Plant Leaf Recognition Using Texture Features and Semi-Supervised Spherical K-means Clustering

Shadi Alamoudi, Xia Hong, Hong Wei
Department of Computer Science
School of Mathematical, Physical and Computational Sciences,
University of Reading, Reading, UK, RG6 6AY

***Abstract*—Automatic plant leave recognition using digital images and machine learning techniques is an important task. The disadvantage of supervised learning techniques is that they are limited to learn from labelled datasets which are often expensive to obtain. In this paper, a novel decision fusion framework is proposed by combining semi-supervised clustering with the well known image features analysis methods in computer vision. Initially the leave image features are generated by applying the Grey Level Co-occurrence Matrix analysis to the processed leave images transformed by Gabor or Laplacian of Gaussian filters. Then an on-line spherical k-means clustering technique, guided by a minimum number of labelled leaves, is used to train the base classifiers. The final decision of classification is produced by selecting classifier which produces the max-cosine value amongst the baseline classifiers. Comparative experiments have been carried out to demonstrate that proposed approaches are suited for automatic leave type recognition.**

I. Introduction

Plants are generally recognized using their distinct features (e.g. leaf colour, leaf shape, leaf texture, trunk, type of fruit, flower etc.) [1]. Plant leaves have a nearly 2D structure and their scanned or photographed images can be used by a computer to extract many features [2]. All such images can be stored in a database while each new image can be compared (using its features) against existing images in the database to recognize its type. Ideally, all the available features of a plant leaf should be used for recognizing its type. However, comparing a large number of features is a computationally complex task and is a time-consuming operation. Hence, practically, only a small number of features are used for this purpose. However, the task of accurately recognizing plants using only a small number of leaf features is not straight forward. If irrelevant features are used, the accuracy of the plant recognition system can be affected. For example, leaf colour is not a very effective feature since many plant types share the same leaf colour. On the other hand, geometrical features (such as aspect ratio, form factor etc.) are very effective, especially when many of these features are used in combination.

Another aspect of leave type recognition is using the extracted features of a plant leave to identify its type. The common approach of supervised learning technique of pattern recognition can be used, i.e. the feature input is mapped to an output based on input-output pairs [3]. In such schemes, labelled training data is used to infer the relationship between the input and output functions. This relationship can be used to map new inputs to known outputs (types). Researches in the area of automatic plant type recognition have gained momentum over a recent few decades. Amongst the first works were Guyer et al.'s statistical pattern classification techniques [4] which were used to identify plants. Many feature extraction techniques have been devoted to feature extraction of plant leaves. Leaf shape [5], [6] is one such feature which has been used extensively. The shape related features include the physiological length/width of the leaf, slimness/aspect ratio, area and perimeter, form factor/roundness, solidity, longest distance, and curvature [7] etc. For example, Im et al. [8] identified a maple leaf using its shape feature. However, it is not clear whether the method can be extended to identify other type of leaves. Similarly, Wang et al. [9] used a centroid-contour distance curve to represent the shape of a leaf. However, the results show that using only shape-based features, limited accuracies can be achieved.

Another important feature which can be used for plant identification is the leaf texture [10]. It can be effectively modelled with the help of a Gabor filter (GF) and a Grey Level Co-occurrence Matrices (GLCM) [11]. Chaki et al. [11] combined the shape-based features with texture features to further enhance the accuracy of plant image recognition techniques. They modelled the shape of a leaf using curvelet transform (CT) and moment invariants [12]. Hickey et al. [13] showed that the leaf venation is also an important feature which can be used to effectively identify plants [7], [14]. Caglayan et al. [15] used leaf colour as a feature to identify plants. In particular, they used the mean, standard deviation, and histograms of the Red, Green, and Blue intensity channels for plant identification. Zhang and Zhang [16] showed that combining different physical features of a leaf can improve the accuracy of leaf (and hence, plant) recognition techniques. They combined colour, texture, perimeter, and area of a tobacco leaf for its identification. However, since the method was only applied on tobacco leaves, its application to other types of leaves may not be as efficient as shown in their works. Mzoughi et al. [17] showed that leaf recognition can be made more effective if domain-specific or botany related knowledge is also used on top of an analysis using leaf characteristics.

In the context of plant leave recognition task, classification techniques are mainly based on Artificial Intelligence (AI) principles. For example, Wu et al. [2] used Artificial Neural Networks (ANN) for plant identification with the help of its

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

morphological features. However, they did not validate their results using a well-known image database. On the other hand, Wu et al [18] used Probabilistic Neural Networks (PNN) for plant image recognition with the help of twelve leaf features. However, the method requires human intervention, hence, it is not fully automatic. Similarly, Belhumeur et al. [19] used Expectation Maximization (EM) as a classification technique. However, there are large variations between the accuracies achieved by this method for different datasets. Du et al. [20] used Move Median Centre (MMC) hyper-sphere classifier in their plant recognition technique. Caglayan et al [15] observed that the accuracy of the classification technique depends on the type of features selected. They tested the performance of four different classification techniques (i.e., NN, Support Vector Machine (SVM), Naïve Bayes, and Random Forest (RF)). They found that when only shape features are used, RF has the highest accuracy while SVM performs the worst. On the other hand, when colour features are also used, RF still performs the best but Naïve Bayes performs worse than SVM. Chaki, Paresh, and Bhattacharya [11] used two neural classifiers for plant image recognition: a neuro-fuzzy controller (NFC) and a feed-forward back-propagation multi-layered perceptron (MLP). They found that shape-based features are not good features for MLP and that texture-based features are more accurate than shape-based features. Hang et al. [21], Simonyan and Zisserman [22] and Lee et al [23] applied convolutional neural network (CNN) based approaches for classification for excellent performance. However, these methods are generally slow and require a large number of labelled training samples to achieve high performance. On the other hand, real-time application requires methods that are both accurate and fast. One such method was developed by Hu et al. [24]. However, this method is based on some assumptions which may not always be true.

