1}^{n} c(i)^2} \qquad \text{(Euclidean Distance (ED))} \qquad (6)$$

$$\sum_{i=1}^{n} \left| \frac{c(i)}{b(i)} \right| \qquad \text{(Percentage of Error (PE))} \qquad (7)$$

In this study, n is the number of moments considered, $a(i)$ is the ith moment of the template image, and $b(i)$ is the ith moment of the test image. A tolerable threshold ρ is selected to decide a test object recognized. This threshold is checked against the least value of the selected similarity measure, which lead to the success or failure as far as recognition of the object is concerned.

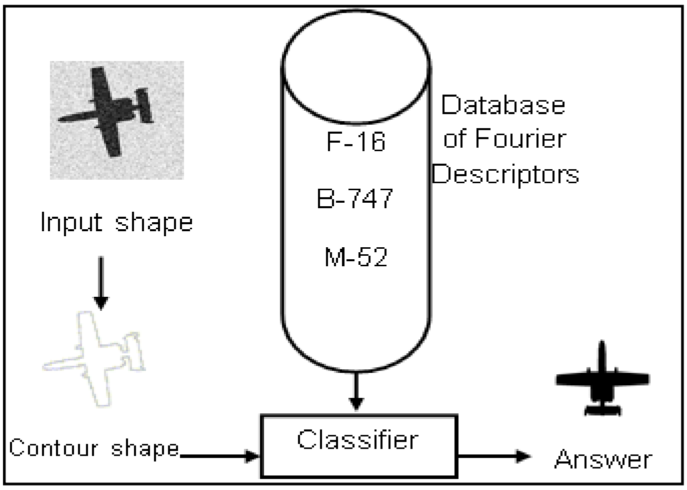


Fig. 1: Pictorial description of the method.

IV. Genetic Algorithm

Genetic algorithm is belonging to the class of evolutionary algorithm that uses selection, crossover and mutation for the process of generating new population. GA is extremely popular evolutionary algorithm which is used to solve complex problems in a very meaningful way. The aim of genetic algorithm is to find exact and appropriate solution to a given problem. GA has been rapidly used in many areas such as pattern recognition, object recognition, image processing, machine learning, computer vision etc. GA produced surprisingly very good results over moments. Proposed technique works as follows [17].

1. Initialization: - The first step for Genetic algorithm in the optimization process is initialization. In this step various parameters are initialized to their desired value. In our simulation we have set the following parameters.

- Bias is set between 0 and 0.2.
- Number of iterations are 40 but we also have set the stopping criteria. If our stopping criteria meet during the specified iterations the simulation ends otherwise it go for the number of iterations specified.
- Numbers of trials are used in between 10 to 20.
- We have also set the stopping criteria which depends on the Hits used in the simulation.

2. Evaluate Fitness function:

The second step of Genetic algorithm is to evaluate the fitness of each particle as in our case we have the weights used as particles, so we compute fitness against each of the particle. Fitness against each particle shows the relative importance of those weights, and is given by:

$$f = 1 - \min(PE) \tag{8}$$

Where f is the fitness and *PE* is the percentage of errors of all the training images for a given set of weights.

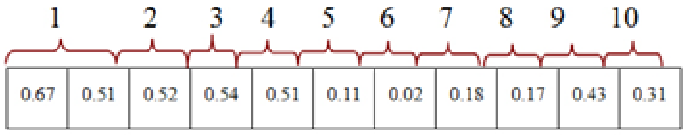
3. In third step stopping criteria has to be checked, if stopping criteria is met then best generation will be obtained and GA will be terminate else jump to step iv

4. Apply crossover:

Crossover is the most important operator of GA. Crossover is applied on previous population. After applying crossover we get new population and again here checked the hit ratio. If current hit ratio is better than previous than previous, population will be replaced with newly generated population, otherwise step iv will be repeated until best hit ratio is achieved. Initially we have assigned random weights. Let us suppose the random weights are

