Mathematical and Computer Modelling 54 (2011) 1030–1036

Contents lists available at ScienceDirect

Mathematical and Computer Modelling

journal homepage: www.elsevier.com/locate/mcm

Textural features in flower classification

D.S. Guru [a], Y.H. Sharath Kumar [a], S. Manjunath [a,b,*]

[a] Department of Studies in Computer Science, University of Mysore, Manasagangotri, Mysore 570 006, India
[b] International School of Information Management, University of Mysore, Manasagangotri, Mysore 570 006, India

ARTICLE INFO

Article history:
Received 14 August 2010
Accepted 4 November 2010

Keywords:
Color texture moments
Gray level co-occurrence matrix
Gabor responses
Flower classification
Probabilistic neural network

ABSTRACT

In this work, we investigate the effect of texture features for the classification of flower images. A flower image is segmented by eliminating the background using a threshold-based method. The texture features, namely the color texture moments, gray-level co-occurrence matrix, and Gabor responses, are extracted, and combinations of these three are considered in the classification of flowers. In this work, a probabilistic neural network is used as a classifier. To corroborate the efficacy of the proposed method, an experiment was conducted on our own data set of 35 classes of flowers, each with 50 samples. The data set has different flower species with similar appearance (small inter-class variations) across different classes and varying appearance (large intra-class variations) within a class. Also, the images of flowers are of different pose, with cluttered background under various lighting conditions and climatic conditions. The experiment was conducted for various sizes of the datasets, to study the effect of classification accuracy, and the results show that the combination of multiple features vastly improves the performance, from 35% for the best single feature to 79% for the combination of all features. A qualitative comparative analysis of the proposed method with other well-known existing state of the art flower classification methods is also given in this paper to highlight the superiority of the proposed method.

1. Introduction

Developing a system for classification of flowers is a difficult task because of considerable similarities among different classes and also due to large intra-class variations. In a real environment, images of flowers are often taken in natural outdoor scenes, where the lighting condition varies with the weather and time. In addition, flowers are often more or less transparent, and specula highlights can make the flower appear light or even white, causing illumination problems. Also, there is lot more variation in viewpoint, occlusions, and scale of flower images. All these problems lead to a confusion across classes and make the task of flower classification more challenging. In addition, the background also makes the problem difficult, as a flower has to be segmented automatically.

Applications of classification of flowers can be found useful in floriculture, flower searching for patent analysis, etc. Floriculture has become one of the important commercial trades in agriculture owing to a steady increase in the demand for flowers. The floriculture industry comprises flower trade, nursery and potted plants, seed and bulb production, micro propagation, and extraction of essential oil from flowers. In such cases, automation of flower classification is essential. Further, flower recognition is used for searching patent flower images to know whether the flower image for which a patent has been requested is already present in the patent image database or not [1]. Since these activities are done manually and are very labor intensive, automation of the classification of flower images is a necessary task.

* Corresponding author at: Department of Studies in Computer Science, University of Mysore, Manasagangotri, Mysore 570 006, India. Tel.: +91 9035228978.
E-mail addresses: dsg@compsci.uni-mysore.ac.in (D.S. Guru), sharath@yahoo.co.in (Y.H. Sharath Kumar), manju_uom@yahoo.co.in (S. Manjunath).

0895-7177/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mcm.2010.11.032

We can find a couple of works carried out in this direction. Nilsback and Zisserman [2] designed a flower classification system by extracting visual vocabularies which represent the color, shape, and texture features of flower images. In order to segment a flower from the background, the RGB color distribution is determined by labeling pixels as foreground and background on a set of training samples, and subsequently the flower is automatically segmented using the concept of interactive graph cuts [3]. In order to extract the color vocabulary, each flower image is mapped onto HSV (hue, saturation, and value) color space, and the HSV values of each pixel of the training images are clustered and treated as the color vocabulary. Shift-invariant feature transform (SIFT) descriptors are used to represent the shape features and the responses of the MR8 filter bank in different orientations are used as texture features. Also, the authors use the combination of all the three visual vocabularies with different weights in order to study the effect of different features. Nilsback and Zisserman [2] considered a dataset of 17 species, each containing 80 images, and achieved an accuracy of 71.76% for a combination of all three features. In order to study the effect of classification accuracy on a large data set, Nilsback and Zisserman in their work [4] considered a dataset of 103 classes, each containing 40 to 250 samples. The low-level features such as color, histogram of gradient orientations, and SIFT features are used. They have achieved an accuracy of 72.8% with an SVM classifier using multiple kernels. Nilsback and Zisserman [5] proposed a two-step model to segment the flowers in color images, one to separate the foreground from background and the other to extract the petal structure of the flower. This segmentation algorithm is tolerant to changes in viewpoint and petal deformation, and the method is applicable in general for any flower class.

