EVOLUTIONARY FEATURE GENERATION FOR CONTENT-BASED AUDIO CLASSIFICATION AND RETRIEVAL

Toni Mäkinen, Serkan Kiranyaz, Jenni Pulkkinen, and Moncef Gabbouj, *Fellow IEEE*

Department of Signal Processing
Tampere University of Technology
P.O. Box 553, Tampere, Finland
email: *firstname.lastname@tut.fi*

ABSTRACT

Many commonly applied audio features suffer from certain limitations in describing the data content for classification and retrieval purposes. To remedy this drawback, in this paper we propose an evolutionary *feature synthesis* (EFS) technique, which is applied over traditional audio features to improve their data discrimination power. The underlying evolutionary optimization algorithm performs both feature *selection* and feature *generation* in an interleaved manner, optimizing also the *dimensionality* of the synthesized feature vector. The process is based on multi-dimensional *particle swarm optimization* (MD PSO) with two additional techniques: the fractional global best formation (FGBF) and simulated annealing (SA). The experimented classification and retrieval performances over a 16-class audio database show improvements of up to 11% when compared to the corresponding performances of the original features.

***Index Terms*—** Feature generation, particle swarm optimization, neural networks, content-based classification

1. INTRODUCTION

Content-based audio classification and retrieval is a widely studied topic in the field of signal processing and, especially, machine learning. Since the pioneer work of [1], the development of supervised machine learning techniques has led to more advanced and expanded classification methods, such as the one proposed in [2], where *support vector machines* (SVM) were successfully applied. Statistical models, specifically *Gaussian mixture models* (GMM) and *hidden Markov models* (HMM) have also provided satisfactory classification and retrieval results (see e.g. [3] and [4], respectively). The idea in these is to estimate the probability density function (pdf) for the feature vectors of each predefined audio class. Recently, studies related to environmental sounds and context recognition have also emerged. In [5], environmental sounds were indexed and retrieved successfully in both indoor and outdoor conditions using HMMs and a modified spectral clustering algorithm, whereas in [6] event histogram-based context recognition was proposed with a versatile collection of environmental sounds, providing a recognition rate of 92.4%. In addition to context recognition, several other applications can be mentioned for audio indexing and retrieval, such as advanced database browsing, query-by-example, and highlight spotting.

In this work, a distinct *feature generation* phase is to precede the actual audio classification and retrieval. An early work of feature generation (proposed in [7] for digit recognition) suggested taking the originally extracted features and combining those in a proper manner to produce more descriptive *new* (or transformed) features. A similar fundamental idea was applied by Krawiec and Bhanu in [8], where the term *evolutionary feature synthesis* (EFS) was first adopted to describe the use of evolutionary algorithms, such as *genetic programming* (GP), for feature generation purposes. The idea was to encode *potential* object recognition procedures, while the training process consisted of co-evolving feature extraction procedures, each being a *sequence* of elementary image processing and feature extraction operations. The method avoided recurring to the means commonly used in recognition systems, whereas the obtained recognition ratios themselves were not superior to those achieved by standard methods. In [9] and [10], the idea of feature generation was brought into audio domain, as GP was used to produce new (artificial) audio features. Encouraging results were reported e.g. over music genre classification, although only 4 classes were involved.

In this paper, we propose an evolutionary feature synthesis (EFS) technique to enhance common audio descriptors. The technique uses multi-dimensional particle swarm optimization (MD PSO) [11] to search for the optimal feature synthesis parameters among a predefined search space. An initial work of the method was reported in [12] for image retrieval, whereas here the focus is on audio classification (by support vector machines) and retrieval. An overview of an ideal feature synthesis process is illustrated in Figure 1, in which considerable improvements in feature discrimination can be observed after the synthesis operation. Contrary to the figure, in our approach also the output feature vector *dimension* is optimized, which is a property being omitted in the previous feature generation approaches.

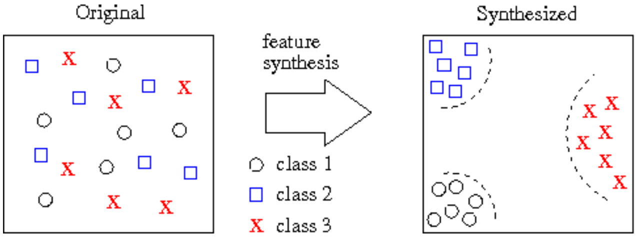

Fig. 1. An illustrative example of an ideal feature synthesis operation over 2-D feature vectors of a 3-class dataset.