Alternatively the clustering technique may be used to identify plant leaves as groups. The idea in clustering is to partition a data set into a group of clusters (or subsets) such that observations within a cluster are more similar to one another than observations in other clusters [25]. Compared to the supervised approaches of classification, clustering is generally unsupervised method which is useful for applications where there are no or lack of labelled data samples, e.g. k-means clustering [26]. Alternatively, other distance metrics can be used. For example, an efficient and popular on-line spherical k-means clustering algorithm exists for clustering text documents [27]. The spherical k-means clustering algorithm is more suitable for data space with high dimensionality. Although traditional clustering techniques are based on the idea of unsupervised learning, recently much interest has been shown in 'semi-supervised' clustering where outcome variables or some other preliminary information about the clusters are known. Hence, clustering methods that can be applied to partially labelled data with other types of outcome measures are known as 'semi-supervised clustering' methods. Furthermore, in many applications, there are a small number of labelled data, which can easily be implemented as semi-supervised clustering by simply using these to form initial cluster centers. To our best knowledge, semi-supervised clustering has seldom been used for the task of plant leave recognition. In this paper, a novel decision fusion framework is proposed which combines the texture feature based technique in [11] with the on-line spherical k-means based semi-supervised clustering technique in [27]. Additionally a novel decision fusion based on maximal cosine in spherical k-means algorithm was introduced for leave type prediction.

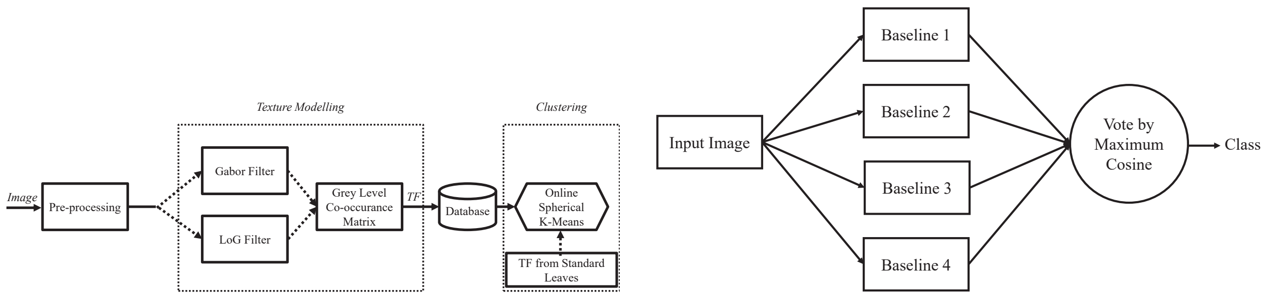
The remainder of the paper is organized as follows. In Section II-A, the proposed decision fusion framework is introduced based on the on-line spherical clustering objective for classification decision amongst baseline classifies, which uses the image texture of plant leaves as features and the on-line spherical k-means technique [27] for clustering. A pre-processing step is included in the proposed framework, as a first step. This step is described in Section II-B. Other two steps of texture feature modeling techniques are presented in Section II-C and Section II-D respectively. The proposed online spherical k-means clustering technique followed by its use of final decision fusion are introduced in Section II-E. In Section III, the experimental set up and the obtained results of the proposed framework are presented. Section IV is devoted to conclusion.

II. The Proposed Plant Leaf Recognition Method Using Texture Features and Semi-Supervised Spherical K-means Clustering

A. The Proposed Plant Leaf Recognition Framework

In this section, the proposed plant leaf recognition system is presented. As shown in Fig. 1(a), the proposed decision fusion framework is composed of baseline classifiers which has three main steps: pre-processing, texture modelling, and clustering. In the first step, an image 'Image' is pre-processed. The output of pre-processing step [11] is the signal 'gs' representing a grey scale signal, which is then used as inputs for the texture modelling stage. In this stage, there are two steps. In the first step, the signal 'gs' is filtered either using a Gabor filter or using a Laplacian of Gaussian (LoG) filter. In the second step, the filtered signal is fed to the Grey Level Co-occurrence Matrix (GLCM) block [11]. For the Gabor filter, the filtered signal is a complex signal which contains both real and imaginary parts. We compared four different types of image signals that are resultant of (i) the imaginary part of a Gabor filter [11];(ii) the real part of a Gabor filter; (iii) both the real and imaginary parts of a Gabor filter and (iv) a Laplacian of Gaussian (LoG) filter, respectively. The GLCM block [11] takes these signals as input to produce a set of texture features 'TF'. The texture features 'TF' are then stored in a database.