0.67	0.51	0.52	0.54	0.51	0.11	0.02	0.18	0.17	0.43	0.31

Now we have to find out the binary of each decimal number. After that for applying crossover in the sense that we have to make pairs of above random weights and then apply crossover on each pair (First pair "0.67 and 0.51", second pair "0.51 and 0.52" and so on) which are as under

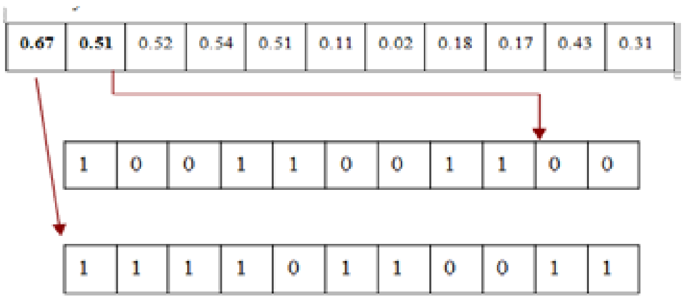


We will convert each decimal number into binary then we have to apply the crossover on the binary of first pair. For instance we have the first pair which is 0.67 and 0.51 and their binaries are

0.67 = 10101011101

0.51 = 10011001100

Now when we apply the crossover on this pair in the sense that we will have to take the first binary number of the first decimal number and last ten binaries of the second decimal number. In the same way we will take the first two binaries of the first decimal number and last nine binaries of the second decimal number. This process will remain continue until we take the ten binaries of the first decimal number and one binary of the second decimal number.



After crossover of these two pairs, we have the following new combination.

As a result we will get new matrix of "10 X 11" which is as under

1	0	0	1	1	0	0	1	1	0	0
1	1	0	1	1	0	0	1	1	0	0
1	1	1	1	1	0	0	1	1	0	0
1	1	1	1	1	0	0	1	1	0	0
1	1	1	1	0	0	0	1	1	0	0
1	1	1	1	0	1	0	1	1	0	0
1	1	1	1	0	1	1	1	1	0	0
1	1	1	1	0	1	1	0	1	0	0
1	1	1	1	0	1	1	0	0	1	0
1	1	1	1	0	1	1	0	0	1	0

New weights (1 X 100)

Then we will get ten new weights of first pair after converting each row into decimal. Since we have ten pairs so we will get 100 new weights after applying the crossover on all the pairs which are as follows:

0.59
0.84
0.97
-
-
-
-
-
0.67
0.74

New weights (1 X 100)

Now we have to apply the crossover on the original weights (random) and the new weights.
For this we have to take the first weight from the new weights and the last ten weights from the original weights. Then we will get the new generation (weights) which are as follows

1 → Index

0.59	0.51	0.52	0.54	0.51	0.11	0.02	0.18	0.17	0.43	0.31

Now we have to check the hit rate of the new generation whether it is better than the previous (original) generation. If the hit rate of new generation is better than the previous generation then the previous generation will be replaced by the new generation.

As we took the first weight from the new weights then in the same way it will go up to 100 but we have to take the last ten weights from the original weight in each step.

2 → Index

0.84	0.51	0.52	0.54	0.51	0.11	0.02	0.18	0.17	0.43	0.31

3 → Index

0.97	0.51	0.52	0.54	0.51	0.11	0.02	0.18	0.17	0.43	0.31

100 → Index

0.74	0.51	0.52	0.54	0.51	0.11	0.02	0.18	0.17	0.43	0.31

As we had taken one by one weight from the new weights, now we have to take weights in pairs from the new weights like

1 → **2**

0.59	**0.84**	0.52	0.54	0.51	0.11	0.02	0.18	0.17	0.43	0.31

99 → **100**

0.67	**0.74**	0.52	0.54	0.51	0.11	0.02	0.18	0.17	0.43	0.31

But during each step we have to keep on checking the hit rate. Similarly we have to take different combinations up to 11. At the end we will get new weights which will better than the original weights.

v) Apply Mutation

If stopping criteria matched then terminate else apply mutation on newly created population and check for hits.
After that we have to find out the binary of each weight.
Then we will mutate the first bit of the first weight. By doing so we will get new decimal number (new weight) and now we have to check the hit rate, if current hit rte is better than previous then we will replace this new weight with the first weight of the previous weights.
Likewise we will mutate each bit. The same process will be applied on the remaining weights. Finally we will have new better generation (weights).
The fitness for each individual try to search the best weights for each individual and check for hit rate makes the genetic algorithm arrive at the highest recognition with optimum descriptors.
vi) If current hit ratio is better than previous, population will be replaced with newly generated population, otherwise step (vi) will be repeated until best hit ratio is achieved.
vi) Now again check here for stopping criteria, if found then terminate otherwise start searching from the start.