The rest of the paper organizes as follows: Section 2 introduces the underlying stochastic optimization algorithm applied in the process, the MD PSO, whereas the feature synthesis process itself is presented in Section 3. The obtained classification and retrieval results are shown in Section 4, and Section 5 concludes the paper.

2. STOCHASTIC OPTIMIZATION ALGORITHM

2.1. Multi-dimensional particle swarm optimization

Particle swam optimization (PSO) was first introduced by Kennedy and Eberhart in [13]. It is a *population-based* optimization technique, in which a swarm of particles propagates iteratively in a predefined search space. After the initialization phase of the algorithm, where the particles are randomly (uniformly) distributed, each particle is evaluated using a proper *fitness function*, $\mathcal{F}[p]$, and moved accordingly within the search space. The ultimate goal is to converge to the global optimum of the search space, for which each particle p has the so-called *social* and *cognitive* terms. The former corresponds to the best position found by the entire swarm (the global best, GB), whereas the latter stands for the best position found by the particle p itself.

In the case of MD PSO, the native PSO operation is extended by allowing the particles to perform *inter-dimensional* jumps within a set dimension range, $d \in [D_{\min}, D_{\max}]$. Thus, the MD PSO searches for the global best solution among *several* search spaces with different dimensions. The dimensional navigation is controlled by a *dimensional* PSO process interleaved with the traditional PSO operations, in which each particle keeps also track of the global and personal best *dimension* (from which the best fitness value so far has been achieved). The pseudo-code and more details about MD PSO can be found in [11].

2.2. Global convergence methods

2.2.1. Fractional global best formation

In order to better avoid local minima during the MD PSO search process, a *fractional global best formation* (FGBF) method [14] is performed within the MD PSO process. The method exploits the potential of individual particle *elements*, evaluating a separate fitness score for each. It then produces a new *artificial global best* (aGB) particle by combining the best elements found from the entire swarm. Whenever the new aGB particle surpasses in fitness the native global best, the aGB is considered as the new global best. In the case of MD PSO, a separate aGB is assigned for *every* dimension. Thus, as illustrated in Figure 2, the aGB particle can be formed by combining elements collected from several dimensions, which further increases the probability of finding aGB particles with improved fitness scores.

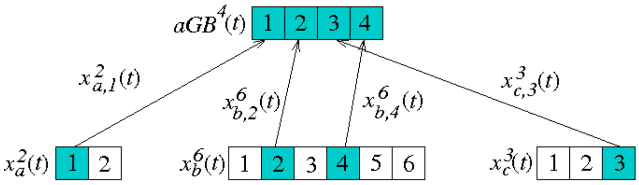

Fig. 2. An illustration of the formation of an aGB particle in dimension 4. Elements of three different particles, a, b, and c (of dimensions of 2, 6, and 3) are combined in the process.

2.2.2. Simulated annealing

As suggested in [15], a *simulated annealing* (SA) algorithm can be utilized within the PSO process to search around the current global best position found by the swarm. In short, after each PSO iteration, a new "neighbor" solution is suggested, which may then replace the current global best. The process is controlled by a specific *temperature* term, T_k, an *update* constant u, and a *cooling* constant, $C < 1$. The number of iterations, $k_{\max}$, needs to be assigned for the SA algorithm, for which the pseudo-code is given in Table 1.

3. EVOLUTIONARY FEATURE SYNTHESIS

3.1. Overview of the system

To meet the objectives assigned for an ideal feature synthesis process, i.e. to perform an optimal feature *selection* and *modification* in an optimal output feature vector dimension, four processing steps are performed. For

Table 1. The SA algorithm in the MD PSO process

```
1. Randomly distribute the particles into the search space.
2. Evaluate the fitness of each particle, $\mathcal{F}[p], p \in [1, P]$.
3. For ($d = D_{\min}$; $d < D_{\max}$) {
    3.1 Set $T_k = T_0$; $k = 0$;
    3.2 While ($k < k_{\max}$) {
        3.2.1 Generate a neighbor solution, $nGB^d$:
            $nGB^d = GB^d + randn(d) * u$;
            // ($randn(d)$ is a $d$-dimensional random vector)
        3.2.2 Evaluate the fitness of $nGB^d$
        3.2.3 Compute $\Delta = \mathcal{F}[nGB^d] - \mathcal{F}[GB^d]$;
        3.2.4 If (min[1, exp($-\Delta/T_k$)] > rand[0,1])
            Set $GB^d = nGB^d$;
        3.2.5 Set $T_{k+1} = CT_k$;   // ($C$ is the cooling constant)
    }
}
4. Update the particle positions, see [11] for details.
5. If (PSO iterations left) return to 2.
```

each new synthesized feature, the system, with a specified *synthesis depth* value K,

1. selects $K+1$ original features $f_0, \dots, f_K$,
2. scales the selected features with weights $w_0, \dots, w_K$,
3. selects K operators, $\theta_1, \dots, \theta_K$, to be applied over the selected and scaled features, and
4. bounds the output with a non-linear operator (here *tangent hyperbolic* is applied).

