Image Classification Technique using Modified Particle Swarm Optimization

Mohd Afizi Mohd Shukran (Corresponding author)
Department of Computer Science, Faculty of Science & Defense Technology
National Defense University of Malaysia (NDUM), Sungai Besi Camp
57000 Kuala Lumpur, Malaysia
Tel: +603 90513400 E-mail: afizi@upnm.edu.my

Yuk Ying Chung
School of Information Technologies, Faculty of Engineering & IT
University of Sydney, 2006 NSW, Australia
Tel: 61-2-9036-9109 E-mail: vchung@it.usyd.edu.au

Wei-Chang Yeh
Advanced Analytics Institute, Faculty of Engineering and Information Technology
University of Technology Sydney, PO Box 123 Broadway, NSW 2007, Australia
Tel: 61-2 9036-9109 E-mail: yeh@ieee.org

Noorhaniza Wahid
Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia
86400 Parit Raja, Batu Pahat Johor
Tel: 60-7-453-8215 Email: nhaniza@uthm.edu.my

Ahmad Mujahid Ahmad Zaidi
Department of Mechanical Engineering, Faculty of Engineering
National Defense University of Malaysia (NDUM), Sungai Besi Camp
57000 Kuala Lumpur, Malaysia
Tel: 60-3 9051-3400 E-mail: mujahid@upnm.edu.my

Received: July 2, 2011 Accepted: July 21, 2011 doi:10.5539/mas.v5n5p150

Abstract

Image classification is becoming ever more important as the amount of available multimedia data increases. With the rapid growth in the number of images, there is an increasing demand for effective and efficient image indexing mechanisms. For large image databases, successful image indexing will greatly improve the efficiency of content based image classification. One attempt to solve the image indexing problem is using image classification to get high-level concepts. In such systems, an image is usually represented by various low-level features, and high-level concepts are learned from these features. PSO has recently attracted growing research interest due to its ability to learn with small samples and to optimize high-dimensional data. Therefore, this paper will introduce the related work on image feature extraction. Then, several techniques of image feature extraction will be introduced which include two main methods. These methods are RGB and Discrete Cosine Transformation (DCT). Finally, several experimental designs and results concerning the application of the proposed image classification using modified PSO classifier will be described in detail.

Keywords: Artificial bee colony algorithm, Data mining, Image classification

1. Introduction

Image classification is an attempt to label (often textual) an image with appropriate identifiers. These identifiers are determined by the area of interest, whether it is general classification for arbitrary pictures (for instance, from the internet), or a specific domain, for instance, medical x-ray images or geographical images of terrain. Image classification is related to *image annotation*, although image annotation generally uses a larger vocabulary (that is, number of classes) than image classification. Image classification is the process of assigning all pixels in a digital image to particular classes according to their characteristics. This characterised data may then be used to produce thematic maps of the image itself. The objective of image classification is to identify and portray, as a unique grey level (or colour), the features occurring in an image. Image classification is perhaps the most important part of digital image analysis. It is very nice to have a *pretty picture,* or an image, showing a magnitude of colours illustrating various features of the images, but it is quite useless unless you know what the colours mean. Therefore, it is essential to extract the features of the images efficiently, without losing important colour information, and reduce redundant colour information. This can be done in two main approaches of image classification: supervised and unsupervised image classification.

Unsupervised image classification does not rely on a training set. Instead, it uses *clustering* techniques which measure the distance between images, and groups the images with common features together. This group can then be labelled with different class-identifiers. Unsupervised classification can be defined as the identification of natural groups or structures within the data. It clusters pixels in a data set based only on their statistics, without using previous knowledge about the spectral classes present in the image. Some of the more commonly used unsupervised classification methods are: Isodata (Witten & Frank, 2005) and k-Means (Witten & Frank, 2005). Moreover, unsupervised classification is a method which examines a large number of unknown pixels and divides them into a number of classes based on natural groupings present in the image values. Unlike supervised classification, unsupervised classification does not require analyst-specified training data. The basic premise is that values within a given colour pixel should be close together in the measurement space (i.e. have similar grey levels), whereas data in different classes should be comparatively well separated (i.e. have very different grey levels) (Lillesand & Kiefer, 1994).

Besides that, supervised classification uses *training sets* of images to create descriptors for each class. The training sets are carefully manually selected to represent a common picture set of that class. The classifier method then analyses the training set, generating a descriptor for that particular class based on the common features of the training set. This descriptor could then be used on other images, which determines if that image is a part of that class. Supervised image classification is a subset of *supervised learning*. Supervised learning can generate models of two types. Most commonly, supervised learning generates a global model that inputs objects to desired outputs. In some cases, however, the map is implemented a set of local models. These local models are treated as inputs in such algorithms. Such algorithms are often implemented using *neural networks*, *decisions trees*, *support vector machines* and *Bayesian statistical methods*. The support vector machines and Bayesian statistical methods show a great promise in this area.

Moreover, supervised classification is a process of using samples of known identity (training data) to classify pixels of unknown identity (Lillesand & Kiefer, 1994). The training data is used to train the classifier, which is tested with testing samples to evaluate the accuracy of the classifier. However, we have used some supervised classifiers in the experimental design. These include traditional classifiers such as K-NN, Naïve Bayes, SVM, and Decision Tree. The proposed methods such as ABC for image classification are in the category of supervised classifiers. This paper will deal with supervised classification, so unsupervised techniques will not be further explored. The rest of this paper is organized as follows. Section 2 presents a brief overview of image feature extraction. Section 3 describes the proposed image classification using modified PSO. Section 4 is dedicated to experimental setup and results. Conclusions are given in Section 5.