V. Algorithm for Object Recognition

The outline of the algorithms is as follows:

1. Clean up the image of noise by using a median filter and then removing all but the largest of the objects in the scene.
2. Find the edge of the image using the edge detector.
3. Find the moments of order three using the Equation 2
4. Find the center of gravity of the image
5. Find the central moments, which eliminates the translation effect.
6. Normalize the moments by dividing them by the m_{00} (size of the image) for removing the scaling effect.
7. Find the rotational invariant moments ϕ_1, ϕ_1, ..., ϕ_{11}
8. Apply Genetic Algorithm to find optimum no of moments.
9. For a given test image, its moment invariants are compared with all the model objects moment invariants using either the Euclidean distance (or the sum of percentage errors) between them.
10. The model object with least Euclidean distance (or least percentage error) is a recognized object.

VI. Results and Analysis

The recognition system is tested by generating the test objects by translating, rotating, scaling, adding noise, and adding occlusion to the model objects contained in a database of different sizes. The test objects were randomly rotated, translated, and scaled. Some were considered without scale of their model sizes. A large data of test objects was used for each of the experiments for testing similarity transformation. Translation, scaling, and rotation were used for the images to test for recognition.
The salt & pepper noise of different densities is added to the objects for generating the noisy test objects. Median filter was used in the experiment to filter the noise, so that the noise remains on the boundary of the object. Median filtering is a type of neighborhood processing that is particularly useful for removing 'salt and pepper' noise from an image [12]. The median filter considers each pixel in the image and it looks at its nearby neighbors to decide whether

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

or not it is representative of its surroundings. Instead of simply replacing the pixel value with the ***mean*** of neighboring pixel values, it replaces it with the ***median*** of those values. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value [12].
As would be seen in the experiments, Moment invariants are not promising for the recognition of occluded objects. Around 20% occlusion was added into the objects to make tests.
Genetic Algorithm is applied on the database of moments obtained from different model objects. The database of these model object moments is trained with transformed objects by using the Genetic algorithm which yields weights for the moments. Genetic Algorithm produces optimized moments by producing weights for the moments which when applied in the process of recognition; it not only increases the recognition rate but also reduces the no of moments used in the process of recognition.
We split the experiments into different categories explained in the following subsections.

A. Experiment I
The first series of experiments has been made to view results for different combinations of the Moment invariants. Five experiments can be seen presenting different scenarios of the combination of Moments, similarity measures and nature of data used. The procedures taken to analyze and test the system are as follows:
Case 01 (M0): In this case, the moment invariants $\varphi 1 - \varphi 7$ are used and the Euclidean distance is considered for comparison.
Case 02 (M1): In this case, the moment invariants $\varphi 1 - \varphi 7$ are used and the PE, instead of the Euclidean distance, is considered as a similarity measure for the sake of comparison.
Case 03 (M2): The effect of similarity transformations on the moment invariants have been studied closely. It has been clear that $\varphi 7$ deviates greatly due to these transformations. Therefore, $\varphi 7$ has been excluded and only $\varphi 1 - \varphi 6$ have been considered.
Case 04 (M3): In this case $\varphi 1 - \varphi 11$ are used as features and PE is considered for comparison.
Case 05 (M4): In this case $\varphi 1 - \varphi 11$ are used except $\varphi 7$ and PE is considered for comparison.