Suppose $\theta_n(f_a, f_b)$, where $n \in [1, K]$, stands for applying a specific operator θ_n over the features f_a and f_b. Then, a formula for synthesizing a new feature s_j can be defined as

$$s_j = \tanh[\theta_K(\theta_{K-1}(\dots \theta_2(\theta_1(w_0 f_0, w_1 f_1), w_2 f_2), \dots), w_K f_K)], \tag{1}$$

that is, first the operator θ_1 is applied to the *weighted* features f_0 and f_1, after which the operator θ_2 is applied to the *result* of the first operation and the weighted feature f_2, and so on. Finally, the operator θ_K is applied to the result of *all* the previous operations and the weighted feature f_K.

The term "evolutionary" applied in this work refers to both the underlying computing technique, the MD-PSO, as well as the nature of the feature synthesis process itself, which can be performed in either one or several *runs*. Here the idea is that each additional run can further synthesize the features from the previous run and further increase the discrimination power. A block diagram of the overall synthesis process is illustrated in Figure 3, where R synthesis runs are performed.

3.2. Particle encoding

In a MD PSO process, the search space dimension, $d \in [D_{\min}, D_{\max}]$, corresponds to the number of features to be synthesized into the output feature vector (FV), that is, the output FV dimension. Each particle position represents a *potential solution* on how to perform the synthesis for the original features. For this, each particle position encapsulates a complete set of synthesis *parameters*: the indices of the selected features, the feature weights, and the selected operators. Accordingly, the j^{th} *element* of the position of a particle p corresponds to a *way* of synthesizing the j^{th} feature of the output feature vector. Thus, each positional element must include the following: $K+1$ feature indices, $K+1$ feature weights, and K operators. For this, the positional elements of each particle are encoded as a vector of length $2K+1$, including $K+1$ "A-type" and K "B-type" components. These define the corresponding synthesis parameters as follows:

$$\begin{aligned} f_n &= \lfloor A_n \rfloor + 1, & n &\in \{0, K\}, \\ w_n &= A_n - \lfloor A_n \rfloor, & n &\in \{0, K\}, \\ \theta_n &= \lceil B_n \rceil, & n &\in \{1, K\}, \end{aligned} \tag{2}$$

where the $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ operators correspond to the *floor* and *ceiling* mathematical integer functions, respectively. The value ranges for the components can be defined based on the *input* feature vector dimension, F, and the total number of operators available, Θ, as $A_n \in [0, F[$ and $B_n \in]0, \Theta]$. The weight values are limited to $0 \le w_n < 1$.

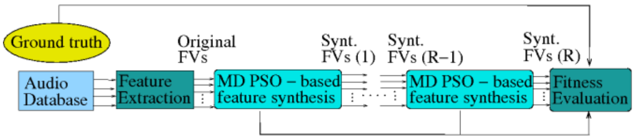

Fig. 3. A block diagram of the proposed EFS with R runs. The arrows correspond to distinct feature vectors (FVs).

To give an example of the encoding, Figure 4 presents a particle p position in dimension 6 with the corresponding synthesis process. The synthesis process of the first element of the output feature vector at run r, *FV*(r), is shown in detail, while a similar process is performed separately for all the output vector elements. For simplicity, the synthesis depth value, K, is set to 3, meaning that $K+1=4$ features, $f_0, \dots, f_3$, are selected from input feature vector *FV*(*r-1*). Thus, as is demonstrated in the figure, each of the particle elements include $2K+1=7$ encoded synthesis parameters, $A_0, \dots, A_3$ and $B_1, \dots, B_3$. The dimension of the input feature vector is $F=8$ and the total number of available operators is $\Theta=5$, meaning that the value ranges for the two component types can be defined as $A_n \in [0{,}8[$ and $B_n \in]0{,}5]$. According to Figure 4, the selected features obtained by the underlying MD PSO process are the 7th, 3rd, 1st, and again the 3rd element of the input feature vector, while the corresponding operators are selected as '+', 'min', and '∗'. Thus, performing the synthesis process as given in (1), the 1st element of the output FV is obtained by

$$s_1 = \tanh\left[\min\left(\left(w_0 f_{[7]} + w_1 f_{[3]}\right), w_2 f_{[1]}\right) * w_3 f_{[3]}\right], \tag{3}$$