2. Image feature extraction

Development of classification of signals (or images, patterns) is significant in various fields of signal processing applications such as acoustic, speech, radar, communication, biomedical and multimedia systems (e.g. (Desai & Shazeer, 1991; Etemad & Chellappa, 1998; Leraned, 1992)). To construct effective classification systems, feature extraction of signals is an essential process. In application systems, the signals are not always expressed as ideal forms. The observed signals are distorted by the effect of noise and measurement, and some ambiguous signals are difficult to classify into a class by binary logic. Moreover, when the feature extraction process is very sensitive to distortions, the classification algorithm does not work well.

To perform successful classification of images, characteristics that describe the images are important, so these characteristics must be extracted by suitable algorithms and stored as features. Features define what characteristics of the images will be relevant to the database so that the database can be relied on to distinguish and classify images. These features allow access only to some characteristics of the images that are selected a priori. After image transformation, features can be derived from images. To classify images, this large class of features is seen as a vector of continuous measurements on the images S = {x1, …, xn}, where n is the size of the dimensional metric space.

Features that are derived from an image are generally represented in a vector space. The difference between two images can be measured by performing a difference function on the two images' vectors (x – y) on which a definition of distance can be based on (Gong, 1998).

Several types of point features, such as colour, are commonly used. These point features can be represented in terms of histograms. Histograms provide a "too-synthetic" description of the feature's values in the image, which is a problem in image classification systems. To solve this problem, detection of regions is employed. With this solution, histograms can provide a more accurate description. The use of point features and region features will be further discussed in the sub-sections.

Moreover, feature extraction is to extract the features which are a good representation of the content of the images. In the proposed method, the content of the images are represented using a colour histogram. A colour histogram denotes the joint probabilities of the intensities of the three colour channels (Smith & Chang, 2005).

Moreover, the most common colour space used in classification is RGB and YUV. However, the RGB colour space does not exhibit perceptual uniformity and the component values are correlated and depend on change in light intensity; prior work (Smith & Chang, 2005) suggested YUV colour space.

2.1 Colour space

Colour is one of the major features in an image. This is why many image classifications utilise colour feature extraction for classification purposes. Colour can be utilised as a feature by employing a colour space, which is then divided into *n* cells, and creating a histogram with *n* bins, where each bin contains the number of pixels in each cell (Gong, 1998). A system will compare the query image's histogram with each model image's histogram based on a similarity metric function (Gong, 1998). Images can be transformed into representations of different colour spaces. Images can represent different information in different colour spaces, thus, the use of different colour spaces can pose different effects on the image classification results. In the following section 2.2 and section 2.3, RGB and YUV colour spaces will be discussed. Then, section 2.4 will introduce the concept of DCT coefficients.

2.2 RGB colour space

RGB is the most popular colour space. Its acronym stands for Red-Green-Blue. RGB are considered the primary colours because a colour can be produced by adding these three predominant colours. For example, pure red and green light produce yellow; red and blue make magenta; blue and green combine to make cyan; and all three together, when mixed at full intensity, create white.

A common use of RGB is for the display of computer graphics on the monitor. Each pixel is represented by red, green and blue values. There are a total of 24 bits of information in a pixel: 8 bits from each colour. This means each colour has a range of 256 possible values. Therefore, RGB values can be specified using three integers between 0 and 255, where each integer represents red, green and blue intensities respectively.

2.3 YUV colour space

YUV, which is also known as YC_bC_r and YP_bP_r, is a colour space where the Y stands for the luminance component (the brightness), and U and V are chrominance (colour) components. The Y component is more important than the other two components to the human eye because the human eye can detect brightness better than colour. YUV signals can be created by the conversion of RBG values. The weighted values of R, G and B are added together to produce a single Y signal, representing the overall luminance of that pixel, whereas the U signal is created by subtracting the Y from the blue value of the RGB, and V by subtracting the Y from the red value. The YUV values can be calculated from RGB values as follows (Wikipedia):

$$Y = 0.299*Red + 0.587*Green + 0.114*Blue \qquad \text{Equation 1}$$

$U = -0.147*Red - 0.289*Green + 0.436*Blue$ Equation 2

$V = 0.615*Red - 0.515*Green - 0.100*Blue$ Equation 3

2.4 Discrete Cosine Transform (DCT)

Classification is widely used for image in Discrete Cosine Transform (DCT), subband, and wavelet transform coding systems. Chen and Smith (Chen & Smith, 1977) proposed a classification method for DCT coding. In this method, each DCT block is classified into one of the four equally populated classes based on its AC energy, and an adaptive quantiser is used in each class being coded. Woods and O'Neil (Woods, 1986) applied a classification method to subband coding. Tse (Tse, 1992) improved the classification performance by having different size classes. Joshi, Fischer and Bamberger (Joshi, 1994) investigated this type of classification technique. They optimized the classification method by maximising classification gain (Joshi, 1994), and applied this technique to subband coding. Jafarkhani, Farvardin and Lee (Jafarkhani, 1994) modified the Chen-Smith classification method for the discrete wavelet transform coding of images.

The Chen-Smith classification method (Chen & Smith, 1977) and its modified versions mentioned above belong to block classification; a whole block of DCT coefficients or data in each subband is classified into one of the predesigned classes according to block energy. Four classes are usually used. The simple formulation of block classification actually restrains its classification performance. Mohd Yusof and Fischer's (Mohd Yusof & Fischer, 2003) simulation results showed that the four-class block classification for DCT image coding can only achieve minimal improvement of the peak signal-to-noise ratio (PSNR) in their entropy-coded lattice vector quantisation systems. Entropy coding is often needed to exploit most of the advantages of the non-stationary nature of images in the variable-rate coding systems with block classification.

Moreover, DCT is a mathematical transformation which is in a class of mathematical operations (Nelson & Gailly, 1995). The famous Fast Fourier Transform is one of these mathematical operations (Nelson & Gailly, 1995). The basic operation of these transforms is to take a signal and transform it from one type of representation into another representation (Nelson & Gailly, 1995). From Figure Figure 3, it shows that the DCT is performed on an NxN square matrix of pixel values to obtain an NxN square matrix of coefficients. This DCT transformation is also reversible, which is shown in Figure 2 (Nelson & Gailly, 1995).