Table 1: Recognition rate for the different cases trained with transformed objects

	Object recognition Using Moments				Object Recognition Using Moments & GA		
	T	N	O		T	N	O
M0	45%	62.50%	7%	M0	58%	74%	9%
M1	56.70%	75%	8%	M1	68%	85%	9%
M2	73%	87.50%	10%	M2	85%	95%	10%
M3	60%	81.25%	9%	M3	70%	91%	9%
M4	75%	81.25%	9%	M4	86%	91%	9%

T: Transformations
N: Noise
O: Occlusion

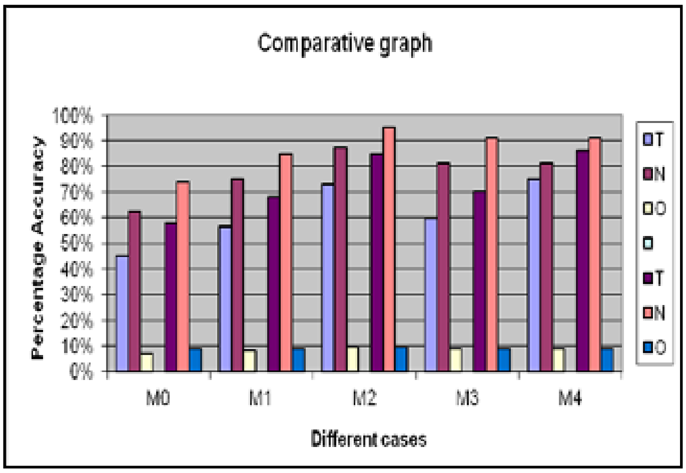


Fig. 2: Comparative Graph.

1. The base case (M0): In this case, the moment invariants $\phi_1 - \phi_7$ are used and the Euclidean distance is considered for comparison. The percentage of recognition recorded with optimized weights in case of just similarity transformations is 58% which was only 45% when using moments without optimization (increase of 13%). Similarly in case of similarity transformations with noise, it is about 74% with optimized weights and is only 62.50% without optimized weights. In case of occluded objects the recognition rate increases from 7% to 9%.
2. Using the percentage of error (PE) for comparison (M1): In this case, the moment invariants $\phi_1 - \phi_7$ are used with PE, instead of the Euclidean distance as similarity measure.
The use of percentage error also increases the recognition rate from 56.70% to 68% for transformed objects and from 75% to 85% for noisy objects and for occluded objects it increases from 8% to 9%. It can be concluded here that although the use of optimization increases the recognition rate but PE for comparison also have better results as compared to Euclidean distance.
Excluding ϕ_7 (M2): It has been determined through experiments that ϕ_7 deviates greatly due to these transformations. Therefore, ϕ_7 has been excluded in this case and only $\phi_1 - \phi_6$ have been considered. The percentage of recognition recorded in case of just similarity transformations increases from 73% to 85%. In case of similarity transformations with noise, it increases up to 7.5% and remains same for occluded objects. It is worth noting that in all cases discussed above, there is minor increase in recognition for occluded objects as compared to transformed and noisy objects.
Adding ϕ_8-ϕ_{11} (M3): In this case $\phi_1 - \phi_1$ are used as features and PE is considered for comparison. The percentage of recognition recorded in case of just similarity transformations is 70%, an increase of 10% from non-optimized moments. In case of similarity transformations with noise, it also increases 10%. And for occluded objects it remains the same. If we compare this case with (M1), it can be clearly seen that the addition of ϕ_8-ϕ_{11} increases the recognition in both cases non-optimized and optimized moments.
Excluding ϕ_7 (M4): In this case $\phi_1 - \phi_1$ are used excluding ϕ_7 .PE is used for comparison. It can be noticed that this case give much better result from all other cases the recognition rate in case of just similarity transformations is 86% which is much better than all the above cases. There is an increase of 11% from non-optimized moments. In case of similarity transformations with noise, it is about 91% (an improvement of about 10%). If we compare this

case with (M2) where only $\phi_1 - \phi_6$ are used and PE is considered for comparison. It can be clearly seen that the recognition rate is improved by only 1% in the case of similarity transformation only and is reduced by about 4% for noisy images. Occlusion tests do not report good results; this can be observed throughout in Table 1. This Table summarizes all the above discussed results and demonstrates numerical measures.