An Online Spherical k-Means (OSKM) clustering technique [27] which was originally used in document classification is then used to cluster these features. Additionally we propose a new semi-supervised clustering, in which the centroids of the clusters are initialized using information from a standard leaves database [28]. Note that clustering based plant leaf



(a) (b)

Fig. 1. The proposed framework for a plant leaf recognition system; (a) construction of four semi-supervised baseline classifiers based on Gabor and LOG filter; and (b) Decision fusion classification by maximizing cosine objective function of four baseline classifiers. The four baseline classifiers are based on (i) the imaginary part of a Gabor filter [11];(ii) the real part of a Gabor filter; (iii) both the real and imaginary parts of a Gabor filter and (iv) a Laplacian of Gaussian (LoG) filter, respectively.

recognition classifier for a larger number of classes is challenging, so the images belonging to ten different classes within the dataset were used, which represents a moderate number of classes. Specifically, the texture features 'TF ” are generated in the same way as training samples through pre-processing, texture modelling stages as described above. The standard leaves database consists of ten representative leaf images from each of the ten classes that are used in this work (with more details in Section III).

As shown in Fig. 1(b), the final decision for predicting any new image type is based on choosing the one of baseline classifier by the measure of maximal cosine as used in the OSKM. In the next four subsections, the pre-processing, texture features modelling (in two steps), and clustering techniques used in the proposed framework are presented in details.

B. Pre-Processing [11]

The pre-processing step is used to make the proposed framework rotation-, translation-, and scale-invariant. For rotation invariance, the major axis of the grey-scale image is aligned with the horizontal axis. For translation invariance, the grey-scale image of the leaf is then fitted within a bounding rectangle. Finally, for scale invariance, the image is rescaled to standard dimensions (or slots) as in [11].

C. Texture Modelling Step 1

In this work, we used four different types of image signals that are resultant of (i) the imaginary part of a Gabor filter (with parameters setting empirically similar to [11]);(ii) the real part of a Gabor filter; (iii) both the real and imaginary parts of a Gabor filter and (iv) a Laplacian of Gaussian (LoG) filter, respectively. Note that our proposed approach is fundamentally different from that in [11], and it only requires a single labelled image per class in training as standard leave images. The Step 1 of the texture modeling is described below.

1) Gabor Filter: The Gabor filter is a linear filter which has been found to be particularly appropriate for texture representation and discrimination [29]. A 2-D Gabor filter is combined with a 2-D Gaussian kernel and a complex sinusoid function. The modulated 2-D Gaussian function in Gabor filters is tunable in size to produce scalable kernels. The sinusoid function makes Gabor filters have controllable orientation and frequency. Due to these advantages, Gabor filters are popularly used to extract texture features in image analysis. It is believed that Gabor filters resemble the performance of the mammalian visual cortical cells, in a sense of extracting features at different orientations and scales. It has shown that Gabor filters outperform other directional filters in recognition of blood vessels in retinal fundus images [30]. In automatic change detection based on SAR images, Gabor filters were applied effectively to deal with spatial invariance [31]. The multi-resolution sensitivity of Gabor filters may be helpful for extracting meaningful features from plant leaves. The Gabor filter (GF) is used to filter the grey scale pre-processed image.

A Gabor filter 'h_{Gabor}' is based on the product of a Gaussian Kernel 'g' and a complex sinusoid 's' as

$$h_{\mathrm{Gabor}}(x,y,\sigma,u,\tau,\phi) = g(x,y,\sigma) \times s(x,y,u,\tau,\phi) \quad (1)$$

where the Gaussian Kernel is defined as

$$g(x,y,\sigma) = \frac{1}{2\pi\sigma^2}\exp\{-\frac{x^2+y^2}{2\sigma^2}\} \quad (2)$$

where σ represents the spread of the Gaussian function in x and y directions. The complex sinusoid is defined by

$$s(x,y,u,\tau,\phi) = \exp\{j2\pi(x\times u\cos\tau + y\times u\sin\tau)+\phi\} \quad (3)$$

where u represents the spatial frequency, τ represents the orientation, while ϕ represents the phase shift. The parameters of the Gabor filter are experimentally determined as $\phi = 3\pi/4$, $\tau = 0.3$ rad, $\sigma = \sigma_1 = 0.65\tau$, and $u = 0.125$. The Gabor filter is used to generate a set of complex signals by convolution of the grey scale image 'gs' with the Gabor filter h^{Gabor} as

$$J = gs \otimes h_{\mathrm{Gabor}}(x,y,\sigma_1,u,\tau,\phi) \quad (4)$$

The complex valued signal J is broken down into its real part 'rG' and imaginary part 'iG' as:

$$\begin{aligned} rG &= \Re(J) \\ iG &= \Im(J) \end{aligned} \quad (5)$$

In this work only one parameter setting was predetermined as our baseline classifier. There should be a range of parameters in Gabor filters which may also work.

2) LoG filter: Alternatively the grey scale pre-processed image can be filtered by Laplacian of Gaussian (LoG) filter. The LoG operator calculates the second spatial derivative of an image, hence the resultant filtered image is modified by highlighting regions of rapid intensity change and is therefore often used for edge detection [33] and bob detection [34].

A Laplacian of Gaussian (LoG) filter 'h_{LoG}' is defined as:

$$h_{\mathrm{Gabor}}(x,y,\sigma) = \frac{(x^2+y^2-2\sigma^2)g(x,y,\sigma)}{\sigma^4\sum_x\sum_y g(x,y,\sigma)} \tag{6}$$

The LoG filter is used to generate a set of real signals by convolution of the grey scale image 'gs' with the Log filter h_{LoG} as

$$rL = gs \otimes h_{\mathrm{Log}}(x,y,\sigma,u,\tau,\phi) \tag{7}$$

$\sigma = \sigma_2 = 0.5$ was used.