Das et al. [1] proposed an indexing method to index flower patent images using domain knowledge. The flower was segmented using an iterative segmentation algorithm with domain knowledge driven feedback. In their work, the image color is mapped to names using the ISCC-NBS color system and X Window system. Each flower image is discretized in HSV color space, and each point on the discretized HSV space is mapped to a color name in the ISCC-NBS and X Window systems in order to index the flowers. Yoshioka et al. [6] performed a quantitative evaluation of petal colors using principal component analysis. They considered the first five principal components (PCs) of a maximum square on the petals. The quantitative evaluation indicates that the different PCs correspond to different color features of petals such as color depth, difference in color depth in upper and lower parts of the image, etc.

Varma and Ray [7] proposed a method for learning the trade-off between invariance and discriminative power for a given classification task. They learn the optimal, domain-specific kernel as a combination of base kernels corresponding to base features which achieve different levels of trade-off, such as rotation invariance, scale invariance, affine invariance, etc. The classification is carried out on the basis of vocabularies of visual words of shape, color, and texture descriptors. The background in each image is removed using graph cuts. Shape distances between two images are calculated as the statistic between the normalized frequency histograms of densely sampled, vector quantized SIFT descriptors of the two images. Similarly, color distances are computed over vocabularies of HSV descriptors and texture over MR8 filter responses. An entire set of weights is learnt, spanning the full range from shape to color.

In their work, Saitoh et al. [8] describe an automatic method for recognizing a blooming flower based on a photograph taken with a digital camera in a natural scene. They have also proposed a method for extracting the flower regions. It is based on "Intelligent Scissors" [9], which find the path between two points that minimizes a cost function dependent on image gradients. The method works under the assumption that the flower is in focus and in the centre of the photograph and that the background is out of focus. Under this assumption, the cost between any two points on the flower is smaller than the cost between a point in the background and a point in the foreground. The midpoint of the image is used as the starting point to identify the flower region. This method requires no prior color information. Saitoh et al. [10] designed a flower classification system which extracts features from both flowers and leaves, and used a piecewise linear discrimant analysis for recognition on a dataset of 34 species each containing 20 sets of wild flowers.

Nilsback and Zisserman [2] noted that color and shape are the major features in flower classification. This is true only when the flower classes considered have little intra-class variation. However, if there is a large variation within a class, for example where species of the same type have different colors, then color may not be the best suitable feature. Hence, in this work, we investigate the suitability of texture features in designing a system for flower classification. The flower is segmented using a threshold-based method, and texture features, namely the color texture moments (CTMs), gray level co-occurrence matrix (GLCM), and Gabor responses, are extracted from the segmented image and used for classification. In considering the color texture moments, we extract moments from different color spaces of the flower images. In the gray level co-occurrence matrix, features such as contrast, energy, entropy, correlation, and homogeneity are taken into account. In the Gabor analysis, we have extracted the first three moments of each of the Gabor responses obtained for different scales and orientations. These features are used for training and classification using a probabilistic neural network.

The organization of the paper is as follows. In Section 2, the proposed method is explained with the support of a block diagram along with a brief introduction to CTMs, the GLCM, and Gabor texture analysis. The experimental results under varying database sizes and a qualitative comparative analysis on the state of the art techniques are discussed in Section 3. The paper is concluded in Section 4.