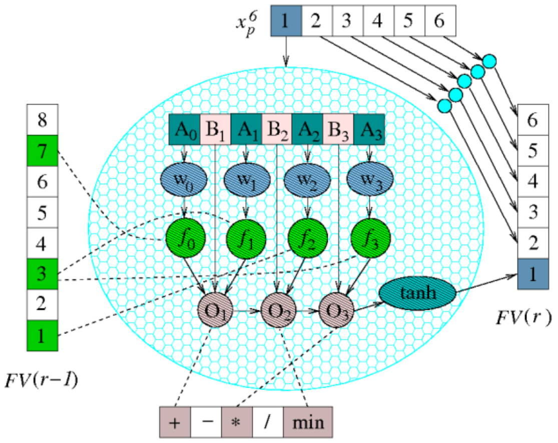

Fig. 4. An example of a particle encoding in a 6-dimensional search space with a synthesis depth set to $K=3$, input feature vector dimension of $F=8$, and the number of operators set to $\Theta=5$.

where $f_{[n]}$ stands for the n^{th} element of the input FV.

Note that by setting $K + 1 = F$, discarding the feature selection as $f_n = f_{[n]}$, and setting each operator θ_n to '+', the approach becomes identical to a *single-layer perceptron* (SLP) classifier. Also, performing several runs with such synthesis parameters corresponds to a *multi-layer perceptron* (MLP) with a one-to-one analogy between the number of hidden layers and the number of runs performed in the synthesis process. In this sense, it can be stated that a regular feed-forward artificial neural network (ANN) is a *special case* of the proposed synthesis approach.

3.3. The fitness function

The fitness measure for evaluating the discrimination ability provided by the synthesized features plays an important role in the whole synthesis process. Here we propose a measure based on clustering criteria. Suppose the different labels of a L-class database are denoted as $l_0, \ldots, l_{L-1}$, and the corresponding class centroids as $\mu_0, \ldots, \mu_{L-1}$. Then, a discrimination measure (DM) can be defined for a set of synthesized feature vectors, $S = \{s\}$, as

$$DM[S] = FP(S) + \delta_{\text{mean}}(l_n)/\delta_{\min}(l_n, l_m),$$

where

$$\delta_{\text{mean}}(l_n) = \frac{1}{L}\sum_{n=0}^{L-1} \frac{\sum_{\forall s \in l_n} \|\mu_n - s\|}{|l_n|}, \tag{4}$$

$$\delta_{\min}(l_n, l_m) = \min_{n \neq m}(\|\mu_n - \mu_m\|).$$

The terms of (4) are defined as follows: $FP(S)$ stands for the number of *false positive* feature vectors occurring among the synthesized feature vectors S (meaning that those feature vectors are actually located in a closer proximity to some other class centroid than their own), $\delta_{\text{mean}}(l_n)$ is the average intra-class distance, and $\delta_{\min}(l_n, l_m)$ corresponds to the minimum centroid distance among all the classes. The $DM[S]$ measure strives for minimizing the intra-class distance, while maximizing the shortest inter-class distance. Ideally, minimizing this fitness measure leads to a situation where each synthesized feature vector is in the closest proximity of its own class centroid, thus leading to a high discrimination among classes as illustrated in Figure 1.

4. EXPERIMENTAL RESULTS

The features obtained by the proposed evolutionary feature synthesis (EFS) technique were tested with classification and retrieval experiments. For this, *three* feature vectors, consisting of *commonly applied* low-level audio features (see e.g. [3], [5]),