D. Texture Modelling Step 2

A Grey Level Co-occurrence Matrix (GLCM) [10] is then used as Step 2 to analyze the texture of these images 'iG','rG','rL', respectively. GLCM, concerning pairs of pixels in certain spatial relations to each other, is second order statistics to describe image properties [32]. Different from single pixel statistics, it expresses the relative frequencies $P(i,j|d,\theta)$ with which two pixels having relative polar coordinates (d,θ) appear with intensities i,j. Depending on the number N of grey levels in the original image, the raw GLCM is an N by N matrix, which is then condensed to relatively few features, such as image energy, entropy, etc. to represent image texture information. For any grey scaled image, based on the 11 GLCM based feature vector $\boldsymbol{f} = [f_1, f_2, ...f_{11}]^{\mathrm{T}}$ with $d = 1$, and used as the basis for the definition of the texture feature (TF) per image described below.

1) GLCM with imaginary part of Gabor filter: The filtered imaginary signal 'iG' is used to compute a GLCM with $d = 1$, $N = 256$. The texture feature (TF) for each image is a 44-element vector by concatenating the 11 features for each of the four angle values, as and $\theta \in \{0°, 45°, 90°, 135°\}$, there are 44 GLCM based features which are generated as

$$TF^{(iG)} = [(\boldsymbol{f}^{(iG)}_{\theta=0°})^{\mathrm{T}}, (\boldsymbol{f}^{(iG)}_{\theta=45°})^{\mathrm{T}}, (\boldsymbol{f}^{(iG)}_{\theta=90°})^{\mathrm{T}}, (\boldsymbol{f}^{(iG)}_{\theta=135°})^{\mathrm{T}}]^{\mathrm{T}} \in \Re^{44} \tag{8}$$

This is the same signal used in [11].

2) GLCM with real part of Gabor filter: Similarly, the filtered real signal 'rG' is used to compute a GLCM with $d = 1$, $N = 256$, while $\theta \in \{0°, 45°, 90°, 135°\}$. The texture feature (TF) is based on 44 GLCM features from real signal 'rG' are

$$TF^{(rG)} = [(\boldsymbol{f}^{(rG)}_{\theta=0°})^{\mathrm{T}}, (\boldsymbol{f}^{(rG)}_{\theta=45°})^{\mathrm{T}}, (\boldsymbol{f}^{(rG)}_{\theta=90°})^{\mathrm{T}}, (\boldsymbol{f}^{(rG)}_{\theta=135°})^{\mathrm{T}}]^{\mathrm{T}} \in \Re^{44} \tag{9}$$

which was not used in [11], but is investigated by this work.

3) GLCM with both real and imaginary parts of Gabor filter: Based on both real and imaginary parts of Gabor filters, there are 88 GLCM based features which given as

$$TF^{G} = [(TF^{(iG)})^{\mathrm{T}}, (TF^{(rG)})^{\mathrm{T}}]^{\mathrm{T}} \in \Re^{88} \tag{10}$$

used as the texture feature.

4) GLCM with Laplacian of Gaussian filter: Alternatively, based on the LoG filtered signal from 'lG', the texture feature vector is given by

$$TF^{(rL)} = [(\boldsymbol{f}^{(rL)}_{\theta=0°})^{\mathrm{T}}, (\boldsymbol{f}^{(rL)}_{\theta=45°})^{\mathrm{T}}, (\boldsymbol{f}^{(rL)}_{\theta=90°})^{\mathrm{T}}, (\boldsymbol{f}^{(rL)}_{\theta=135°})^{\mathrm{T}}]^{\mathrm{T}} \in \Re^{44} \tag{11}$$

where the same parameters of $d = 1$, $N = 256$ are used to produce 44-element vector by concatenating the 11 features from GLCM features of four angle values of $\theta \in \{0°, 45°, 90°, 135°\}$.

Based on above texture features, four types of baseline classifiers are investigated using semi-supervised online spherical K-means clustering as follows.

E. Semi-supervised Online Spherical K-Means Clustering using Texture Features

In the proposed framework, each of the baseline classifier is based on the Online Spherical K-Means (OSKM) algorithm [27] which incrementally maximise the cosine cost function

$$L = \sum_{i=1}^{n}\sum_{TF_i \in S_k} \frac{TF_i^{\mathrm{T}}\boldsymbol{\mu}_k}{\|TF_i\|\|\boldsymbol{\mu}_k\|} \tag{12}$$

where K is a predetermined number of clusters, and n data points TF_i are partitioned by K clusters S_k, $k = 1, ...K$ spherically using any of the the texture feature (TF) modeling schemes as described above. In OSKM [27], for a given data point TF_i, the closest cluster centre $\boldsymbol{\mu}_k(TF_i)$ is incrementally updated as

$$\boldsymbol{\mu}_k^{(new)}(TF_i) = \frac{\boldsymbol{\mu}_k(TF_i) + \eta TF_i}{\|\boldsymbol{\mu}_k(TF_i) + \eta TF_i\|} \tag{13}$$

where $\eta > 0$ is a small learning rate. The main advantage of clustering using cosine cost is that it is more suitable for data with high dimensionality, and clearly in our application the number of features is high. It is also advantageous to make use of the systematic information in the database of a given labelled data set, referred to as the standards leaves (one labelled image per class). We simply propose the new initialization scheme in which the centroids of the clusters are initialized using the respective texture features (TF) from each labelled image of standard leaves. The proposed framework is referred to as the Semi-supervised online spherical k-means (SOSKM), as shown in Algorithm 1. In this work the learning rate is set $\eta = 0.05$. The algorithm converges quickly. The number of training epochs $Iter$ is set as 25 as it is found to be appropriate. To make future prediction for a new leave image denoted as TF_i, (14) is used.