2. Proposed method

The proposed method has training and classification phases. In the training phase, from a given set of training images the texture features (CTMs/GLCM/Gabor/Combination) are extracted and used to train the system using a probabilistic neural network. In the classification phase, given a test image, the flower is segmented and the above-mentioned texture features

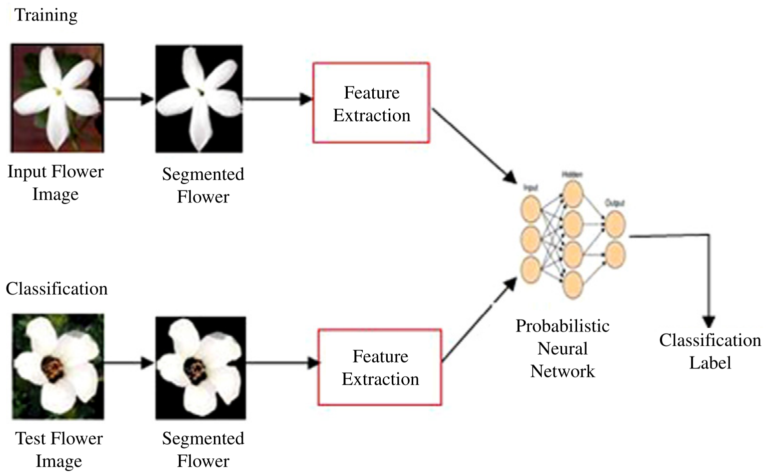

Fig. 1. Block diagram of the proposed work.

Fig. 2. Segmentation results on a few sample images. (a) Input images and (b) segmented images.

are extracted. These features are queried to the probabilistic neural network to know the flower class label. The block diagram of the proposed method is given in Fig. 1.

2.1. Flower segmentation

The first step in flower classification is to segment the flower image by removing the unwanted background region. In general, autonomous segmentation is one of the most difficult tasks in image processing. Flowers in images are often surrounded by greenery in the background. In order to avoid matching the green background region, rather than the desired foreground region, the image has to be segmented. To segment the flower image, we use a semi-automated threshold-based segmentation algorithm [11]. A given image is transformed to the HSV plane and an intensity histogram corresponding to each channel is extracted. The histogram intensity values corresponding to two dominant regions belonging to the background and the flower are identified. Based on these intensity values, the flower is segmented. Fig. 2 shows the results of flower segmentation using the threshold-based method on a few sets of images with a cluttered background.

2.2. Feature extraction

As our interest is to study only texture features for flower classification, for a segmented flower image we extract the color texture moments [12], the gray level co-occurrence matrix [13] and the Gabor response matrix for texture analysis. The following subsection gives a brief introduction to color texture moments, the GLCM and Gabor texture features.

2.2.1. Color texture moments (CTMs)

Yu et al., in [12], developed combined color and texture moments for image retrieval. In their work, the image was converted to different color spaces, namely, RGB, HSV, YUV, and (SVcosH, SVsinH, V). A local Fourier transform (LFT) was performed on all the channels of the image with eight different templates, and for each resulting channel image the first two moments were calculated, and 48 features were obtained for a given single image. Experimentally, they found that the (SVcosH, SVsinH, V) color space is better than the other color spaces, and these features were termed color texture moments. In our work, we have used these color texture moments as one of the features.

2.2.2. Gray level co-occurrence matrix (GLCM)

Texture feature calculations use the contents of the GLCM to give a measure of the variation in intensity at a pixel of interest. First proposed by Haralick et al. [13] in 1973, they characterize the texture using a variety of quantities derived from second-order image statistics. Co-occurrence texture features are extracted from an image in two steps. First, the pairwise spatial co-occurrences of pixels separated by a particular angle and distance are tabulated using a gray level co-occurrence matrix (GLCM). Second, the GLCM is used to compute a set of scalar quantities that characterize different aspects of the underlying texture. The GLCM is a tabulation of how often different combinations of gray levels co-occur in an image or image section [13]. The GLCM is an $N \times N$ square matrix, where N is the number of different gray levels in an image. An element $p(i, j, d, \theta)$ of a GLCM of an image represents the relative frequency, where i is the gray level of pixel p at location (x, y) and j is the gray level of a pixel located at a distance d from p in the orientation θ. While GLCMs provide a quantitative description of a spatial pattern, they are too unwieldy for practical image analysis. Haralick et al. [13] thus proposed a set of scalar quantities for summarizing the information contained in a GLCM. They originally proposed a total of 14 quantities, or features; however, typically only subsets of these are used [14]. The five GLCM-derived features contrast, homogeneity, energy, entropy, and correlation are extracted and used in this work.

