Preeti Malhotra* and Dinesh Kumar

An Optimized Face Recognition System Using Cuckoo Search

DOI 10.1515/jisys-2017-0127

Received October 3, 2016; previously published online July 26, 2017.

Abstract: The development of an effective and efficient face recognition system has always been a challenging task for researchers. In a face recognition system, feature selection is one of the most vital processes to achieve maximum accuracy by removing irrelevant and superfluous data. Many optimization techniques, such as particle swarm optimization (PSO), genetic algorithm (GA), ant colony optimization, etc., have been implemented in face recognition systems mainly based on two feature extraction methods: discrete cosine transform (DCT) and principal component analysis (PCA). In this research, a nature-inspired well-known algorithm, namely cuckoo search, has been implemented for face recognition. Further, a hybrid method consisting of DCT and PCA is applied to extract the various features by which recognition can be made with a high rate of accuracy. To validate the proposed methodology, the results are also compared with the existing methodologies, such as PSO, differential evolution, and GA.

Keywords: Principal component analysis, cuckoo search, genetic algorithm, particle swarm optimization, face recognition, differential evolution, discrete cosine transform.

1 Introduction

One of the most active research areas in the last two decades is face recognition (FR), which consists of multiple fields like signal processing, computer vision, and pattern recognition. These multiple disciplines are due to several applications in perfunctory indexing of image databases: identity authentication, human-machine interaction, security system, and video surveillance [3]. Many strategies for FR have been developed, as reported in the literature [35].

Training and recognition are two main processes of any FR system. Both of these processes involve various steps. The first step is preprocessing. In the second step, features are extracted through a feature extraction step, which is followed by a feature selection step. The next steps are creation of template data and classification. Among all these stages, feature extraction is one of the essential stages that influence the performance of the system. For an FR system, a good feature extractor must be designed in such a way that it can select the best features (features that are not sensitive to some random environmental variations like pose, variation in scale, facial expressions, and illumination).

Feature-based technique and appearance-based technique are two main categories of feature extraction algorithms for FR. In the feature-based technique, geometric characteristics like distance and angles between various face components are to be matched and compared; on the other hand, in the appearance-based technique, the whole face is considered as input and compared with such multiple faces [12].

There are many well-known FR techniques. Some of these are discrete cosine transform (DCT), Fisher's linear discriminant analysis [20], eigenface method [22], independent component analysis [2], and discrete wavelet transform [17], etc. Eigenface method is one of the appearance-based methods that use principal component analysis (PCA). It is a standard method among all the methods mentioned above.

*Corresponding author: **Preeti Malhotra**, Department of Computer Science and Engineering, Guru Jambheshwar University of Science and Technology, Hissar, Haryana, India, e-mail: preetianeja2005@gmail.com

Dinesh Kumar: Department of Computer Science and Engineering, Guru Jambheshwar University of Science and Technology, Hissar, Haryana, India

Feature extraction involves building a new set of features from the original feature set in order to reduce dimensions from d to m . Feature extraction methods not only retain the relevant information but also present most of the information with fewer and most discriminative features. Feature selection looks for the most favorable set of n features out of total m features, which results in a good recognition rate for the classifier where $n \leq m$ [6].

Thus, many metaheuristic algorithms inspired by nature have been developed to solve the problem of optimization [4]. These algorithms are widely used to solve engineering optimization problems with a complex nature [5]. These are based on physical systems and biological behavior in nature. Examples of such algorithms are genetic algorithm (GA) [9], immune algorithm, differential evolution (DE) algorithm [11, 13, 24], cuckoo search (CS), particle swarm optimization (PSO) [7, 8, 10, 16, 29], ant colony optimization (ACO), harmony search, and many more. These algorithms work on the basis of a mechanism that improves the solution vector at each step, capitulate optimal design parameters, and overcome the computational drawbacks of traditional mathematical optimization methods [28, 30]. Now, researchers are paying more attention toward hybrid algorithms in solving optimization problems. Hybrid algorithms have shown outstanding reliability and efficiency when applied to engineering optimization problems [25–27, 32, 33].

Moreover, Yang and Deb [23] proposed a new metaheuristic model for continuous optimization (i.e. CS algorithm), which is based on the proliferation tactic of cuckoo birds. It is based on the fact that many species lay their eggs in the nests of other host birds. This approach has confirmed the production of good results when compared with other nature-inspired techniques [31, 34].