By repetitively applying Algorithm 1 using four baseline classifiers in Section II-C-II-D can be obtained based on texture features $\{TF^{(iG)}, TF^{(rG)}, TF^{(G)}, TF^{(rL)}\}$ respectively.

Algorithm 1 The semi-supervised online spherical k-means (SOSKM) for leave classification

Require: n unlabeled training images 'gs_n'. Cluster number K. Learning rate η. A predetermined number of training epochs $Iter$.

Require: For each class, a single labeled image from standard leave database 'gs_k^{lab}' , $k = 1, ..., K$.

Ensure: A partition of the data vectors given by the cluster identity vector $Y = \{y_1, ..., y_n\}$, $y_n \in \{1, ..., K\}$, and K unit-length clusters $\{\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K\}$.

1: Generate n unit-length data vectors TF_i, $i = 1, ...n$, based on features of 'gs_n' using a chosen texture modeling scheme in Section II-C-II-D.

2: Initialization: Initialize the unit-length K cluster centroid vectors. $\{\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K\}$ using texture based features from standard leaves images 'gs_k^{lab}' using the same texture modeling scheme.

3: **for** $t = 1, ..., Iter$ **do**

4:   **for** $i = 1, ..., n$ **do**

5:     For each data vector TF_i, find the closest centroid

$$y_n = \arg\max_k \{TF_i^{\mathrm{T}} \boldsymbol{\mu}_k\}, \quad \text{for} \quad k = 1, ...K \tag{14}$$

6:     Estimate each centroid as

$$\boldsymbol{\mu}_{y_n} \leftarrow \frac{\boldsymbol{\mu}_{y_n} + \eta TF_i}{\|\boldsymbol{\mu}_{y_n} + \eta TF_i\|} \tag{15}$$

7:   **end for**

8: **end for**

9: Return: $Y = \{y_1, ..., y_n\}$, $y_n \in \{1, ..., K\}$ and $\{\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_K\}$.

Algorithm 2 The proposed decision fusion framework using SOSKM baseline classifiers for new image class prediction

Require: The clusters from four baseline classifiers, as $\{\boldsymbol{\mu}_1^{(c)}, ..., \boldsymbol{\mu}_K^{(c)}\}$, where $c \in \{(iG), (rG), (G), (rL)\}$.

Require: Any unlabeled new images (test data) 'gs_{new}'.

Ensure: The predicted class label $y_{new}^{(com)}$ by combining results from baseline classifier, $y_{new}^{(com)} \in \{1, ..., K\}$.

1: For 'gs_{new}', obtain the respective features $\{TF^{(iG)}, TF^{(rG)}, TF^{(G)}, TF^{(rL)}\}$.

2: **for** $c \in \{(iG), (rG), (G), (rL)\}$ **do**

3:   Find the closest centroid of baseline classifier

$$y_{new}^{(c)} = \arg\max_k \{[TF^{(c)}]^{\mathrm{T}} \boldsymbol{\mu}_k^{(c)}\}, \quad \text{for} \quad k = 1, ...K, \tag{16}$$

4: **end for**

5: Find the classifier $c^{(best)}$ that produce maximum cosine cost function value

$$c^{(best)} = \arg\max_c \frac{[TF^{(c)}]^{\mathrm{T}} \boldsymbol{\mu}_{y_{new}}^{(c)}}{\|TF^{(c)}]\| \|\boldsymbol{\mu}_{y_{new}}^{(c)}\|}, \quad c \in \{(iG), (rG), (G), (rL)\} \tag{17}$$

where $c^{(best)}$ is the selected classifier, and the corresponding combined class label.

6: Return predicted class label using combination as $y_{new}^{(com)} = y_{new}^{(c^{(best)})}$, $y_{new}^{(com)} \in \{1, ..., K\}$, and $c^{(best)} \in \{(iG), (rG), (G), (rL)\}$.

Algorithm 2 presents the proposed decision fusion framework using SOSKM baseline classifiers for new image class prediction, in which the final leave classification result is obtained by maximizing the respective cosine cost functions on competition of $L^{(c)}$, where (c) denotes the classifier type, and it is expected that the combination of LoG with Gabor filters is expected to provide complementary features of leave images for improved classification performance. Note that in spite of possible differences in the scale of features for various baseline classifiers, their cosine cost functions are normalized measures with maximal value of one, so it can easily used for decision fusion by voting. Similarly this method is general so that other baseline classifiers may also be used, which will be our future work.

III. Experimental Studies

The experiments were performed using Matlab on a personal computer with 1.6 GHz Intel Core i5 processor, 8 GB RAM, and macOS Sierra operating system. The popular Flavia image dataset [18], [28] was used during the experiments. Note that from experience the classification performance based on a small number of classes is better, but plant leaf recognition classifier for a larger number of classes is more useful and challenging. In this work images belonging to ten different classes within the dataset were used, which is a moderate number of classes. The problem of clustering based classi-

TABLE I
Comparative classification accuracies (mean $\pm$ standard deviation) (%) (a) Training data set and (b) Test data set.

