MICA at ImageClef 2013 Plant Identification Task

Thi-Lan LE, Ngoc-Hai PHAM
International Research Institute MICA – UMI2954 – HUST
Thi-Lan.LE@mica.edu.vn , Ngoc-Hai.Pham@mica.edu.vn

I. Introduction

In the framework of ImageClef 2013 [1], plant identification task [2], we have submitted three runs. For the first run named Mica Run 1, we employ GIST descriptor with k-nearest neighbor (kNN) for all sub-categories. Concerning Mica Run 2, we observe that global descriptors such as color histogram and texture are able to distinguish classes of two sub-categories that are flower and entire. For the others sub-categories, we still employ GIST descriptor and kNN. Based on our work for leaf identification, for the third run (named MICA run 3), we have proposed to apply our method for leaf images for both SheetasBackground and Natural background. For the remaining subcategories, we used the same method as the two first runs. Concerning our method for leaf images, we firstly apply Un-sharp Masking (USM) on SheetAsBackground images. Then, we extract Speeded-Up Robust Features (SURF). Finally, we used Bag-of-Words (BoW) for calculating feature vector of each image and Support Vector Machine (SVM) for training the model of each class in training dataset and for predicting class id of new images. In this paper, we describe in detail the algorithms used in our runs.

II. Our plant identification methods

1. Plant identification method of MICA run 1

Results of variety of state of the art scene recognition algorithms [3] shown that GIST features[1] [11] obtains an acceptable result of outdoor scene classification (appr. 73 – 80 %). Therefore, in this study, we would like to investigate if GIST features are still good for plant identification. In this section, we briefly describe procedures of GIST feature extractions proposed in [4].

To capture remarkable/considering of a scene, Oliva et al in [4] evaluated seven characteristics of a outdoor scenes such as naturalness, openness, roughness,

[1] *Gist feature present a brief observation or a report at the first glance of a outdoor scene that summarizes the quintessential characteristics of an image*

expansion, ruggedness, so on. The authors in [11] suggested that these characteristics may be reliably estimated using spectral and coarsely localized information. Steps to extract GIST features are explained in [4]. Firstly, an original image is converted and normalized to gray scale image I(x,y). We then apply a prefiltering to reduce illumination effects and to prevent some local image regions to dominate the energy spectrum. The image I(x,y) is decomposed by a set of Gabor filters. The 2-D Gabor filter is defined as follows:

$$h(x,y) = e^{-\frac{1}{2}\left(\frac{x^2}{\delta_x^2}+\frac{y^2}{\delta_y^2}\right)} e^{-j2\pi(u_0 x + v_0 y)}$$

The parameters (δ_x, δ_y) are the standard deviation of the Gaussian envelope along vertical and horizontal directions; (u_0, v_0) refers to spatial central frequency of Gabor filters. The configuration of Gabor filters contains 4 spatial scales and 8 directions. At each scale (δ_x, δ_y), by passing the image I(x,y) through a Gabor filter h(x,y), we obtain all those components in the image that have their energies concentrated near the spatial frequency point (u_0, v_0). Therefore, the gist vector is calculated using energy spectrum of 32 responses. We calculated averaging over each grid of 16 x 16 pixels on each response.

Totally, a GIST feature vector is reduced to 512 dimensions. After feature extraction procedure, K-Nearest neighbor (K-NN) classifier is selected for classification. Given a testing image, we found K cases in the training set that have minimum distance between the gist vectors of the input images and those of the training set. A decision of the label of testing image was based on majority vote of the K label found. The fact that no general rule for selected appropriate dissimilarity measures (Minkowsky, Kullback-Leibler, Intersection..). In this work, we select Euclidian distance that is usually realized in the context of image retrieval. In our run, the value of K is 32.

2. Plant identification method of MICA run 2

Concerning the second run, for two sub-categories (flower and stem), we apply global descriptors that are color histogram combining with color moment and texture. These features are described in [5]. For the remaining subcategories, we still use GIST and kNN as described in the first run.

3. Plant identification method of MICA run 3

For this run, based on the obtained result of our work on leaf identification, we apply our method for leaf images for both SheetasBackground and Natural background. This method is shown in Fig. 1.

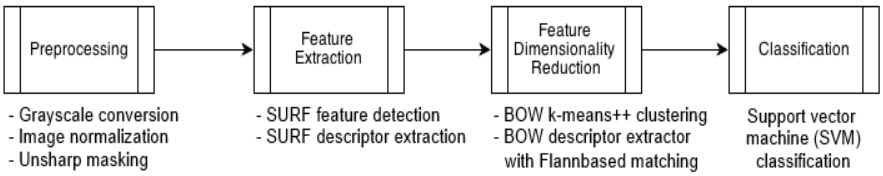

Figure 1: Plant identification method for Leaf category in MICA run 3

a. Preprocessing

First of all, segmentation methods are usually used for leaf shape recognition, while our search focuses on local features such as leaf veins and textures. Secondly, while segmentation methods work well with most uniformed background image data, they may not work well with complex background image. Thus, using segmentation methods in our work may constraint the possibility of further system development.