In this paper, an optimized FR system is discussed and represented using a CS-based feature selection approach. To discover an effective and efficient solution space, the CS algorithm is used, which provides the optimal feature subset. This algorithm is used as a feature selection algorithm and selects the best features out of the features extracted using the hybrid DCT-PCA method. Minimum Euclidean distance is used as a fitness function in the CS algorithm, which repetitively helps in selecting the best nest.

The main contribution of this paper is the implementation of an optimization algorithm for the feature selection problem of an FR system, which is not only simple due to the presence of a single parameter, P_a , but also efficient in producing results when compared with other population-based algorithms. Other contributions of our work are as follows: (i) due to the simplicity, our approach does not increase the overall complexity of CS; (ii) our approach does not destroy the structure of CS, so it is still very simple; and (iii) our approach produces good recognition rate both in the absence and presence of noise.

The paper is structured as follows. An overview of the hybrid DCT-PCA-based feature extraction method is described in Section 2. Section 3 briefly introduces the CS algorithm, while Section 4 covers in detail the CS-based feature selection algorithm. The performance of the CS-based feature selection algorithm is tested and analyzed in Section 5. In Section 6, we come to a conclusion.

2 Hybrid Method for Feature Extraction

In the literature, it has been noticed that two or more techniques are needed to improve the performance of a system to a great extent [15]. This integrated method is called the hybrid method. Two methods used in this paper to represent the face accurately are DCT and PCA [18]. Both methods aim to reduce the dimensions of data. The extraction of meaningful features helps the hybrid approach increase the efficiency and recognition rate of the system. The methodology for improving the accuracy of the FR system is presented in this paper. First, we use DCT to compress the input image and then PCA is used to extract the features.

2.1 DCT

In 1974, Ahmed et al. [1] proposed the DCT technique, which is used in imaging and video compression. However, its first-time use was conducted on image compression. When applied to an input sequence, it

decomposes input into a weighted sum of basis cosine sequences. DCT produces the coefficients that are used to recreate the original signal using inverse DCT easily.

$f(x, y)$ is an input image of size $m \times n$. The two-dimensional DCT coefficients are as follows:

$$c(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} f(x, y) \cos \frac{\pi(2x+1)u}{2m} \cos \frac{\pi(2y+1)v}{2n}, \quad (1)$$

where

$$\alpha(u) = \sqrt{\frac{1}{m}} \text{ for } u = 0$$

$$\alpha(u) = \sqrt{\frac{2}{m}} \text{ for } u = 1, 2, \dots, m-1$$

and

$$\alpha(v) = \sqrt{\frac{1}{n}} \text{ for } v = 0$$

$$\alpha(v) = \sqrt{\frac{2}{n}} \text{ for } v = 1, 2, \dots, n-1.$$

The space domain coordinates are the variables m and n , and the frequency domain coordinates are the variables u and v .

2.2 PCA

Extraction of feature matrix is the first and important step of the FR system. One of the oldest feature extraction techniques developed by Turk and Pentland [22] is PCA, which is used to extract informative and non-redundant features.

Consider a set of faces represented by F vector $F = \{f_1, f_2, f_3, \dots, f_N\}$. (2)

Next, we calculate the average face $\bar{F}$ and subtract it from the vector F as follows:

$$F' = F - \bar{F}, \quad (3)$$

Where

$$\bar{F} = \frac{1}{N} \sum_N F_i.$$

We compute the scatter matrix using

$$S_T = \sum_N (F_i - \bar{F})(F_i - \bar{F})^T. \quad (4)$$

Now, we find the eigenvectors of the scatter matrix. The eigenvectors and the eigenvalues take the following form:

$$\begin{aligned} \bar{V} &= [V_1, V_2, V_3, \dots, V_m] \\ \lambda &= [\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_m] \end{aligned} \quad (5)$$

k eigenvectors should be kept corresponding to the k largest eigenvalues.

For given values of eigenvectors, $U=[V_1, V_2, V_3, \dots, V_K]$, we project the test image f_{test} onto the eigenvectors to match the test image eigenvectors with training images. Those training images are determined on the basis of Euclidean distance, whose weight is found nearest and considered to be the best match for identification.

$$\delta_d = U_d^T f_{\text{test}} \text{ for } d = 1, 2, \dots, d. \quad (6)$$

If PCA is applied to the original face images (face dimension size 92×112), it will produce a covariance matrix of larger size that cannot be calculated with the majority of computers because of less memory. Therefore, we use a hybrid approach in which the face images are first DCT transformed. Then, a selected block size of the DCT coefficients is introduced to the PCA, resulting in a lower-dimensional feature vector that carries the most prevalent features.