- STATISTICS (39-D): audio signal statistics (mean, variance, standard deviation, average deviation, skewness, kurtosis) + band energy ratio, spectral centroid, transition rate, fundamental frequency, irregularity, flatness, and tonality,
- MFCC (39-D): 13$^{\text{th}}$ order coefficients + deltas,
- ACOUSTIC (38-D): tri-stimulus, smoothness, spectral spread, spectral roll-off, RMS, amplitude, inharmonicity, spectral crest, loudness, noisiness, power, odd-to-even ratio, and 6 sub-band powers,

were extracted from a general audio database of 1421 clips. Frame features of 40 ms were first extracted, which were then merged and averaged over longer segments to decrease the total number of feature vectors per clip. The database included 16 pre-defined general audio classes of wide range: *male/female speech, male/female singing, whistling, bird sounds, dog barking, fire sounds, breaking glass, classical/rock/techno music, motorcycle sounds, footsteps, applause,* and *crowd cheering*. The samples were gathered from *TIMIT* corpus (speech), *RWC music database* (music), and *StockMusic.com* net store (environmental sounds). Such a database was selected to demonstrate the EFS scalability and performance with several types (and rather high number) of classes. The EFS parameters were *empirically* set to no. of particles $P = 500$, no. of MD PSO iterations $I = 800$, and synthesis depth $K = 5$. The output dimension range was set to $[D_{\min}, D_{\max}] = [24{,}45]$, while the total number of operators, listed in Table 2 for features f_a and f_b, was set to $\Theta = 18$. Finally, the number of iterations for the SA algorithm was, also empirically, set to $k_{\max} = 25$.

In all the shown experiments, a randomly selected EFS *train* set (45% of the original features) was first used in searching the optimal synthesis parameters (f, w, θ). These were then used to synthesize the actual output features for the *whole* dataset. In the first experiment, the discrimination measure (DM) was evaluated for all the feature vectors before and after the synthesis. The obtained results are presented in Table 3, demonstrating a considerable improvement with all the features (recall that the smaller the DM value, the better the separation between individual classes). This strongly supports the suitability of the EFS to

Table 2. List of operators applied for features f_a and f_b

$\boldsymbol{\theta_n}$	**formula**	$\boldsymbol{\theta_n}$	**formula**
0	$-f_a$	**9**	$f_a * f_b$
1	$-f_b$	**10**	$10(f_a * f_b)$
2	$\max(f_a, f_b)$	**11**	f_a/f_b
3	$\min(f_a, f_b)$	**12**	$\sin\big(100\pi(f_a + f_b)\big)$
4	$f_a * f_a$	**13**	$\cos\big(100\pi(f_a + f_b)\big)$
5	$f_b * f_b$	**14**	$\tan(100\pi f_a * f_b)$
6	$f_a + f_b$	**15**	$\tan\big(100\pi(f_a + f_b)\big)$
7	$10(f_a + f_b)$	**16**	$0.5 * \exp\big(-(f_a - f_b) * (f_a - f_b)\big)$
8	$f_a - f_b$	**17**	$0.5 * \exp\big(-(f_a + f_b) * (f_a + f_b)\big)$

Table 3. The obtained discrimination measures (DM) for the original and synthesized features

	STATISTICS	**MFCC**	**ACOUSTIC**
Orig. DM	4092	10560	3895
Synth. DM	1086	4948	1290

Table 4. The classification error (CE) statistics obtained by SVM over the original and synthesized features

	STATISTICS	MFCC	ACOUSTIC
Orig. CE (%)	25.4 ± 2.8	11.1 ± 1.5	19.1 ± 2.5
Synth. CE (%)	14.4 ± 1.5	10.7 ± 2.0	17.0 ± 1.9

clustering tasks. Next, classification evaluations were performed in a *five-fold* cross-validation manner, so that every sample in the database was tested during the process. The classification was done using SVMs (libSVM, [16]) with *sigmoid* kernel and different parameter combinations, ranging from $\gamma = \{0.5^1, \dots, 0.5^4\}$ and $C = \{2^0, \dots, 2^3\}$. Such a kernel selection makes the SVM model equivalent to a 2-layer-perceptron ANN. As can be seen in Table 4, the smallest average classification errors demonstrate clear improvements with the synthesized features, especially with the STATISTICS features. The MFCCs are designed to be treated more as “a whole”, so that the rather sparse feature selection may diminish the performance obtained with them.

As a final experiment, Table 5 shows the *retrieval* average precisions (AP) obtained for each feature vector. *Precision* stands for the percentage of true positives in the retrieved results, whereas the average was obtained after querying all the database items one by one (by taking the Euclidean distance between the FVs of the query item and each database item). Interestingly, now the MFCCs show an increase of 11% in AP score after 2 EFS runs, and the AP values of other features are also notably improved. Moreover, performing consecutive EFS runs can further enhance the results in most cases. The minor contradiction obtained between the MFCC classification and retrieval performance may suggest that the used fitness function is actually more applicable for clustering than classification.