Instead of using segmentation methods for image preprocessing, we decided to apply image normalization and Unsharp Mask (USM) algorithms for grayscale-converted image. These preprocessing algorithms help enhance the detail of our input image as well as improve system performance.

The preprocessing procedure will be taken in three main steps:

- Grayscale conversion: Convert the original image to grayscale image, which is a matrix of pixel intensities.
- Image normalization: Change the range of pixel intensity values in order to enhance the grayscale image.
- Unsharp masking: Sharpen the image local details to help feature extraction more accurate.

b. Grayscale Conversion

Grayscale images (also known as intensity level image) are digital image in which the value of each pixel is a single sample, that is carries only intensity information. Grayscale images are commonly called black-and-white image and exclusively composed of shades of gray, varying from weakest intensity (black) to strongest intensity (white) [6].

A common strategy to convert image to grayscale is to match the luminance of the grayscale image to the luminance of the color image. Usually, the luminance component of the grayscale image in the YUV and YIQ models used in PAL and NTSC is calculated by:

$$Y' = 0.299R + 0.587G + 0.144B$$

R, G and B in the above equation represent Red, Green and Blue channels respectively. The coefficients represent human perception of colors, in particular that humans are more sensitive to green and least sensitive to blue.

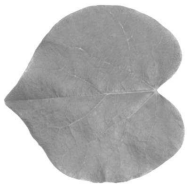

Figure 2: A grayscale leaf image

c. Image normalization

Image normalization, also known as dynamic range expansion [7], is an image processing technique that changes the range of pixel intensity values. Sometimes, it is referred to as contrast stretching or histogram stretching method because of its ability to enhance photographs which have poor contrast due to glare.

The main purpose of dynamic range expansion is to bring the image in to a range that is more familiar to normal sense. Its motivation is often to achieve consistency in dynamic range of images to avoid mental distraction or fatigue. In our research, image normalization algorithm helps balance the intensity level distribution of our image. Thus, normalized image has better contrast which results in a clearer representation of local leaf characteristics.

Normalization transforms an n-dimensional grayscale image:

$$I: \{X \subseteq R^n\} \rightarrow \{Min, \dots, Max\}$$

With intensity values in the range , into a new image:

$$I_N = \{X \subseteq R^n\} \rightarrow \{newMin, \dots, newMax\}$$

With intensity values in the new range . The linear normalization of a grayscale digital image is performed according to the formula:

$$I_N = (I - Min)\frac{newMax - newMin}{Max - Min} + newMin$$

In our research, we desire to have all input images intensity levels to be normalized from 0 to 255. The normalization is applied directly to the grayscale image produced from grayscale conversion process above.

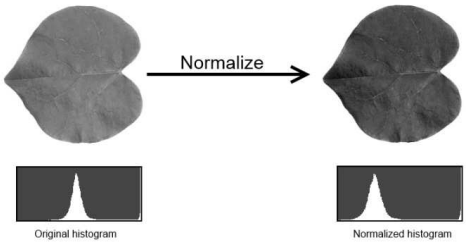

Figure 3: Image normalization produces better image contrast

d. Unsharp masking (USM)

Unsharp masking (USM) is a digital image processing technique which sharpens the image to amplify the local details. An unsharp mask cannot create additional detail, but it can greatly enhance the appearance of detail by increasing small-scale acutance.

Digital unsharp masking can be done by combining two images: The original image is called negative image and the blur version of the original image which called positive image.

First of all, we make the positive image by applying Gaussian blur filter to a clone version of the normalized grayscale image produced from the last step. As our original leaf image is a normalized grayscale image, we apply the equation of Gaussian filter in one dimension matrix using the following Gaussian function:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

The blurred version of the original is then subtracted away from the original to detect the presence of edges, creating the unsharp mask (effectively a high-pass filter). Contrast is then selectively increased along these edges using this mask leaving behind a sharper final image. An unsharp mask improves sharpness by increasing acutance, although resolution remains the same. In short, there are two steps in unsharp masking:

- Step 1: Detect Edges and Create Mask
- Step 2: Increase Contrast at Edges

The process of unsharp masking is described below:

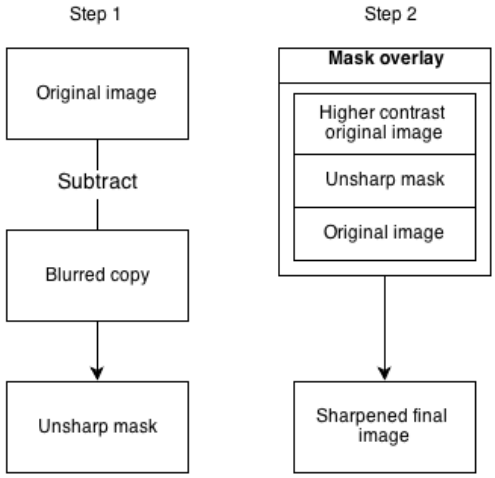

Figure 4: Two steps of unsharp masking procedure

Note that the “mask overlay” is when image information from the layer above the unsharp mask passes through and replaces the layer below in a way which is proportional to the brightness in that region of the mask. The upper image does not contribute to the final for regions where the mask is black, while it completely replaces the layer below in regions where the unsharp mask is white.

Final image shows better local characteristics of the leaf as we expected.

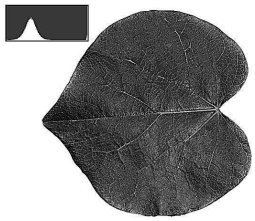

Figure 5: Normalized and sharpened grayscale leaf image with new histogram

e. Feature Extraction

After processing all input images, the image matrices will then be passed through SURF feature detector and descriptor extractor, which is the local feature extraction approach. We prefer local feature extraction approach over shape feature extraction because it gives us the advantage of recognizing and classifying complex background leaf images. In our research, we take full advantage of the implemented OpenCV methods, which are configurable and bug-free.

In the first step of feature extraction, SURF feature detector received preprocessed image as input. It attempted to detect points of interest (keypoints) over the whole image using Fast Hessian detection algorithm. In the first run of our system, the detector operated as it with no parameterization, meaning that the minimum Hessian threshold, number of octaves and number of octave layers are default. However, in later experiment, we tried to modify the parameters in order to challenge the system at different running conditions.

If no keypoints were found in an input image, the system will give a warning message, ignore the error image and continue the extraction process.

Below figure illustrates the keypoints founded by SURF feature detector.

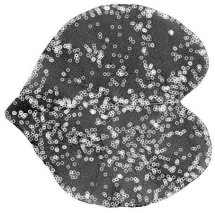

Figure 6: Detected keypoints

The second step of feature extraction involves computing SURF descriptors based on detected keypoints (known as descriptor extraction process). From detected keypoint, descriptor based on sum of Haar Wavelet responses is computed. The computed descriptors for each image are then stored into the memory as a matrix vector of floating point numbers, which represents the detected keypoints, their size and orientation. The number of rows in descriptor matrix represents the number of detected keypoints, while the number of columns is the size of each keypoint, typically 64 or 128.

In our research, we ran feature extraction algorithm on a computer with high memory capacity, thus there were no limitation to the number of detected keypoints.

In practice, we expect to have the maximum number of keypoints to be 2000, in order to make our final software runs smoothly on any computer with average memory capacity.

f. Feature Dimensionality Redution

In combination with SURF feature extraction method, we use BOW model to reduce the dimensionality of our computed SURF descriptors. In our research, we use a slightly different version BOW model from OpenCV library which is composed of two main components: BOW k-means trainer and BOW image descriptor extractor.

As discussed above, computed SURF descriptors of each image will be stored in the machine memory as a descriptor matrix. This matrix will then become input for BOW k-means trainer in order to cluster the histograms of descriptors which have similar characteristics into separate visual words. BOW k-means trainer takes the pre-defined dictionary size to be parameter K for k-means++ algorithm introduced in [8]. This parameter has impact on not only the speed of the system but also its performance. Determining the suitable dictionary size parameter is a real challenge and there were no existing research about this issue. Hence, we chose to give BOW trainer a fix dictionary size value each time we ran the system. Specifically, we selected four dictionary size values which are 256, 512, 1024 and 2048. With different dictionary size values, the system produced different classification outcomes, which will be discussed in the next chapter of this work.