3 CS Algorithm

Cuckoos are touchy in nature due to their attractive sound and imitation approach, and they are known as charming birds. Some female cuckoos lay their eggs into the host bird's nest and remove the eggs of the host bird in order to increase the growth probability of their own eggs. The color of the eggs placed in the host bird's nest changes to that of the host bird's eggs. If, by chance, the host bird identifies that the eggs in the nest are different from its own eggs, then it will destroy either the cuckoo eggs or the whole nest. This process led to the emergence of the CS algorithm.

Generally, cuckoo birds lay their eggs earlier than the host bird in order to create space for their own eggs. This is also to ensure that a large part of the host bird feed is received by their chicks.

A standard CS algorithm can be described using three ideal assumptions:

1. The cuckoo selects a random nest and puts one egg at a time.
2. The best nest will proceed to the next generations.
3. There are fixed numbers of host nests, and the probability of the host bird identifying the cuckoo egg is P_a .

$$P_a \in [0, 1].$$

To generate a new solution, $W^{(t+1)}$ for cuckoo i , a Levy flight is performed as follows:

$$W_i^{(t+1)} = W_i^t + \alpha \oplus \text{Levy}(\lambda), \quad (7)$$

where α is the step size linked with the size of the problem ($\alpha > 1$). The symbol $\oplus$ denotes bit-wise multiplication. λ is the scaling parameter. However, Levy flight is based on random walk in which the step size is generated on the basis of the current position and the evolution possibility of next states. The Levy flight distribution can be implemented by many ways [19]. The Mantegna algorithm is one of the methods for symmetric Levy distribution. Step length L used in Mantegna algorithm is calculated by

$$L = \frac{p}{|q|^{1/\alpha}}, \quad (8)$$

where p and q denote the normally distributed stochastic variables with standard deviations.

$$\sigma_p = \left[\frac{\Gamma(1+\alpha) \sin(\pi\alpha/2)}{\Gamma\left(\frac{1+\alpha}{2}\right) \alpha 2^{(\alpha-1)/2}} \right]^{1/\alpha} \text{ and } \sigma_q = 1.$$

The appropriate step size should be chosen for proper functioning of Levy flights; otherwise, it will produce new solutions that will jump outside the design domain.

Figure 1: Flowchart of the CS Algorithm.

Figure 1 represents the CS algorithm flowchart.

4 CS-Based Feature Selection

The main function of the CS algorithm is to find the most representative subset of features. Host nests are represented by features extracted from DCT-PCA. To handle the progress of the algorithm, Euclidean distance is used as a fitness function. The pseudo-code for CS-based feature selection is as follows [21]:

1. Input the face images for the FR.
2. Feature extraction: extract features using the hybrid DCT-PCA method.
3. Feature selection: select the most significant features, assigning the parameters as follows:
 - n : number of host nests
 - C : no. of cuckoos in a nest
 - t : generation step
 - $MaxGen$: maximum generation
 - a. Generate initial population of n host nests, X_i ($i=1, 2, \dots, n$).
 - b. *while* ($t < MaxGen$)
 - {

- ```

for ($i=0$; $i \leq C$; $t++$)
{
 Move the cuckoo to the new nest with step size L .
 Fitness F_i is calculated.
 Randomly select nest j .
 if ($F_t > F_j$)
 $F_j = F_i$
}

A fraction of worst nests with probability P_a is disposed, and new ones are built.
The current best solution is held.
The current best solution is shifted to the next generation.
}

c. Promote the nest having the maximum fitness to the next generation.
4. Classifier: Euclidean distance may be defined as the distance between two points measured in Euclidean space.

```

$$ED = \sqrt{\sum_{i=1}^N (X_i - Y_i)^2}, \quad (9)$$

where  $X_i, Y_i$  are the coordinates of points in dimension  $i$ .

## 5 Experimental Results

Figure 2 shows the block diagram of the proposed FR system. The diagram presents the processing of an input image in two different stages: (i) training stage and (ii) recognition stage.

The ORL gray scale face database [14] is used to evaluate the performance of the proposed CS-based algorithm. It consists of 40 distinct persons each having 10 different images. These images vary in terms of facial expressions and facial details. The dimensional size of an image is  $92 \times 112$  pixels, with 256 gray levels per pixel.