(A)

	Unsupervised clustering	Semi-supervised clustering
GLCM with Imaginary part of Gabor filter	90.3 ± 2.7	93.5 ± 2.4
GLCM with Real part of Gabor filter	90.7 ± 2.8	92.2 ± 2.5
GLCM with both real and imaginary parts of Gabor filter	90.0 ± 3.7	93.5 ± 3.2
GLCM with LoG filter	91.8 ± 2.5	93.0 ± 2.2
The proposed decision fusion scheme	93.5 ± 3.0	$\mathbf{94.5 \pm 2.4}$

(B)

	Unsupervised clustering	Semi-supervised clustering
GLCM with Imaginary part of Gabor filter	90.2 ± 2.4	92.0 ± 1.0
GLCM with Real part of Gabor filter	90.5 ± 3.7	92.3 ± 2.1
GLCM with both real and imaginary parts of Gabor filter	89.7 ± 1.9	92.3 ± 4.1
GLCM with LoG filter	90.5 ± 4.1	92.2 ± 3.1
The proposed decision fusion scheme	92.5 ± 2.1	$\mathbf{93.7 \pm 2.0}$

TABLE II

CONFUSION MATRICES BASED ON GLCM WITH IMAGINARY PART OF GABOR FILTER FOR TEST DATA SET (BASELINE 1); THE RESULTS ARE GIVEN IN PERCENTAGE % (MEAN (STD)). (A) UNSUPERVISED CLUSTERING AND (B) SEMISUPERVISED CLUSTERING.

(A)

Actual Class	Predicted Class 1	2	3	4	5	6	7	8	9	10
1	93.3 (8.6)	0	0	0	0	0	0	0	6.67 (8.6)	0
2	0	86.7 (17.2)	5.0 (8.05)	0	1.67 (5.27)	5.0 (8.05)	0	0	0	1.67 (5.27)
3	0	1.67 (5.27)	95.0 (8.05)	0	0	0	3.33 (7.03)	0	0	0
4	0	6.67 (8.6)	0	83.3 (0.0)	0	10 (8.6)	0	0	0	0
5	0	0	0	0	100 (0)	0	0	0	0	0
6	0	16.67 (0.0)	0	0	0	71.67 (8.05)	0	5 (8.05)	6.67 (8.61)	0
7	0	0	0	0	0	0	100 (0)	0	0	0
8	0	0	1.67 (5.27)	10 (8.6)	5 (8.05)	0	0	83.3 (0)	0	0
9	0	0	0	0	3.33 (7.02)	0	0	0	96.67 (7.02)	0
10	0	0	0	0	0	0	1.67 (5.27)	0	0	98.3 (5.27)

(B)

Actual Class	Predicted Class 1	2	3	4	5	6	7	8	9	10
1	100 (0)	0	0	0	0	0	0	0	0	0
2	0	90 (8.6)	0	0	0	3.33 (7.03)	0	0	6.67 (8.6)	
3	0	0	96.67 (7.03)	0	3.33 (7.03)	0	0	0	0	0
4	0	13.33 (7.03)	0	81.67 (12.30)	1.67 (5.27)	0	0	0	3.33 (7.03)	0
5	0	0	0	0	96.67 (7.03)	0	3.33 (7.03)	0	0	0
6	0	0	0	0	0	83.33 (0)	0	16.67 (0)	0	0
7	0	0	0	11.67 (8.05)	0	0	88.33 (8.05)	0	0	0
8	0	0	0	0	5 (8.05)	0	0	95 (8.05)	0	0
9	0	0	0	13.33 (7.03)	0	0	0	0	86.67 (7.03)	0
10	0	0	0	0	0	0	0	0	0	100 (0)

TABLE III

CONFUSION MATRICES BASED ON GLCM WITH REAL PART OF GABOR FILTER FOR TEST DATA SET (BASELINE 2); THE RESULTS ARE GIVEN IN PERCENTAGE % (MEAN (STD)). (A) UNSUPERVISED CLUSTERING AND (B) SEMISUPERVISED CLUSTERING.

(A)

Actual Class	Predicted Class 1	2	3	4	5	6	7	8	9	10
1	91.67 (14.16)	0	0	0	5 (8.05)	0	0	3.33 (7.03)	0	0
2	0	96.67 (7.03)	0	0	0	3.33 (7.03)	0	0	0	0
3	0	0	98.33 (5.27)	0	0	1.67 (5.27)	0	0	0	0
4	0	0	0	93.33 (8.61)	0	0	0	6.67 (8.61)	0	0
5	0	0	13.33 (7.03)	0	86.67 (7.03)	0	0	0	0	0
6	0	1.67 (5.27)	0	0	0	98.33 (5.27)	0	0	0	0
7	0	0	11.67 (8.05)	0	3.33 (7.03)	0	85 (5.27)	0	0	0
8	0	0	8.33 (8.78)	0	0	6.67 (8.61)	0	85 (5.27)	0	0
9	0	0	1.67 (5.27)	0	0	0	0	0	98.33 (5.27)	0
10	0	0	0	0	0	0	0	0	0	100 (0)

(B)