After clustering the histograms of descriptors into different visual words using BOW trainer, we saved the clustered dictionary into local machine storage as XML file. The dictionary produced by BOW trainer is actually a matrix of floating point number in which the number of columns is the size of SURF descriptor, and the number of rows is the size of BOW dictionary. Below figure shows a sample dictionary data.

```
<?xml version="1.0"?>
<opencv_storage>
<Dictionary type_id="opencv-matrix">
  <rows>256</rows>
  <cols>64</cols>
  <dt>f</dt>
  <data>
    1.26852980e-003 -9.53011971e-004 2.46735802e-003 2.27324921e-003
    6.21023076e-003 5.32565173e-004 1.06226299e-002 7.79518718e-003
    5.93685370e-004 1.08432339e-003 5.26582776e-003 5.57951536e-003
    -5.68680553e-005 3.78244877e-005 8.91003467e-004 9.35436867e-004
    -1.88452774e-004 -3.32415588e-002 2.63734031e-002 4.17957082e-002
    2.30729692e-002 -3.70868057e-001 1.63490057e-001 4.12834823e-001
    2.00425088e-001 -1.66891620e-001 2.58941144e-001 2.26805508e-001
    -1.53817795e-003 7.04522303e-004 6.29969873e-003 4.93702479e-003
    -4.88365768e-003 -5.46044856e-003 1.54936882e-002 1.53211374e-002
    4.15388122e-003 -2.28936528e-003 6.38177991e-002 6.61938414e-002
    4.24850792e-001 5.45422062e-002 4.54869509e-001 1.28891572e-001
    -2.96043209e-003 -1.57665706e-003 7.63143413e-003 6.35610241e-003
    -4.47499828e-004 2.98397004e-004 1.77854300e-003 1.65892940e-003
    1.49319563e-002 1.20269759e-002 2.32196581e-002 2.28974521e-002
```

Figure 7: Sample dictionary data

BOW image descriptor extractor component then takes the dictionary data as its input and extracts (computes) the BOW descriptor matrix for each input image

keypoint. This process involves using Fast Library for Approximate Nearest Neighbors (FLANN) matcher [9] to match between features. The result BOW descriptors are stored as actual feature set for our classifier.

g. Classification

The classification layer of our system takes the feature set from the last layer as its input. In our research, we prefer using Support Vector Machine (SVM) as our supervised learning method. Similar to other supervised learning approaches, depend on the type of input data, SVM has two main functions which are training and testing. OpenCV library has already provided all needed functions for SVM training and testing, thus we made full use of the library for our system benefits.

SVM classifier takes two main parameters for classification:

- SVM type: Type of SVM formulation. In our research, we define this parameter as C_SVC or C-Support Vector Classification. It allows imperfect separation of classes for n-class classification with penalty multiplier C for outliers.
- *Kernel type:* Type of SVM kernel. Here we use Radial Basis Function (RBF) kernel for our classification which has the equation $K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}, \gamma > 0$. This is the most popular kernel type for SVM.

Other support parameters are selected automatically by the system library function which has less impact on the classification performance.

SVM training time depends mostly on the size of the input data. When experimenting with Flavia dataset, we noticed that our training time range from 1 to 2 hours depends mostly on the size of BOW dictionary.

III. Results and Discussion

Concerning SheetAsBackground, our third run has obtained the highest score among three runs (0.314) while the first and the second have the low value of score (0.09). This means that the method based on SURF, BOW and SVM is robust for SheetAsBackground category.

With the Natural Background, MICA run 2 has a greater value of score than MICA run 1 and MICA run 3. The main reason is that the global descriptor used in MICA run 2 is effective for the Flower category. The obtained score for Leaf category for MICA run 3 is relatively high. This prove that the method based on SURF, BOW and SVM is robust not only for leaf with SheetAsBackground but also for leaf with natural background.

The results of our runs show that GIST is robust for scene classification but it is not relevant descriptor for plant identification.

Table 1: Obtained results of our runs with natural background

	MICA run 1	MICA run 2	MICA run 3
Entire	0.016	0.016	0.016
Flower	0.013	**0.086**	0.013
Fruit	0.048	0.048	0.048
Leaf	0.014	0.014	**0.11**
Stem	0.014	0.014	0.014
Naturalbackground	0.023	0.053	0.042

References

1. Caputo, B., et al. *ImageCLEF 2013: the vision, the data and the open challenges*. in *CLEF2013*. 2013.
2. Goëau, H., et al. *The ImageCLEF 2013 Plant Identification Task*. in *CLEF 2013*. 2013. Valencia, Spain, 2013.
3. Quattoni, A. and A.Torralba, Recognizing Indoor Scenes. In Proceeding of the International Conference on Computer Vision and Pattern Recognition, 2009: p. 1-8.
4. Oliva, A. and A. Torralba, Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. Int. J. Comput. Vision, 2001. 42(3): p. 145-175.
5. Thi-Lan Le, Alain Boucher, An interactive image retrieval system: from symbolic to semantic, International Conference on Electronics, Information, and Communications (ICEIC), 16-18 août 2004, Hanoi (Vietnam).
6. Johnson, S., *Stephen Johnson on Digital Photography*. 2006: O'Reilly Media.
7. Rafael C. González, R.E.W., *Digital Image Processing*. 2007: Prentice Hall.
8. David Arthur, S.V. *k-means++: The Advantages of Careful Seeding*. in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. 2007.
9. Marius Muja, D.G.L. *Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration*. in *VISAPP International Conference on Computer Vision Theory and Applications*. 2009.