2.2.3. Gabor filter responses

Texture analysis using filters based on Gabor functions falls into the category of frequency-based approaches. These approaches are based on the premise that the texture is an image pattern containing a repetitive structure that can be effectively characterized in a frequency domain, such as the Fourier domain. An attractive mathematical property of Gabor functions is that they minimize the joint uncertainty in space and frequency. They achieve the optimal trade-off between localizing the analysis in the spatial and frequency domains [14]. Also, a Gabor filter is a linear filter whose impulse response is defined by a harmonic function multiplied by a Gaussian function. Because of the multiplication–convolution property (Convolution theorem), the Fourier transform of a Gabor filter's impulse response is the convolution of the Fourier transform of the harmonic function and the Fourier transform of the Gaussian function, and it is given by

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right)\cos\left(2\pi\frac{x'}{\lambda} + \psi\right). \quad (1)$$

Here $x' = x\cos\theta + y\sin\theta$ and $y' = x\sin\theta + y\cos\theta$, and λ represents the wavelength of the cosine factor, θ represents the orientation of the normal to the parallel stripes of a Gabor function, ψ is the phase offset, σ is the Gaussian envelope, and γ is the spatial aspect ratio specifying the ellipticity of the support of the Gabor function. A filter bank of Gabor filters with various scales and rotations is created. In this work, we have considered scales of 0, 2, 4, 6, 8, and 10 and orientations of 0°, 45°, 90°, and 135°. For each response image obtained we extract the first three moments as features.

2.3. Flower classification

We have used a probabilistic neural network (PNN) as a classifier. The motivation behind using a PNN is due to its simple structure and training manner. The most important advantages of a PNN are that (i) the training is easy and instantaneous, (ii) additionally, it is robust to noisy examples and (iii) the speed of a PNN is high. For more information on PNNs, readers are advised to refer to [15].

3. Experimental results

In this work, we have created our own database despite the existence of other database, as the existing databases have less intra-class variations or no change in viewpoint. In order to create the database we collected flower images from the World Wide Web, and we also took photographs of flowers that can be found in and around Mysore city, Karnataka, India. The images were taken to study the effect of our proposed method with large intra-class variation. The dataset consists of 35 species of flower, with 50 images of each. The images are rescaled so that the smallest dimension is 250×250 pixels.

Fig. 3(a) shows sample images of 25 classes and Fig. 3(b) shows 5 samples of 5 classes showing high intra-class variability. The large intra-class variability and the small inter-class variability make this dataset very challenging, and subtle. Given a set of training samples, a probabilistic neural network is trained, and later it is used to classify the given test flower. In this experiment, we intend to study the effect of database size on classification accuracy. The experiment was conducted on databases of 15, 20, 25, 30, and 35 classes under varying training samples from 20 to 40, with a step of 1 per class.

The experimental results for various numbers of classes and training samples are shown in Figs. 4–8. In Table 1, we have compared the effects of all the three features (CTMs, GLCM, and Gabor responses) for various database size by fixing the number of training samples to be 60% and keeping the remaining 40% for testing. Individually, for 15 and 20 classes, the combination of CTMs, GLCM and Gabor responses achieves good accuracy, whereas for 25 and 30 classes the combination of CTMs and Gabor responses achieves good classification accuracy. Finally, for 35 classes, the combination of CTMs, GLCM and Gabor responses gives a good classification accuracy of 79%. Thus it is clear that larger the number of classes the combination of all the texture features perform better.