Actual Class	Predicted Class 1	2	3	4	5	6	7	8	9	10
1	100 (0)	0	0	0	0	0	0	0	0	0
2	0	88.33 (15.81)	0	5 (8.05)	0	5 (8.05)	0	1.67 (5.27)	0	0
3	0	0	86.67 (7.03)	0	0	11.67 (8.05)	0	0	0	1.67 (5.27)
4	0	0	0	85 (5.27)	13.33 (7.03)	1.67 (5.27)	0	0	0	0
5	0	0	0	0	93.33 (8.61)	0	0	6.67 (8.61)	0	0
6	0	1.67 (5.27)	11.67 (8.05)	0	0	86.67 (7.02)	0	0	0	0
7	0	0	0	0	3.33 (7.03)	0	96.67 (7.03)	0	0	0
8	0	0	0	0	8.33 (8.78)	0	0	91.67 (8.78)	0	0
9	0	0	5 (8.05)	0	0	0	0	0	95 (8.05)	0
10	0	0	0	0	0	0	0	0	0	100 (0)

fication for a larger number of classes will be our future work. Note that these known class labels are *neither used* in the training of baseline classifiers *nor* the decision fusion, *but only* for the test of the proposed approaches. In terms of the content of the images, most of the images are based on plant species found in East Asia. We will validate the proposed semi-supervised algorithms, in which the features of standard leave images are used for initialization, by comparing unsupervised clustering based on random initialisation for each of the baseline classifiers. Specifically for each of these ten classes, eleven images are used plus one standard leave image in experiments. The training data set is composed of the one standard leave image and five randomly selected images per class, and the remaining six images are used as testing data set to test the generalization performance. The process was repeated to obtain ten realisations of training/test data sets. The four baseline classifiers are trained using the Algorithm1 over the training data sets to obtain the classification results for both training data set and corresponding test data set, these are then averaged over the ten random realisations. Similarly the classification results via decision fusion are obtained and averaged over the ten random realisations. The performance of the proposed model combination classifier using max-cosine value amongst the baseline classifiers is validated for both unsupervised and semi-supervised schemes.

The results of the proposed leave image recognition framework are summarised in Table I. Since the four baseline classifiers have comparable performances, this shows that the real part of Gabor, or both real and imaginary, and LoG can also be used as discriminative features for plant leave classification as proposed in [11], but it is interesting to note that by using both real and imaginary features may not produce a superior baseline classifier.

For Table I it can be clearly seen that the proposed semi-supervised clustering approaches consistently outperform the unsupervised clustering due to the use of the one labelled image for each baseline classifier. Furthermore the idea of combining baseline learning using max-cosine value can significantly improves both training and test performance for unsupervised methods, and further improvement over the semi-supervised methods. The high classification performance has shown that it is promising to develop unsupervised methods for

TABLE IV

CONFUSION MATRICES BASED ON GLCM WITH BOTH REAL AND IMAGINARY PARTS OF GABOR FILTER FOR TEST DATA SET (BASELINE 3). THE RESULTS ARE GIVEN IN PERCENTAGE % (MEAN (STD)); (A) UNSUPERVISED CLUSTERING AND (B)SEMISUPERVISED CLUSTERING.

(A)

Actual Class	Predicted Class 1	2	3	4	5	6	7	8	9	10
1	95 (8.05)	0	0	0	0	0	5 (8.05)	0	0	0
2	0	95 (8.05)	0	0	0	3.33 (7.03)	0	0	1.67 (5.27)	0
3	0	0	88.33 (8.05)	6.67 (8.61)	0	1.67 (5.27)	1.67 (5.27)	1.67 (5.27)	0	0
4	0	0	13.33 (7.03)	83.33 (0)	0	3.33 (7.03)	0	0	0	0
5	0	3.33 (7.02)	0	0	96.67 (7.02)	0	0	0	0	0
6	0	8.33 (8.78)	0	0	0	80 (15.32)	1.67 (5.27)	0	10 (8.61)	0
7	0	0	0	8.33 (8.78)	0	0	91.67 (8.78)	0	0	0
8	0	3.33 (7.03)	18.3 (5.27)	0	0	0	0	78.33 (8.05)	0	0
9	0	0	3.33 (7.03)	0	1.67 (5.27)	5 (8.05)	0	0	90 (11.65)	0
10	0	0	0	0	0	0	0	0	0	100 (0)

(B)

Actual Class	Predicted Class 1	2	3	4	5	6	7	8	9	10
1	88.33 (11.24)	0	3.33 (7.03)	0	8.33 (8.78)	0	0	0	0	0
2	0	98.33 (5.27)	0	0	1.67 (5.27)	0	0	0	0	0
3	0	0	95 (8.05)	0	1.67 (5.27)	1.67 (5.27)	0	0	0	1.67 (5.27)
4	0	6.67 (8.61)	0	90 (8.61)	1.67 (5.27)	0	1.67 (5.27)	0	0	0
5	0	0	0	0	96.67 (7.03)	3.33 (7.03)	0	0	0	0
6	0	3.33 (7.03)	1.67 (5.27)	0	0	80 (10.54)	15 (5.27)	0	0	0
7	0	0	1.67 (5.27)	1.67 (5.27)	1.67 (5.27)	0	95 (8.05)	0	0	0
8	0	0	1.67 (5.27)	0	10 (8.61)	0	0	88.33 (8.05)	0	0
9	0	0	0	10 (8.61)	0	0	0	0	90 (8.61)	0
10	0	0	0	0	0	0	0	0	0	100 (0)

TABLE V

CONFUSION MATRICES BASED ON GLCM WITH LAPLACIAN OF GAUSSIAN FILTER FOR TEST DATA SET (BASELINE 4); THE RESULTS ARE GIVEN IN PERCENTAGE % (MEAN (STD)); (A) UNSUPERVISED CLUSTERING AND (B)SEMISUPERVISED CLUSTERING.