In addition to that, each image's colour feature is represented by a colour histogram in the database. Basically, there are five types of histograms tested in the proposed system. One is the RGB colour histogram and the other four are the histograms based on DCT coefficients. For each image, its colour feature is extracted in RGB colour space. This creates the RGB colour histogram for the corresponding image by pixel counting. For constructing DCT coefficients, RGB values are converted into YUV colour space using the conversion formula stated in section 0 of this paper. The image is further divided into 8x8 blocks and DCT is performed on each block to obtain the DC and AC components. Then the four different combinations of DCT coefficients, which are used to create four histograms, are constructed using the DC and AC components. There are a total of four different types of DCT coefficients as image features that have been extracted for proposed modified PSO classifier in the experiments used in section . The four DCT coefficients that are used in the system are described below:

S1 -> [DC]

S2 -> [AC_{01}; AC_{10}; AC_{11}]

S3 -> [DC; AC_{01}; AC_{10}; AC_{11}]

S4 -> [DC; AC_{01}; AC_{02}; AC_{10}; AC_{11}; AC_{12}; AC_{20}; AC_{21}; AC_{22}]

Furthermore, there are 800 images in JPEG format. These images are then converted into RAW format. Each of the five histograms (RGB and DCT coefficients - S1, S2, S3 and S4) consists of three colour bins which are:

1. Primary bins (8 colour bins)
2. Secondary bins (8 colour bins)
3. Third bins (8 colour bins)

Each of the colour bins consists of 8 feature vectors. Therefore, the total number of feature vectors is 24 feature vectors which make 24 dimensions. Each vector represents the pixel values in each image. These vector values have been resulted from the transformation using the DCT algorithm and RGB colour space.

3. The proposed modified PSO for Image Classification

In this section, we will introduce the proposed image classifier using modified PSO which was developed by our research team (Yeh, Chang, & Chung, 2009). It belongs to the category of Swarm Intelligence methods; it is also an evolutionary computation method inspired by the behaviour of natural swarms such as bird flocks and fish schools. In PSO, a solution is encoded as a finite-length string called a particle or individual. All of the particles have fitness values which are evaluated by the fitness function to be optimized. Each particle flies in the search space with a velocity that can be dynamically adjusted according to its own previous flying experience and its companions' flying experience. The *i*th particle's position is represented as $X_i = (x_{i1}, x_{i2} ... x_{in})$. The best previous position (the position with the best fitness value) of the *i*th particle called *pbest* is represented as $P_i = (p_{i1}, p_{i2} ... p_{in})$. The global best position attained by any particle so far (called *gbest*) is also recorded and stored in a particle denoted by $G_i = (g_{i1}, g_{i2} ... g_{in})$. The coordinates $x_{(i,j)}$ of these particles have a rate of change (velocity) $v_{(i,j)}$, $d = \{1, 2, . . . , n\}$. After finding the two best values, the particle updates its velocity and position with the following equation:

$$v_{ij}^{t} = w \cdot v_{ij}^{t-1} + c_1 \cdot \rho_1 \cdot \left(p_{ij}^{t-1} - x_{ij}^{t-1}\right) + c_2 \cdot \rho_2 \cdot \left(g_{ij}^{t-1} - x_{ij}^{t-1}\right) \quad \text{Equation 4}$$

$$x_{ij}^{t} = x_{ij}^{t-1} + v_{ij}^{t-1} \quad \text{Equation 5}$$

In Equation 4, the first part is the previous velocity of the particle; in Equation 5 is the "cognition" part, representing the private thinking of the particle itself, where c_1 is the individual factor; the third is the "social" part, representing the collaboration among the particles, where c_2 is the societal factor (Kennedy & Eberhart, 1995). The acceleration constant, the individual factor (c_1) and social factor (c_2) represent the weighting of the stochastic acceleration terms that pull each particle toward *pbest* and *gbest* positions. Low values allow particles to roam far from target regions before being tugged back, while high values result in abrupt movement toward, or past, target regions. The value of (c_1 + c_2) is usually limited to 4 (Kennedy & Eberhart, 1995). Particles' velocities on each dimension are limited to a maximum velocity, V_{max}. It determines the size of the steps through the solution space each particle is allowed to take. If V_{max} is too small, particles may not explore sufficiently beyond locally good regions. They could become trapped in local optima. On the other hand, if V_{max} is too high, particles might fly past good solutions. Equation 4 is used to calculate the particle's new velocity according to its previous velocity and the distances of its current position from its own best experience (position) and the group's best experience. Then the particle flies toward a new position according to Equation 5. The performance of each particle is measured according to a pre-defined fitness function (Kennedy & Eberhart, 1995).

To apply PSO, several parameters including the number of population (*m*), cognition learning factor (c_1), social learning factor (c_2), inertia weight (*w*), and the number of iterations or CPU time should be determined properly. We conducted the preliminary experiments, and the complete computational procedure of the PSO algorithm can be summarized as follows.

Input:

m: the swarm size; c_1, c_2: positive acceleration constants

w: inertia weight

MaxV: maximum velocity of particles

MaxGen: maximum generation

MaxFit: maximum fitness value.

Output:

gbest: Global best position.

Compared with other EAs, such as the GA, PSO converge faster. The convergence efficiency of PSO is influenced little by the number of dimensions. However, a huge number of dimensions may cause some other problems, such as local optimization. It means particles prematurely converge at local optimal positions and stop searching for the real global optimum. Therefore, there have been many researchers working on this through parameter adjusting, diversity increases and algorithm variation in order to improve PSO's globally optimal ability.