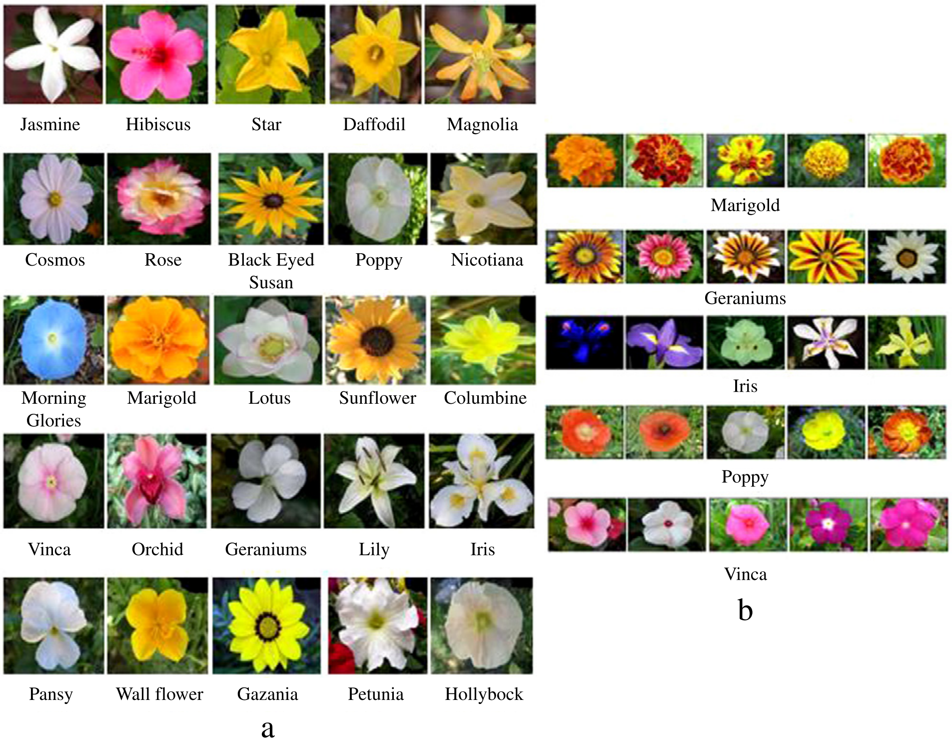

Fig. 3. (a) Sample flower images of 25 flower classes considered in this work. (b) Sample images of five different classes with large intra-class variations.

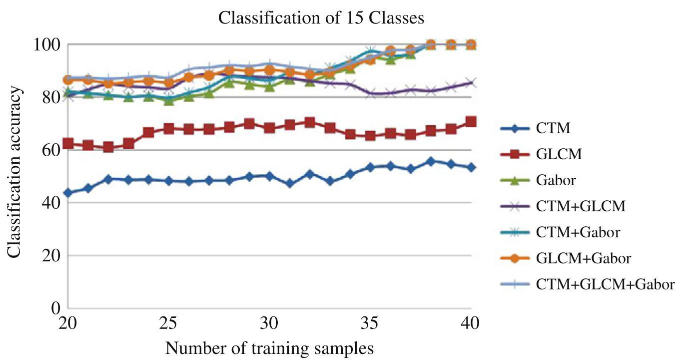

Fig. 4. Classification accuracy for various training samples for 15 classes using different combinations of features.

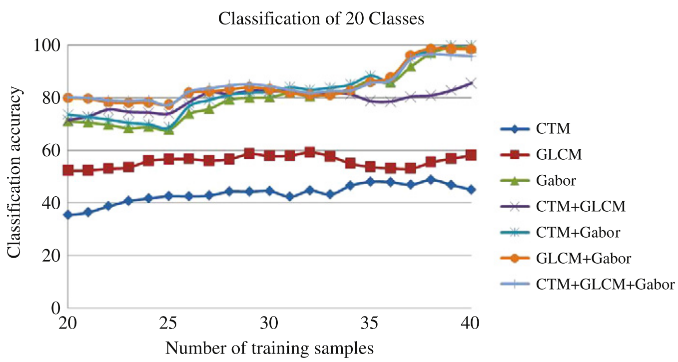

Fig. 5. Classification accuracy for various training samples for 20 classes using different combinations of features.