For the experiment, first training data were histogram equalized, then features were extracted using the DCT-PCA method. In order to reduce dimensions, only 50% of features were retained. The next step is to select the optimal features for this CS algorithm. The test results of the CS algorithm were compared with the GA-, DE-, and PSO-based algorithms. During the experiments, the values of different parameters were varied and optimal results were obtained on these values, as shown in Tables 1–4. In our experiments, we have considered two different cases.

**Case 1:** We took five images from each class as training data and the remaining five as testing data.

Figure 2: Proposed FR System.

**Table 1:** CS Parameters.

|                                          |     |
|------------------------------------------|-----|
| Number of nests                          | 15  |
| Probability of abandoning nest ( $P_a$ ) | 0.4 |
| Number of iterations                     | 25  |

**Table 2:** PSO Parameters.

|                           |     |
|---------------------------|-----|
| Swarm size $N$            | 15  |
| Cognitive parameter $C_1$ | 2   |
| Social parameter $C_2$    | 2   |
| Inertia weight $w$        | 0.6 |
| Number of iterations      | 25  |

**Table 3:** GA Parameters.

|                             |     |
|-----------------------------|-----|
| Population size             | 15  |
| Crossover probability $P_c$ | 0.5 |
| Mutation probability $P_m$  | 1   |
| Number of iterations        | 25  |

**Table 4:** DE Parameters.

|                              |      |
|------------------------------|------|
| Population size              | 15   |
| Crossover probability $P_c$  | 0.2  |
| Initial inertia weight value | 0.35 |
| Final inertia weight value   | 0.95 |
| Number of iterations         | 25   |

**Case 2:** Of the images, 40% were taken as training data and the remaining 60% as testing data.

The first experiment was performed for both cases of the ORL face database. Here, we have analyzed the performance of four nature-inspired algorithms, like GA, PSO, DE, and CS, for 10, 20, 30, and 40 classes. The results (Figures 3 and 4) show that as the number of features increases, the recognition rate also increases and eventually reaches a constant value after a number is reached. The results also showed that the proposed method produces better results than other methods, and the proposed method has maximum recognition rate with fewer features. The CS algorithm produces 100% recognition rate with only six features when 10 classes are used. This shows the efficiency of the proposed method.

In the second experiment, the performance of population-based algorithms was analyzed in the presence of noise. Figures 5 and 6 demonstrate the result of noise on different algorithms. The results showed that the CS-based feature selection algorithm is less affected by noise as compared to the GA-, PSO-, and DE-based algorithms.

On the basis of the size of DCT coefficients, the CS-based feature selection algorithm was evaluated in the third experiment. A subset of DCT coefficients corresponding to the upper left corner of the DCT array was retained. The original  $92 \times 112$  DCT array was divided into subsets of sizes  $50 \times 50$ ,  $40 \times 40$ ,  $30 \times 30$ , and  $20 \times 20$ , to act as input to the consequent feature selection phase. In Figures 7 and 8, a comparison among various algorithms is made, showing the recognition rate and execution time with respect to the DCT array vector. We have found that the CS-based feature selection algorithm gives better recognition rate than the PSO-, DE-, and GA-based algorithms for the same number of features. It has been observed that with increase in the size of the DCT array, the computational time also increases.

**Figure 3:** Recognition Rate for Different Numbers of Features for ORL Case 1: (I) 40 Classes, (II) 30 Classes, (III) 20 Classes, and (IV) 10 Classes.

**Figure 4:** Recognition Rate for Different Numbers of Features for ORL Case 2: (I) 40 Classes, (II) 30 Classes, (III) 20 Classes, and (IV) 10 Classes.

It has been further observed, in terms of the recognition rate, that the CS-based selection algorithm gives better recognition rate than the GA, DE, and PSO algorithms.

The time complexity of any algorithm is measured either at compile time or at run time. The compile time of all these population-based algorithms is the same, i.e.  $O(m \cdot n)$ . However, when we measured the run time of these algorithms, less time in the PSO-, DE-, and GA-based feature selection algorithms was observed than in the CS-based selection algorithm. Thus, it can be computationally shown that CS is immoderate compared with DE, PSO, and GA; however, the potential of CS in finding the optimal features compared to PSO, DE, and GA demonstrates its computational inefficiency.

In the next experiment, we have compared the proposed feature selection algorithm with the feature extraction methods shown in Figure 9. It has been noticed that the CS-based feature selection in the present work gives a good recognition rate with a small number of features. Here, we have considered two cases. In

**Figure 5:** Effect of Noise on the Recognition Rate of Different Algorithms for ORL Case 1: (I) 40 Classes, (II) 30 Classes, (III) 20 Classes, and (IV) 10 Classes.