B. Experiment II

In this category of experiments, just before examining the test images, moment database is loaded and the moments are normalized. This is done by using GA in order to give optimized weights for each moment. The Euclidian distance between the templates' moments and the target is computed as a similarity measure. The results achieved are very good as compared to previous experiments. One hundred images were tested and obtained the results as mentioned in Table II.

Table II: Recognition rates for the different cases.

Moments	10% noise 10 degree rotation	20% noise 20 degree rotation	10 noise 10degree rotation 0.25scale	Arbitrary (different type of transformation ions)	Ocluded	Noise
First 7	95%	75%	8%	93%	15%	85%
11	95%	75%	5%	90%	12%	81%

VII. Some observations

Here are some observations for the whole discussion in the paper:

In this paper we have been noticed that different combinations of Moment Invariants were giving relatively different success results and were less unanimity among the different combinations of moments in recognizing. Combination of first five and combination of first six moments almost shows the same level of recognition accuracy level. Similarly seven and eight are at the same level; and nine and ten also for a pair that stays at the same level of performance. Noise (salt and pepper) has a minimal effect on the recognition ability of Moment Invariants. The affect in recognition is there, but very slightly. Moment Invariants by using GA perform not so well in the presence of occlusion in the image. The occlusion is a big issue on recognition object, especially, when we use Moment descriptors.

VIII. Conclusion and Future Work

This work has been reported to make a practical study of the Moments and Genetic Algorithm to the application of Object Recognition. Genetic Algorithm is used for the optimization of these Moments. The implementation was done on a P-IV PC using MATLAB 7.1.0. The ultimate results have variations depending upon the selection of Moments and Data size. The variety similarity measures and different combinations of invariant moment features, used in the process, make a difference to the recognition rate. The results have been tested using 2 similarity measures, 11 moment invariants, and different size databases. Different combinations of these parameters implied different results. Three similarity measures, including ED, and PE, provided different recognition results. The images used are all bitmapped images, further investigations are being done with some more complex images.

Several modifications to the base case are investigated in Section 5 and compared it with the results of non-optimized moments. It is shown that when Genetic algorithm is applied on these Moments it increases the recognition rate more than 10%. The replacement of Euclidean distance with the percentage of error for moment comparison results also have a very good impact and have an improvement of more than 10% in the recognition rate. One more factor is that of ϕ7. Excluding ϕ7 gives rise to an improvement of about 20% over the base case. Moreover, the effect of adding four more moments is considered. In this way the use of Genetic Algorithm and use of PE, excluding ϕ7 gave recognition rate up to 86% for transformed and 91 for noisy objects. The results achieved are very good as compared to previous results. However there is minor increase for occluded objects.

The images that have to be recognized but failed to be recognized by most of the Moment Invariant combinations are to be analyzed further. This leads to the theory of optimization to find out appropriate features or attributes in the image that made it difficult to be recognized. The methodology of GA has been utilized successfully for this purpose. Using GA, to find the most suitable descriptors and to assign weights for these descriptors, improved dramatically the recognition rate using the least number of descriptors.

In future, author would like to treat the problem as multi-objective optimization method, also try to enhance the GA by tuning different parameters to maximize the recognition rate while minimizing the number of descriptors.

IX. AcknowledgEment

First and foremost, the authors would like to thank the ALLAH Almighty, who guides us always in the path of knowledge and wisdom. We thank the editors and anonymous reviewers for their valuable comments to significantly improve the quality of this paper.

We are very much grateful to the Research Panel of **Al- Jouf University** who helped a lot to complete this research work.

We are also thankful to the King Khalid University for their support in research.