3.1 Individual Updates

The underlying principle of the traditional PSO is that the next position of each particle is a compromise of its current position, the best position in its history so far, and the best position among all existing particles. Equation

4 and Equation 5 are very easy and efficient ways to decide next positions for the problems with continuous variables, but not trivial and well-defined for the problems with discrete variables and sequencing problems. To overcome the drawback of PSO for discrete variables, a novel method to implement the PSO procedure has been proposed based on the following equation after C_w, C_p, and C_g are given:

$$x_{ij}^{t} = \begin{cases} x_{ij}^{t-1} & if\ R_{ij}^{t} \in [0, C_w) \\ p_{ij}^{t-1} & if\ R_{ij}^{t} \in [C_w, C_p) \\ g_{\mathrm{i}} & if\ R_{ij}^{t} \in [C_p, C_g) \\ x & if\ R_{ij}^{t} \in [C_g, 1) \end{cases} \qquad \text{Equation 6}$$

The proposed individual update process is described below:

1) Initialise: Initialise parameters and population with random position and velocities.

2) Evaluation: Evaluate the fitness value (the desired objective function) for each particle.

3) Find the *gbest*: If the fitness value of particle *i* is better than its best fitness value (*pbest*) in history, then set current fitness value as the new *pbest* to particle *i*.

4) Find the *gbest*: If any *pbest* is updated and it is better than the current g*best*, then set g*best* to the current value.

5) Update position: For every dimension of a particle, generate a random variable called *R* that is between 0 and 1. If $0 \le R < C_w$ is true then the original individual will be kept, else if $C_w \le R < C_p$ is true then the original dimension will be replaced by *pbest*'s corresponding one, else if $C_p \le R < C_g$ is true then the original individual will be replaced by g*best*'s corresponding one, else if $C_g \le R < 1$ is true then the original dimension will be replaced by a new value which is generated randomly.

6) Stopping criterion: If the number of iterations or CPU time is met, then stop; otherwise go back to Step 2.

3.2 Comparison

In the traditional PSO, each particle needs to use more than two equations, generate two random numbers, four multiplications, and three summations in order to move to its next position. Thus, the time complexity is O (6nm) for the traditional PSO. However, the proposed modified PSO does not need to use the velocity, it only uses one random, two multiplications, and one comparison after C_w, C_p and C_g are given. The main reason why velocity is not used is because it reduces the processing time and decrease the complexity of the algorithm. Therefore, the proposed modified PSO is more efficient than the other PSOs.

Figure Figure 8 is the flow diagram to explain the proposed process of individual update.

3.3 Encoding

A classification rule contains two parts: the antecedent and the consequent. The former is a series of logical tests, and the latter gives the class while an instance is covered by this rule. These rules take the following format:

where feature is the attribute of a dataset, lower bound and upper bound are the feature's lowest and highest value respectively. Each clause (dimension) is composed of a feature, its lower bound and upper bound. The position representation of each individual (particle) contains N clauses (dimensions) except the last cell, ClassX, which is the predictive class of the rule.

3.4 Fitness Function

The fitness function that has been used for the proposed modified PSO algorithm for image classification is shown in Equation 7.

$$\text{Cover percentage} = \frac{TP}{N} \qquad \text{Equation 7}$$

In Equation 7, *N* is the total number of records which belong to the class predicted by the rules. The prediction strategy balanced the effect of fitness value and cover percentage for the final predicted class. We need to choose the value of α and β carefully since they will affect the classification accuracy.

3.5 Rule Discovery

The goal of classification rule mining is to discover a set of rules with high quality (accuracy). To achieve this, appropriate lower bound and upper bound for each attribute (feature) are searched for. In the initial stage, for each attribute we set its position of upper bound between a randomly chosen seed example's value and that value added to the range of that feature. Similarly, the value of lower bound is initialised at a position between the seed example's value, and that value minus the range of that feature. The procedure is defined as:

Lower bound = $k_1 * (S - R)$ Equation 8

Upper bound = $k_2 * (S + R)$ Equation 9

where *S* is the corresponding attribute value of a randomly chosen instance; *R* is the range of corresponding attribute value for all training instances; k_1 and k_2 are two random numbers between 0 and 1.

This seeding strategy may generate the upper or lower bound outside the range from training instances. The purpose of this is to enlarge the search space because the original range from training instances may not benefit the validation instances. For those attributes only containing nominal variables, we set its lower bound equal to the upper bound.

3.6 Rule Pruning

After a rule has been generated, it is put into a rule pruning procedure. The main goal of rule pruning is to remove irrelevant clauses that might have been unnecessarily included in the rule. Moreover, rule pruning can increase the predictive power of the rule, helping to improve the simplicity of the rule. The process of rule pruning is as follows:

1) Evaluate a rule's quality.

2) Tentatively remove terms from each rule and see if each term can be removed without the loss of rule quality. If yes, remove it. Then move onto the next term and eventually the next rule. This process is repeated until no term can be removed.

3.7 Prediction Strategy

After we generate a rule set, a series of testing instances are used to measure its classification accuracy. For each instance, it will go through every element in a rule set and get a prediction value for the corresponding class when it is covered by a rule. The prediction function is defined as follows:

Prediction value = α * *rule quality* + β **rule cover percentage* Equation 10

where α and β are two parameters corresponding to the importance of rule quality and rule cover percentage, $\alpha \in [0, 1]$ and $\beta = 1-\alpha$. The prediction value for each class is cumulative and the result is the class with the highest prediction value..

4. Experimental Design and Results

To thoroughly investigate the performance of the proposed modified PSO algorithm, we have conducted experiments with it on a number of datasets. The datasets are taken from extracting image datasets from two methods which are RGB Histogram and DCT coefficients. Basically, there are 800 images in the datasets which are in JPEG format. We have categorized these 800 images into five categories: 1) Building 2) Airplane 3) Bottle 4) Camel 5) Flower. Table2 shows the sample images for each category. Each of the images has been resized 160x160 with 24 dimensions for RGB and DCT. In Table 1, the selected data sets are summarised in terms of the number of instances, and the number of the classes of the dataset. These datasets have been widely used in other comparative studies.