**Figure 6:** Effect of Noise on the Recognition Rate of Different Algorithms for ORL Case 2: (I) 40 Classes, (II) 30 Classes, (III) 20 Classes, and (IV) 10 Classes.

**Figure 7:** (A) Recognition Rate and (B) Execution Time for Dimensions of Various DCT Feature Vectors for ORL Case 1.

**Figure 8:** (A) Recognition Rate and (B) Execution Time for Dimensions of Various DCT Feature Vectors for ORL Case 2.

**Figure 9:** Graph Indicating the Comparative Recognition Rate for CS vs. PCA and CS vs. DCT-PCA for Various Numbers of Features.

the first case, the PCA method gives a recognition rate of 83% using 96 features, whereas a recognition rate of 88% using only 10 features has been procured using the present technique.

For the DCT-PCA technique, a recognition rate of 96.5% using 96 features has been found, whereas the same recognition rate of 96.5% by using only 34 features has been observed with the present method.

## 6 Conclusions

In the present work, feature selection is addressed as an optimization problem and a new technique has been put forth for a better solution of this problem. Initially, the present technique uses DCT-PCA for feature extraction, and afterwards the CS algorithm is used for feature selection.

Various experiments have been performed to evaluate the performance of the CS technique against other nature-inspired algorithms such as GA and PSO under different conditions. It has been observed that the CS-based feature selection technique provides a far better recognition rate as compared to other such methods. It was further observed that the present method used gives better recognition with fewer features not only in the absence but also in the presence of noise. As shown in Figure 9, the present approach (i.e. CS) gives an 88% recognition rate by using only 10 features, which is only 10% of the features used in PCA. Moreover, it gives a 96.5% recognition rate by using only 34 features, which is almost 35% of the features used in the DCT-PCA approach.

The experimental results showed that the CS-based feature selection method is able to produce a good recognition rate with an optimal number of features in the presence of noise as well as with varying numbers of features subset.