References

[1] *G. H. Granlund, Fourier Preprocessing for hand print character recognition, IEEE Trans. Computers, Vol C- 21, Feb. 1972, pp. 195-201.*

[2] *M. Sarfraz, Object Recognition using Fourier Descriptors: Some Experiments and Observations: Computer Graphics, Imaging and Visualization – Techniques and Applications, Banissi, E., Sarfraz, M., Huang, M. L., and Wu, Q. (Eds.), ISBN: 0-7695- 2606- 3, IEEE Computer Society, USA, 2006, pp. 281- 286.*

[3] *A New Kernel Method for Object Recognition: Spin Glass-Markov Random Fields BARBARA CAPUTO Doctoral Thesis Stockholm, Sweden 2004.*

[4] *G. H. Granlund, Fourier Preprocessing for hand print character recognition, IEEE Trans. Computers, Vol C-21, Febr. 1972, pp. 195-201.*

[5] *A Project led by Julien Boeuf and Pascal Belin, andsupervised by Henri Maître: http://www.tsi.enst.fr/tsi/enseignement/ressources/mti/descript_fourier/index.html*

[6] *M. Sarfraz, Object Recognition using Moments: Some*

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

Experiments and Observations, Proceeding of the International Conference on Geometric Modeling and Imaging (GMAI), IEEE Computer Society Press, 2006.
[7] C.T. Zahn, R.Z. Rhoskies, Fourier descriptors for plane closed curves, IEEE trans. Compu. 21 (1972) 269-281.
[8] M-K. Hu, Visual Pattern Recognition by Moment Invariants, IRE Trans. Inf. Theory IT-8, pp.179-187, 1962.
[9] Y. Li, Reforming the theory of invariant moments for pattern recognition, Pattern Recognition, Vol. 25, pp. 723-730, 1993.
[10] A. Khotanzad and Y.H. Hong, Invariant image recognition using Zernike Moments, IEEE Transactions Pattern Analysis and Machine Intelligence, Vol.16(10), pp. 976-986, 1994.
[11] Dudani, S. A., Breeding, K. J., McGhee, R.B., Aircraft identification by moment invariants, IEEE-T on Computers, Vol. 26(1), pp. 39-46, Jan. 1977.
[12] W.H. Wong, W.C. Siu, K.M. Lam, Generation of moment invariants and their uses for character recognition, Pattern Recognition Lett. Vol.16, pp. 115-123, 1995.
[13] J. Flusser, T. Suk, Pattern recognition by affine moment invariants, Pattern Recognition, Vol. 26, pp. 167-174, 1993.
[14] http://homepages.inf.ed.ac.uk/rbf/ CVonline/LOCAL_COP IE S/SHUTLER3/
[15] L. L. Yudell, Mathematical functions and approximations, pp. 432, Academics Press Inc, 1975. http://www.cee.hw.ac.uk/hipr/html/median.html
[16] Dudani, S. A., Breeding, K. J., McGhee, R.B., Aircraft identification by moment invariants, IEEE-T on Computers, Vol. 26(1), pp. 39-46, Jan.1977.
[17] M. Sarfaraz, Mehmood-ul-Hassan, M.Iqbal "Object Recognition using Fourier Descriptors and Genetic Algorithm" socpar, pp. 318-323, International Conference of Soft Computing and Pattern Recognition, 2009.

Authors' Profile

Mahmood ul Hassan is currently a Lecturer in Computer Science department, College of Science and Arts Tabarjal, Al-Jouf University, Saudi Arabia. He received the M.S. degree from COMSATS University Abbottabad, Pakistan. From 2008 to 2010, he was a Research Assistant in COMSATS Abbottabad, Pakistan.

Muteb Alruwaili has MSc Computer science from Glamorgan University,UK. He has been Lecturer in Computer department at Al Jouf university, Saudi Arabia.

Jawad Usman Arshad

Mr. Jawad Usman Arshed serving as Assistant Professor in Department of Computer Science, Karakoram International University, Gilgit Baltistan. I have done MS in Computer Science from COMSATS Abbottabad. My field of interest is Image Processing, Computer Vision and Semantic web.

Khalid Mahmood is currently a Lecturer in Information Systems department, College of Science and Arts Mahayil Asir at King Khalid University, Saudi Arabia. He received his MS in Computer Science degree from the department of Computer Science, International Islamic University Islamabad, Pakistan, in the year 2011.