4.1 Pre-processing Datasets

The pre-processing of the experiment datasets is done using different types of colour space. In this experiment, the image features are in two methods. Moreover, there are, in total, two different extracted features have been tested in the proposed modified PSO for classification accuracy. The two feature extraction algorithms used in the experiments are RGB Histogram and DCT coefficients

For image extraction for the RGB Histogram and DCT coefficient, each image's colour feature is represented by a colour histogram in the database. Basically, there are five types of histograms tested in the proposed system.

One is the RGB colour histogram and the other four are the histograms based on DCT coefficients. For each image, its colour feature is extracted in RGB colour space. This creates the RGB colour histogram for the corresponding image by pixel counting. For constructing DCT coefficients, RGB values from JPEG images are converted into YUV colour space using the conversion formula stated in Equation 1, Equation 2 and Equation 3. The image is further divided into 8x8 blocks, and DCT is performed on each block to obtain the DC and AC components as described in Figure 3. Then the four combinations of DCT coefficients, which are used to create four histograms, are constructed using the DC and AC components.

4.1.1 Experiment 1: RGB and DCT Coefficients Dimensionality Testing

4.1.1.1 Experimental Setup

The purpose of this test is to obtain the classification accuracy results based on different dimensions of the 800 images' histograms. These histograms are based on RGB values and DCT coefficients. The dimensions tested are 15, 18, 21, and 24. This test is performed based on five traditional classification algorithms (K-NN, Naïve Bayes, SVM, and C4.5) using two DM software which are Weka (Mark, et al., 2009) and Orange (J. & B., 2005). As a result, the test would determine the effect of different dimension sizes on the accuracy of the results and their suitability for classification, and find the optimum dimension for higher classification accuracy.

4.1.1.2 Results and Discussions

Based on the results in Table 3 to Table 7, it is found that image datasets that are extracted using 24 dimensions for RGB and DCT coefficients (S1, S2, S3, and S4) have the highest classification accuracy. However, images that are extracted with 15 dimensions have the lowest accuracy for RGB Histogram and DCT coefficients (S1, S2, S3, and S4). Therefore, it is concluded that the optimal dimensions for gaining the highest classification accuracy is 24 dimensions. If we use less than 24 dimensions, the classification accuracy will dramatically decrease.

Besides that, Table 8 shows that among the DCT coefficients with 24 dimensions that have the highest classification accuracy is DCT coefficient S3, with an average of 94.40% among the five classifiers that have been tested. This is mainly because DCT coefficient S3 has the optimal coefficients compared to other DCT coefficients. Therefore, it can be concluded that DCT coefficient S3 is the optimal coefficient and can be used as a comparison to the proposed modified PSO classifier in the next phase.

4.1.2 Experiment 2: Testing of the Proposed modified PSO for the Image Data Classification

4.1.2.1 Experimental Setup

The experiment was carried out to compare predictive accuracy of discovered rule lists by well-known 10-fold cross-validation procedures. Each data set is divided into 10 partitions, each method is run 10 times, using a different partition as the test set, and the other nine partitions as the training set each time. The two parameters α and β in Equation 10 are important to the final validation accuracy. Slight adjustment could change the results significantly. In our experiment, we set $\alpha = \beta = 0.5$. Moreover, we have used DCT coefficient S3 in this experiment because, according to Table 8, it has the highest classification accuracy compared to other coefficients.

After the cross-validation of two data sets, we get the average validation accuracy of each data set. The result of these two image features using the proposed modified PSO is shown in Table 9. We compare these results with the other five algorithms in Table 10. We have used Weka's (Mark, et al., 2009) software to implement the C4.5 classifier on the two different image features. Furthermore, we have also used Orange's (J. & B., 2005) software to implement a traditional classifier such as K-NN, Naïve Bayes, and SVM. The results are shown in Table 10.

4.1.3 Results and Discussions

According to Table 9, RGB datasets has the highest average validation accuracy with 98.28%. The lowest average accuracy would be the DCT datasets that would give 91.44% validation classification accuracy. As for training accuracy, DCT datasets has the highest average training accuracy with 96.35%, whereas RGB has the lowest average training accuracy with 94.78%. Furthermore, we have compared proposed modified PSO against these algorithms as they are considered industry standards. The results of the five algorithms are shown in Table 10. According to Table 10, the highest average classification accuracy would be the proposed modified PSO with average classification accuracy 93.23%. Nevertheless, proposed modified PSO classifier also has the highest accuracy for RGB Histogram and DCT Coefficient (S3), with 98.28% and 91.44% respectively. The main reason why proposed modified PSO performed well for those datasets is because it has the ability to produce highest optimal solutions and thus reduce misclassified problems. In conclusion, the comparison clearly states the competitiveness of the proposed modified PSO with other algorithms.

5. Conclusion

In this chapter, we have introduced the proposed modified PSO algorithm that provided a mechanism of classification rule mining based on PSO and effective feature extraction processes on image datasets. Moreover, we have conducted experiments on the process of image feature extraction using two main methods which are RGB Histogram and DCT coefficients. After image feature extraction process, both datasets have been produced. These two datasets have been used as testing datasets in the next experiments which involve testing the accuracy of the proposed modified PSO algorithm. In testing the proposed modified PSO for data classification, it is found that the rule discovering procedure and the accuracy rate of datasets is higher compared to the other five traditional algorithms.

References

Chen, W., & Smith, C. (1977). Adaptive coding of monochrome and color images. *IEEE Transactions on Communications, 25*(11), 1285-1288. doi:10.1109/TCOM.1977.1093763, http://dx.doi.org/10.1109/TCOM.1977.1093763

Desai, M., & Shazeer, D. J. (1991). *Acoustic transient analysis using wavelet decomposition*. Paper presented at the Neural Networks for Ocean Engineering, 1991., IEEE Conference on. Retrieved from 10.1109/ICNN.1991.163324

Etemad, K., & Chellappa, R. (1998). Separability-based multiscale basis selection and feature extraction for signal and image classification. *Image Processing, IEEE Transactions on, 7*(10), 1453-1465.