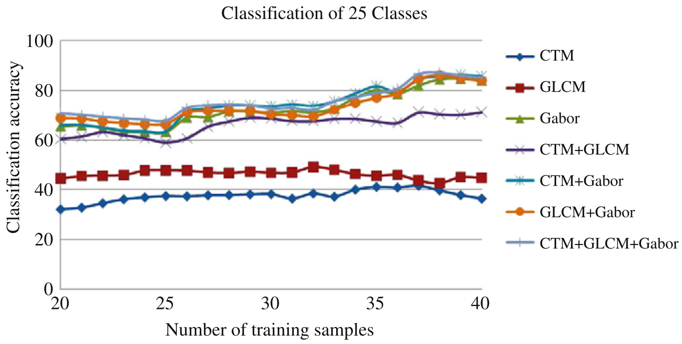

Fig. 6. Classification accuracy for various training samples for 25 classes using different combinations of features.

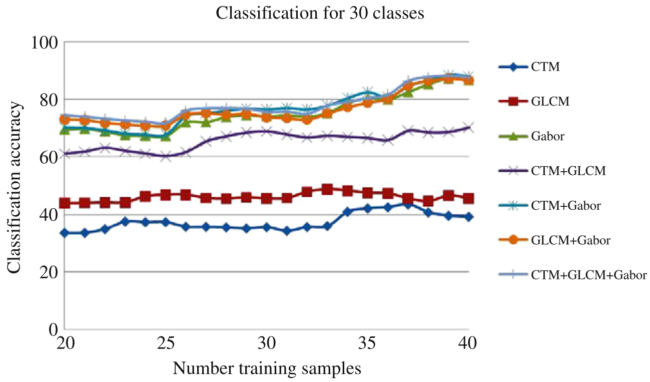

Fig. 7. Classification accuracy for various training samples for 30 classes using different combinations of features.

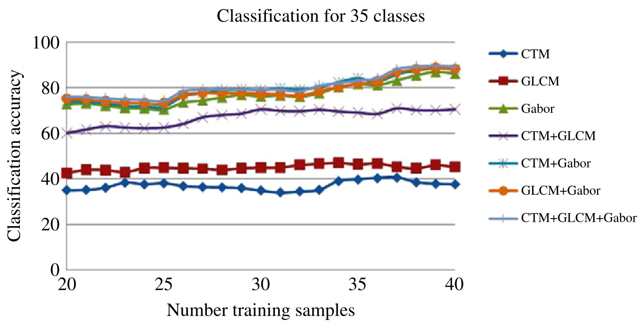

Fig. 8. Classification accuracy for various training samples for 35 classes using different combinations of features.

Table 1
Accuracy of the proposed model for various database sizes for different combinations of features.

Features	Database size				
	15	20	25	30	35
CTMs	50	44.5	38.2	35.66	35
GLCM	68.3	57.75	46.8	45.66	44.85
Gabor	84	80.25	71	74.16	76.14
CTMs + GLCM	87.33	82.5	68.6	69	70.57
CTMs + Gabor	86.33	82.25	**73.6**	**76.5**	78.85
GLCM + Gabor	90.33	83.25	70.4	73.83	77.42
CTMs + GLCM + Gabor	**92.66**	**84.5**	72.6	75.66	**79**

3.1. Comparison with previous work

In order to study the effectiveness of the proposed method with that of the other well-known methods in the literature, we carried out a qualitative comparative analysis of the results on our data set. The comparative analysis can be seen in Table 2. We compare the performance of our method on the 17-class flower dataset which was introduced in [2]. In [2], the

Table 2
Qualitative comparison of the proposed method with other well-known methods of flower classification.