## Bibliography

- [1] N. Ahmed, T. Natarajan and K. R. Rao, Discrete Cosine Transform, *IEEE Trans. Comput.* **23** (1974), 90–93.
- [2] M. S. Bartlett, H. M. Lades and T. J. Sejnowski, Independent component representations for face recognition, *Proc. SPIE* **3299** (1998), 528–539.
- [3] R. Chellappa, C. L. Wilson and S. Sirohey, Human and machine recognition of faces: a survey, *IEEE Proc.* **83** (1995), 705–740.
- [4] C. Deepika and J. Nithya, Nature inspired metaheuristic algorithms for multilevel thresholding image segmentation – a survey, *WASET Int. J. Math. Comput. Natural Phys. Eng.* **8** (2014), 1318–1323.
- [5] I. Durgun and A. R. Yildiz, Structural design optimization of vehicle components using cuckoo search algorithm, *Mater. Test.* **54** (2012), 185–188.
- [6] E. Emary, H. M. Zawbaa and A. E. Hassanien, Binary grey wolf optimization approaches for feature selection, *Neurocomputing* **172** (2016), 371–381.
- [7] M. M. M. Farag, T. Elghazaly and H. A. Hefny, Face recognition system using HMM-PSO for feature selection, in: *12th International Computer Engineering Conference (ICENCO)*, Cairo, pp. 105–110, 2016.
- [8] H. Gökdağ and A. R. Yildiz, Structural damage detection using modal parameters and particle swarm optimization, *Mater. Test.* **54** (2012), 416–420.
- [9] M. T. Harandi, Feature selection using genetic algorithm and its application to face recognition, *IEEE Conf. Cybernet. Intell. Syst.* **2** (2004), 1368–1373.
- [10] J. Kennedy and R. Eberhart, Particle swarm optimization, in: *Proceeding IEEE International Conference on Neural Networks*, pp. 1942–1948, 1995.
- [11] R. N. Khushaba, A. Al-Ani and A. Al-Jumaily, Feature subset selection using differential evolution, in: *International Conference on Neural Information Processing (ICONIP 2008): Advances in Neuro-Information Processing*, pp. 103–110, 2009.
- [12] D. Kumar, S. Kumar and C. S. Rai, Memetic algorithms for feature selection in face recognition, in: *Eighth International Conference on Hybrid Intelligent Systems*, pp. 931–934, 2008.
- [13] R. Maheshwari, M. Kumar and S. Kumar, Optimization of feature selection in face recognition system using differential evolution and genetic algorithm, in: *Proceedings of Fifth International Conference on Soft Computing for Problem Solving, Advances in Intelligent Systems and Computing*, 437, pp. 363–374, Springer, 2016.
- [14] ORL Cambridge Face Database, AT&T Laboratories Cambridge, <http://www.cam-orl.co.uk/facedatabase.html>, Retrieved 1994.
- [15] N. Öztürk, A. R. Yıldız, N. Kaya and K. Öztürk, Neuro-genetic design optimization framework to support the integrated robust design optimization process in CE, *Concurr. Eng. Res. Appl.* **14** (2006), 5–16.
- [16] R. M. Ramadan and R. F. Abdel-Kader, Face recognition using particle swarm optimization-based selected features, *Int. J. Signal Process. Image Process. Pattern Recognit.* **2** (2009), 51–65.
- [17] A. S. Samra, S. E. Gad Allah and R. M. Ibrahim, Face recognition using wavelet transform, fast Fourier transform and discrete cosine transform, in: *Proceedings of 46th IEEE International Midwest Symposium Circuits and Systems (MWSCAS'03)*, 1, pp. 272–273, 2003.
- [18] M. Sharika, Application of DCT blocks with principal component analysis for face recognition, in: *International Conference on Signal, Speech and Image Processing*, pp. 107–111, Corfu, Greece, 2005.
- [19] H. R. Soneji and R. C. Sanghvi, Towards the improvement of cuckoo search algorithm, *Int. J. Comput. Inform. Syst. Indust. Manage. Appl.* **6** (2014), 77–88.
- [20] D. L. Swets and J. J. Weng, Using discriminant eigenfeatures for image retrieval, *IEEE Trans. Pattern Anal. Mach. Intell.* **18** (1996), 831–836.
- [21] V. K. Tiwari, Face recognition based on cuckoo search algorithm, *Indian J. Comput. Sci. Eng.* **3** (2012), 401–405.
- [22] M. Turk and A. Pentland, Eigenfaces for recognition, *J. Cognit. Neurosci.* **3** (1991), 71–86.
- [23] X. S. Yang and S. Deb, Cuckoo search via Lévy flights, in: *Proceedings of World Congress on Nature & Biologically Inspired Computing*, pp. 210–214, 2009.
- [24] W. A. Yang, Q. Zhou and K. L. Tsui, Differential evolution-based feature selection and parameter optimisation for extreme learning machine in tool wear estimation, *Int. J. Produc. Res.* **54** (2016), 4703–4721.
- [25] A. R. Yildiz, Hybrid Taguchi-harmony search algorithm for solving engineering optimization problems, *Int. J. Indust. Eng. Theory Appl. Pract.* **15** (2008), 286–293.
- [26] A. R. Yildiz, A new design optimization framework based on immune algorithm and Taguchi's method, *Comput. Ind.* **60** (2009), 613–620.
- [27] A. R. Yildiz, A novel particle swarm optimization approach for product design and manufacturing, *Int. J. Adv. Manuf. Technol.* **40** (2009), 617–628.
- [28] A. R. Yildiz, A comparative study of population-based optimization algorithms for turning operations, *Inform. Sci.* **210** (2012), 81–88.
- [29] A. R. Yildiz, A new hybrid particle swarm optimization approach for structural design optimization in automotive industry, *J. Autom. Eng.* **226** (2012), 1340–1351.

- [30] A. R. Yildiz, Comparison of evolutionary based optimization algorithms for structural design optimization, *Eng. Appl. Artif. Intell.* **26** (2013), 327–333.
- [31] A. R. Yildiz, Cuckoo search algorithm for the selection of optimal machining parameters in milling operations, *Int. J. Adv. Manuf. Technol.* **64** (2013), 55–61.
- [32] A. R. Yildiz, A new hybrid bee colony optimization approach for robust optimal design and manufacturing, *Appl. Soft Comput.* **13** (2013), 2906–2912.
- [33] A. R. Yildiz, Hybrid Taguchi-differential evolution algorithm for optimization of multi-pass turning operations, *Appl. Soft Comput.* **13** (2013), 1433–1439.
- [34] A. R. Yildiz and K. N. Solanki, Multi-objective optimization of vehicle crashworthiness using a new particle swarm based approach, *Int. J. Adv. Manuf. Technol.* **59** (2011), 367–376.
- [35] W. Zhao, R. Chellappa, P. J. Phillips and A. Rosenfeld, Face recognition: a literature survey, *ACM Comput. Surveys* **35** (2003), 399–458.