Gong, Y. (1998). *Intelligent Image Databases Towards Advanced Image Retrieval*. Massachusetts: Kluwar Academic Publishers.

J., D., & B., Z. (2005). Orange: From Experimental Machine Learning to Interactive Data Mining. from www.ailab.si/orange

Jafarkhani, J., Farvardin, N., Lee, C. (1994). *Adaptive image coding based on the discrete wavelet transform*. Paper presented at the IEEE Conf on Image Processing.

Joshi, R., Fischer, T., Bamberger, R. (1994). *Optimum classification in subband coding of images*. Paper presented at the IEEE Conf. on Image Processing.

Kennedy, J., & Eberhart, R. (1995). *Particle swarm optimization.* Paper presented at the Neural Networks, 1995. Proceedings., IEEE International Conference on.

Leraned, R. E., Karl, W.C., Willsky, A.S. (1992). *Wavelet Packet Based Transient Signal Classification*. Paper presented at the IEEE TSTFA.

Lillesand, T. W., & Kiefer, R. W. (1994). *Remote Sensing and Image Interpretation* (Third ed.). New York: Wiley.

Mark, H., Eibe, F., Geoffrey, H., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations, 11*(1).

Mohd Yusof, Z., & Fischer, T. (2003). *An entropy-coded lattice vector quantizer for transform and subband image coding*. Paper presented at the IEEE Conf. on Image Processing.

Nelson, M., & Gailly, J. L. (1995). *The Data Compression Book*. New York: M&T Books.

Smith, J. R., & Chang, S. F. (2005). *Tools and Technique for color and image retrieval.* Paper presented at the IS &T/SPIE proceedings: Storage & Retrieval for Image and Video Database IV.

Tse, Y. T. (1992). *Video coding using global/local motion compensation, classified subband coding, uniform threshold quantization and arithmetic coding.* Unpublished Ph.D thesis, University of California, Los Angeles.

Wikipedia. YCbCr Color Space. from http://en.wikipedia.org/wiki/YCbCr

Witten, I. H., & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*: Elsvier.

Woods, J., O'Neil, S. (1986). Subband coding of images. *IEEE Transactions on Acoustics, Speech, and Signal processing, 34*(10), 1278-1288. doi:10.1109/TASSP.1986.1164962, http://dx.doi.org/10.1109/TASSP.1986.1164962

Yeh, W.-C., Chang, W.-W., & Chung, Y. Y. (2009). A new hybrid approach for mining breast cancer pattern using discrete particle swarm optimization and statistical method. *Expert Systems with Applications, 36*(4), 8204-8211. doi:10.1016/j.eswa.2008.10.004, http://dx.doi.org/10.1016/j.eswa.2008.10.004

Table 1. Datasets used in the data mining experiment

Dataset	Features	Instance	Class
RGB	24	800	5
DCT	24	800	5

Table 2. Five classes of images

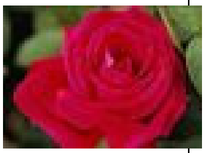
Class Number	1	2	3	4	5
Class Name	Building	Airplane	Bottle	Camel	Flower
Sample Image					
Number of Images in Class	60	50	240	350	100

Table 3. Average K-NN classification accuracy (%) by image dimension sizes

K-NN Classifier						
Dimension Size	RGB	S1	S2	S3	S4	Average Accuracy by Dimension Size
24 dimensions	83	94	90	95	92	90.8
21 dimensions	53	45	46	50	48	48.4
18 dimensions	43	41	36	39	37	39.2
15 dimensions	34	17	20	24	21	23.2
Average accuracy RGB and DCT coefficients	53.25	49.25	48	52	49.5	

Table 4. Average Naïve Bayes classification accuracy (%) by image dimension sizes

Naïve Bayes Classifier						
Dimension Size	RGB	S1	S2	S3	S4	Average Accuracy by Dimension Size
24 dimensions	81	92	90	93	91	89.4
21 dimensions	51	44	46	49	47	47.4
18 dimensions	42	42	32	38	35	37.8
15 dimensions	36	19	21	25	20	24.2
Average accuracy RGB and DCT coefficients	52.5	49.25	47.25	51.25	48.25	

Table 5. Average SVM classification accuracy (%) by image dimension sizes

SVM Classifier						
Dimension Size	RGB	S1	S2	S3	S4	Average Accuracy by Dimension Size
24 dimensions	83	96	89	94	91	90.6
21 dimensions	53	44	46	50	48	48.2
18 dimensions	43	41	36	37	36	38.6
15 dimensions	33	17	22	24	21	23.4
Average accuracy RGB and DCT coefficients	53	49.5	48.25	51.25	49	

Table 6. Average C4.5 classification accuracy (%) by image dimension sizes

C4.5 Classifier						
Dimension Size	RGB	S1	S2	S3	S4	Average Accuracy by Dimension Size
24 dimensions	81	95	90	93	92	90.2
21 dimensions	53	45	43	50	48	47.8
18 dimensions	41	42	36	38	37	38.8
15 dimensions	36	17	20	22	21	23.2
Average accuracy RGB and DCT coefficients	52.75	49.75	47.25	50.75	49.5	

Table 7. Average modified PSO classification accuracy (%) by image dimension sizes

Proposed Modified PSO Classifier						
Dimension Size	RGB	S1	S2	S3	S4	Average Accuracy by Dimension Size
24 dimensions	87	94	90	97	95	92.60
21 dimensions	56	45	46	50	48	49.00
18 dimensions	42	41	38	39	37	39.40
15 dimensions	37	17	20	22	21	23.40
Average accuracy RGB and DCT coefficients	55.50	49.25	48.50	52.00	50.25	