Title	Species	Size	Features	Classifiers	Accuracy in%
[2]	17	1360	1. Color vocabulary 2. Shape vocabulary 3. Texture vocabulary 4. Combined vocabulary	Nearest-neighbor classifier	71.76
[4]	103	8189	1. Color 2. SIFT on the foreground region 3. SIFT on the foreground boundary 4. Histogram of gradients	Multiple kernel classifier	72.8
[7]	17	1360	1. Color vocabulary 2. Shape vocabulary 3. Texture vocabulary 4. Combined vocabulary	Support vector machine	82.55
Proposed method	35	1750	GLCM Gabor CTMs CTMs + GLCM CTMs + Gabor GLCM + Gabor CTMs + GLCM + Gabor	Probabilistic neural network	44.85 76.14 35 70.57 78.85 77.42 **79**

features are visual word histograms of color, shape, and texture. The nearest-neighbor classifier using a weighted distance on the three histograms has given a recognition rate of 71.76%. Using the same features, but a multiple kernel classifier, [7] achieves a recognition performance of 82.55%, on the same 17-class database. Finally, using the features computed in [4] and the multiple kernel classifier leads to a performance of 88.33% in [4].

Nilsback et al., in their recent work [4], have designed a flower classification system for a large database of flowers consisting of 103 flower species. They have used color features in the HSV plane and SIFT features extracted on the foreground region and foreground boundary. For classification they used a multiple kernel classifier, and they achieved 72.8% classification accuracy. Though it appears to be less, the number of classes is large. Compared to the work in [2,7], our dataset contains 35 classes and has achieved 79% of classification accuracy, which is comparable. On the other hand, for 15 and 20 classes, we can observe that the combination of all features yields 92.66% and 84.5% classification accuracy. In the proposed method, it should be noted that the combination of CTMs, GLCM and Gabor responses performs better than the individual features themselves.

4. Conclusion

In this paper, we have proposed a probabilistic neural network-based flower classification method with the use of texture features. Suitable texture features such as CTMs, GLCM, and Gabor responses are explored for the purpose of flower classification. It is observed that using the proposed textural features one can achieve relatively a good classification accuracy when compared to any other available features. We have created our own database of flowers of 35 classes, each containing 50 flower images and conducted an experiment under varying database size and we studied the size effect on the classification accuracy. The experimental results have shown that using combined features outperforms any individual feature.

References

[1] M. Das, R. Manmatha, E.M. Riseman, IEEE Intelligent Systems 14 (1999) 24–33.
[2] Nilsback Maria-Elena, A. Zisserman, in: CVPR'06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1447–1454.
[3] Y.Y. Boykov, M.P. Jolly, in: Eighth IEEE International Conference on Computer Vision, 2001, ICCV 2001, Proceedings, vol. 1, pp. 105–112.
[4] Nilsback M. Elena, Z. Andrew, in: ICVGIP'08: Proceedings of the 2008 Sixth Indian Conference on Computer Vision, Graphics and Image Processing, pp. 722–729.
[5] Nilsback Maria-Elena, Z. Andrew, Image and Vision Computing 28 (2010) 1049–1062.
[6] Y. Yoshioka, H. Iwata, R. Ohsawa, S. Ninomiya, Euphytica 139 (2004) 179–186.
[7] M. Varma, D. Ray, in: IEEE 11th International Conference on Computer Vision, pp. 1–8.
[8] T. Saitoh, K. Aoki, T. Kaneko, in: 17th International Conference on Pattern Recognition, ICPR'04, pp. 27–30.
[9] E.N. Mortensen, W.A. Barrett, in: SIGGRAPH'95: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, pp. 191–198.
[10] T. Saitoh, T. Kaneko, in: 15th International Conference on Pattern Recognition, ICPR'00, pp. 507–510.
[11] R.C. Gonzalez, R.E. Woods, S.L. Eddins, Digital Image Processing Using MATLAB, Prentice-Hall, Inc., 2003.
[12] H. Yu, M. Li, H.-J. Zhang, J. Feng, in: Proceedings of International Conference on Image Processing, pp. 929–932.
[13] R.M. Haralick, K. Shanmugam, I. Dinstein, IEEE Transactions on Systems, Man and Cybernetics 3 (1973) 610–621.
[14] S.D. Newsam, C. Kamath, in: SPIE Conference on Data Mining and Knowledge Discovery: Theory, Tools, and Technology, pp. 21–32.
[15] P.D. Wasserman, Advanced Methods in Neural Computing, John Wiley and Sons, Inc., 1993.