5. CONCLUSIONS

An evolutionary feature synthesis (EFS) approach for providing discriminative (artificial) features for audio classification and retrieval purposes was proposed. The applied multi-dimensional PSO algorithm proved being able to find the (near-) optimal parameters for the synthesis process. The EFS combines feature selection and feature generation, being also capable of searching the optimal solution among several output vector dimensions. Depending on the original feature vector, the EFS method is capable of improving the classification and retrieval precisions by over 10% with consecutive EFS runs. Testing the method with other databases, classifiers, and fitness functions are potential topics for future research.

Table 5. The average precision (AP) retrieval performances over original and synthesized features

Feature set	Original AP (%)	EFS Run 1 AP (%)	EFS Run 2 AP (%)	EFS Run 3 AP (%)
STATISTICS	44.7	51.4	53.2	54.9
MFCC	35.0	39.4	46.0	44.6
ACOUSTIC	48.5	50.6	52.1	52.6

REFERENCES

[1] E. Wold, T. Blum, D. Keislar, and J. Wheaton, “Content-based classification, search, and retrieval of audio”, *in IEEE Multimedia Journal*, vol. 3, no. 3, pp. 27-36, 1996.

[2] G. Guo and S. Z. Li, “Content-based audio classification and retrieval by support vector machines”, in *IEEE Trans. Neural Networks*, vol. 14, no. 1, pp. 209-215, 2003.

[3] G. Tzanetakis and P. Cook, “Musical genre classification of audio signals”, in *IEEE Trans. Speech and Audio Processing*, vol. 10, no. 5, Jul. 2002.

[4] M. Helén and T. Virtanen, “Audio query by example using similarity measures between probability density functions of features”, in *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2010, ID 179303, 12 p, Jan. 2010.

[5] G. Wichern, J. Xue, H. Thornburg, B. Mechtley, and A. Spanias, “Segmentation, indexing, and retrieval for environmental and natural sounds,” in *IEEE Trans. Audio, Speech and Language Processing*, vol. 18, no. 3, pp. 688–707, 2010.

[6] T. Heittola, A. Mesaros, A. Eronen, and T. Virtanen, ”Audio context recognition using audio event histograms”, in Proc. *18th European Signal Processing Conference (EUSIPCO)*, Aalborg, Denmark, pp. 1272-1276, Aug. 2010.

[7] P.D. Gader and M.A. Khabou, “Automatic feature generation for handwritten digit recognition”, in *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 12, pp. 1256-1261, 1996.

[8] K. Krawiec and B. Bhanu, "Visual learning by evolutionary feature synthesis," in Int. Conf. on *Machine Learning*, pp. 376-383, Washington, DC, Aug. 2003.

[9] I. Mierswa and K. Morik, “Automatic feature extraction for classifying audio data”, in *J. Machine Learning*, vol. 58, no. 2-3, pp. 127-149, 2005.

[10] F. Pachet and P. Roy, “Analytical features: a knowledge-based approach to audio feature generation,” *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2009, Article ID 153017, 23 pages, 2009.

[11] S. Kiranyaz, T. Ince, A. Yildirim, and M. Gabbouj, “Evolutionary artificial neural networks by multi-dimensional particle swarm optimization”, in *Neural Networks*, vol. 22, pp. 1448-1462, Dec. 2009.

[12] S. Kiranyaz, J. Pulkkinen, T. Ince, and M. Gabbouj, ”Multi-dimensional evolutionary feature synthesis for content-based image retrieval”, in Proc. *IEEE Int. Conf. on Image Processing*, Sep. 2011.

[13] J. Kennedy, R Eberhart., “Particle swarm optimization”, in Proc. *IEEE Int. Conf. On Neural Networks*, vol. 4, Perth, Australia, pp. 1942–1948, 1995.

[14] S. Kiranyaz, T. Ince, A. Yildirim, and M. Gabbouj, “Fractional particle swarm optimization in multi-dimensional search space”, in *IEEE Trans. Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 40, no. 2, pp. 298-319, April 2010.

[15] F. Zhao, Q. Zhang, D. Yu, X. Chen, and Y. Yang, “A hybrid algorithm based on PSO and simulated annealing and its applications for partner selection in virtual enterprise”, in *Advances in Intelligent Computing*, vol. 3644, pp. 380-389, 2005.

[16] C.-C. Chang, C.-J. Lin, “LIBSVM: a library for support vector machines”, in *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1-27, Apr. 2011.