Table 8. Average classification accuracy for each DCT coefficient with 24 dimensions

Classifier	S1	S2	S3	S4
K-NN	94	90	95	92
Naïve Bayes	92	90	93	91
SVM	96	89	94	91
C4.5	95	90	93	92
Modified PSO	94	90	97	95
Average DCT coefficients	94.20	89.80	94.40	92.20

Table 9. Ten-fold cross-validation for six datasets using modified PSO classifier

	RGB		**DCT (S3)**	
Number of Fold	**Training accuracy**	**Validation accuracy**	**Training accuracy**	**Validation accuracy**
1	**95.05**	**98.28**	**97.92**	**95.33**
2	**94.21**	**98.71**	**97.44**	**92.57**
3	**95.05**	**98.71**	**96.96**	**91.71**
4	**94.74**	**97.85**	**95.96**	**94.43**
5	**95.67**	**97.85**	**99.92**	**85.31**
6	**95.05**	**98.71**	**98.94**	**89.00**
7	**94.12**	**97.85**	**98.68**	**90.33**
8	**95.05**	**98.71**	**80.21**	**79.50**
9	**94.43**	**97.42**	**98.34**	**97.75**
10	**94.42**	**98.71**	**99.10**	**98.43**
Average	**94.78**	**98.28**	**96.35**	**91.44**

Table 10. Comparison results of data classification experiment

Data Set	Proposed Modified PSO (%)	C4.5 (%)	K-NN(%)	SVM(%)	Naïve Bayes(%)
RGB Histogram	98.28	93.70	96.63	96.06	97.07
DCT Coefficient (S3)	91.44	65.43	69.65	69.16	70.22
Average Classification Accuracy	93.23	85.63	87.59	84.76	86.04

$$\text{DCT(i,j)} = \frac{2}{N}\ \text{C(i) C(j)} \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} \text{pixel(x,y) COS}\left[\frac{(2x+1)i\pi}{2N}\right] \text{COS}\left[\frac{(2y+1)i\pi}{2N}\right]$$

$$\text{C(x)} = \frac{1}{\sqrt{2}}\ \text{if x is 0, else 1 if x > 0}$$

Figure 1. The Discrete Cosine Transform

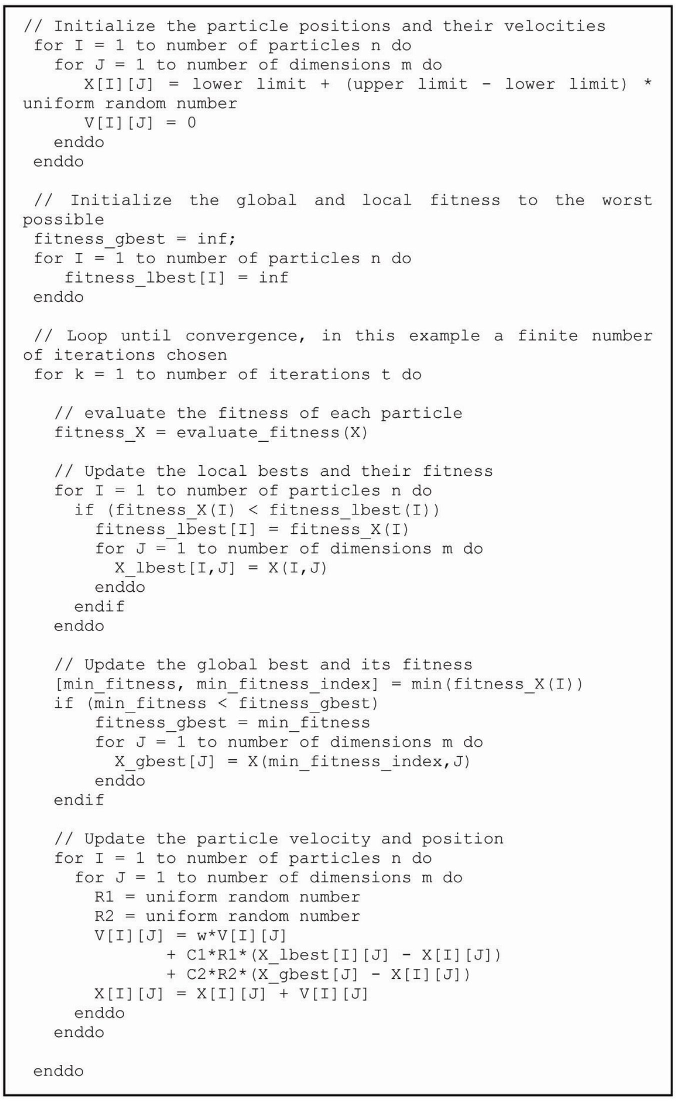
$$\text{Pixel(x,y)} = \frac{2}{N}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \text{C(i) C(j) DCT(i,j) COS}\left[\frac{(2x+1)i\pi}{2N}\right] \text{COS}\left[\frac{(2y+1)i\pi}{2N}\right]$$

$$\text{C(x)} = \frac{1}{\sqrt{2}}\ \text{if x is 0, else 1 if x > 0}$$

Figure 2. The Inverse Discrete Cosine Transform

DC	AC_{10}	AC_{20}					
AC_{01}	AC_{11}	AC_{21}					
AC_{02}	AC_{12}	AC_{22}					

Figure 3. DCT on a JPEG image showing more important components



```
// Initialize the particle positions and their velocities
 for I = 1 to number of particles n do
   for J = 1 to number of dimensions m do
      X[I][J] = lower limit + (upper limit - lower limit) *
uniform random number
      V[I][J] = 0
   enddo
 enddo

 // Initialize the global and local fitness to the worst
possible
 fitness_gbest = inf;
 for I = 1 to number of particles n do
    fitness_lbest[I] = inf
 enddo

 // Loop until convergence, in this example a finite number
of iterations chosen
 for k = 1 to number of iterations t do

   // evaluate the fitness of each particle
   fitness_X = evaluate_fitness(X)

   // Update the local bests and their fitness
   for I = 1 to number of particles n do
     if (fitness_X(I) < fitness_lbest(I))
       fitness_lbest[I] = fitness_X(I)
       for J = 1 to number of dimensions m do
         X_lbest[I,J] = X(I,J)
       enddo
     endif
   enddo

   // Update the global best and its fitness
   [min_fitness, min_fitness_index] = min(fitness_X(I))
   if (min_fitness < fitness_gbest)
       fitness_gbest = min_fitness
       for J = 1 to number of dimensions m do
         X_gbest[J] = X(min_fitness_index,J)
       enddo
   endif

   // Update the particle velocity and position
   for I = 1 to number of particles n do
     for J = 1 to number of dimensions m do
       R1 = uniform random number
       R2 = uniform random number
       V[I][J] = w*V[I][J]
              + C1*R1*(X_lbest[I][J] - X[I][J])
              + C2*R2*(X_gbest[J] - X[I][J])
       X[I][J] = X[I][J] + V[I][J]
     enddo
   enddo

 enddo
```

Figure 4. Standard PSO algorithm

Feature0	Lower bound	Upper bound	……	FeatureN	Lower bound	Upper bound	Class X

Figure 6. The form of encoding

```
FOR EACH class C
     Add all training instances into training dataset
     WHILE (all instances belong to class C have not been covered)
            Run the MODIFIED PSO to generate new rule
                       Add the new rule into rule set
            Remove instances covered by the new Rule
    ENDWHILE
END FOR
Prune Rule set removing unnecessary terms and/or rules
Order rules in Rule set according to rule's quality and cover percentage
```

Figure 7. Rule discovery procedure

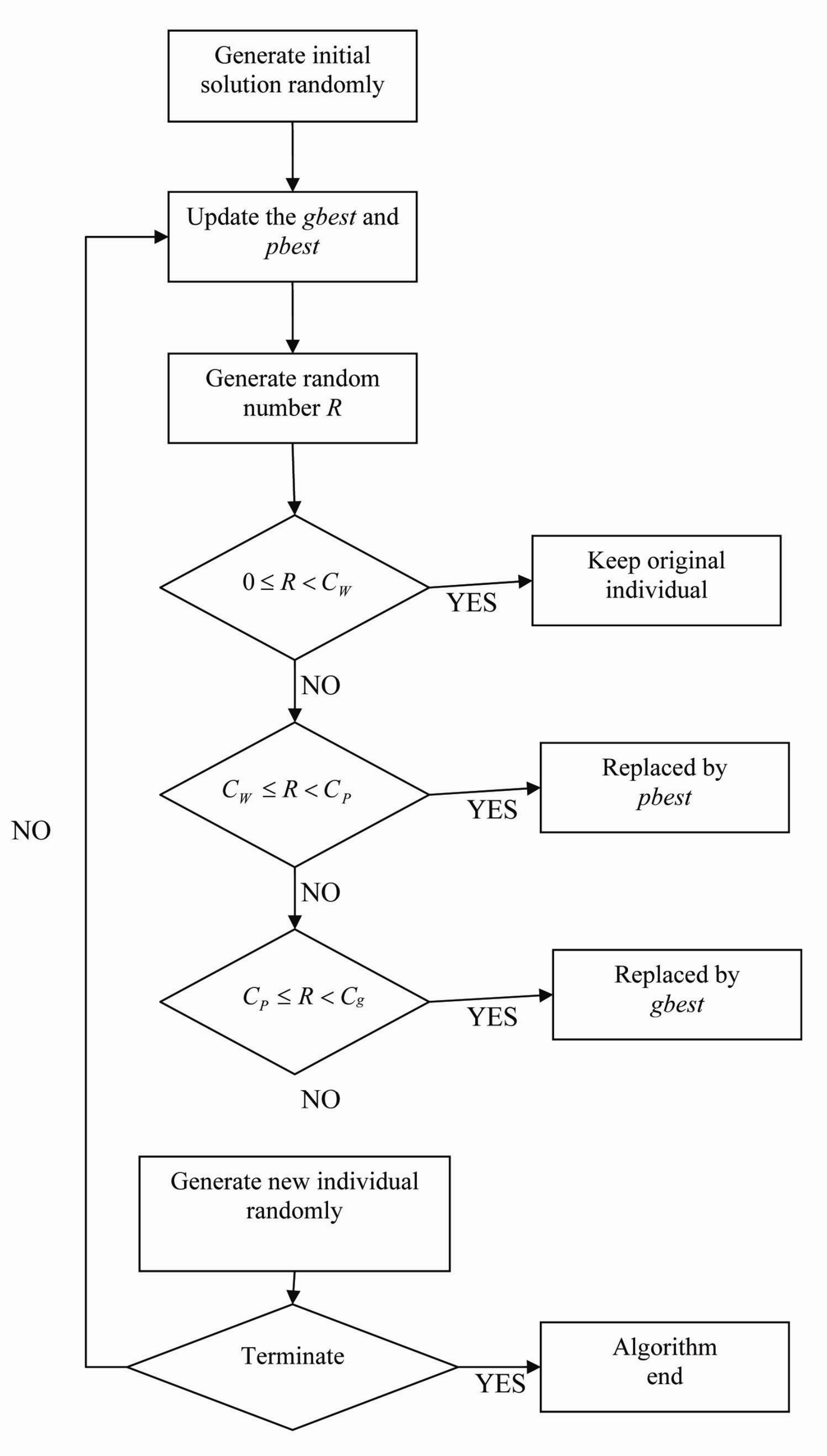


Figure 8. The flowchart of the individual update