(A)

Actual Class	Predicted Class 1	2	3	4	5	6	7	8	9	10
1	100 (0)	0	0	0	0	0	0	0	0	0
2	0	98.33 (5.27)	0	0	0	1.67 (5.27)	0	0	0	0
3	0	0	76.67 (16.1)	0	11.67 (8.05)	11.67 (8.05)	0	0	0	0
4	0	0	0	95 (8.05)	0	0	5 (8.05)	0	0	0
5	0	0	0	0	90 (8.61)	10 (8.61)	0	0	0	0
6	0	6.67 (8.61)	0	0	0	91.67 (11.79)	0	1.67 (5.27)	0	0
7	0	0	10 (8.61)	0	11.67 (8.05)	0	75 (16.19)	3.33 (7.03)	0	0
8	0	0	0	0	11.67 (8.05)	6.67 (8.61)	0	81.67 (14.59)	0	0
9	0	1.67 (5.27)	3.33 (7.03)	0	0	0	0	0	95 (8.05)	0
10	0	0	0	0	0	0	1.67	0	0	100 (0)

(B)

Actual Class	Predicted Class 1	2	3	4	5	6	7	8	9	10
1	100 (0)	0	0	0	0	0	0	0	0	0
2	0	95 (11.25)	0	0	0	3.33 (7.03)	0	0	1.67 (5.27)	0
3	0	0	85 (14.59)	0	5 (8.05)	10 (8.61)	0	0	0	0
4	0	0	0	93.33 (8.61)	0	0	0	6.67 (8.61)	0	0
5	0	0	0	0	88.33 (13.72)	3.33 (7.03)	8.33 (8.78)	0	0	0
6	0	0	0	0	0	93.33 (8.61)	0	6.67 (8.61)	0	0
7	0	0	0	0	1.67 (5.27)	0	98.33 (5.27)	0	0	0
8	0	11.67 (8.05)	5 (8.05)	0	0	0	0	83.33 (13.61)	0	0
9	0 0	6.67 (8.61)	10 (8.61)	0	0	0	0	0	83.33 (15.71)	0
10	0	0	0	0	0	0	0	0	0	100 (0)

identifying leave images. The confusion matrices are presented in Table II- Table V for the test data sets, which provide more detailed information on the performance for each class. It can be seen that the classification performance are not balanced, indicating the data dependent nature of the clustering algorithm since there is no clear classification boundary which could be adjusted as in a supervised model. This imbalance is actually understandable as it indicates that the clusters of these images in feature space are not uniformly distributed, and there is no reason they should be. The average of the classification rates of all classes is consistent with results in Table I, with Table V(b) as the best.

IV. CONCLUSIONS

In this paper, we have introduced a novel decision fusion framework by combining semi-supervised clustering classifiers and image features analysis. Automatic plant leave recognition using digital images and unsupervised machine learning techniques is an important task since leave datasets are largely unlabelled, limiting the use of supervised machine learning techniques. In this work, based on comparative studies and evaluation based on real leave data sets, we demonstrate that semi-supervised clustering approaches are well suited for automatic leave type recognition using texture-based features including Gabor and Laplacian of Gaussian filters as inputs of classifiers, assisted by a minimum number of labelled leaves. Initially the proposed framework builds the baseline classifiers in which a number of texture-based features including Gabor and Laplacian of Gaussian filters are applied followed by the Grey Level Co-occurrence Matrix to generate leave image features. Then an on-line spherical k-means clustering technique, guided by a minimum number of labelled leaves, is used. The final decision of classification is produced based on the classifier which produces the max-cosine value amongst the baseline classifiers. Human and animal learning is largely unsupervised since people discover the structure of the world by observing it, not by being told the name of every object. Our future works will investigate other unsupervised schemes with appropriate computer vision features and other decision fusion mechanism for more accurate automatic leave type recognition.

TABLE VI

CONFUSION MATRICES BASED ON THE PROPOSED ALGORITHM 2 FOR TEST DATA SET; THE RESULTS ARE GIVEN IN PERCENTAGE % (MEAN (STD)).(A) UNSUPERVISED CLUSTERING AND (B) SEMISUPERVISED CLUSTERING.

(A)

Actual Class	Predicted Class 1	2	3	4	5	6	7	8	9	10
1	100 (0)	0	0	0	0	0	0	0	0	0
2	0	93.33 (8.61)	0	5 (8.05)	0	1.67 (5.27)	0	0	0	0
3	0	0	93.33 (8.6)	0	0	6.67 (8.61)	0	0	0	0
4	0	0	0	98.33 (5.27)	0	1.67 (5.27)	0	0	0	0
5	0	0	1.67 (5.27)	0	96.67 (7.03)	0	1.67 (5.27)	0	0	0
6	0	0	0	1.67 (5.27)	0	90 (8.6)	0	8.33 (8.78)	0	0
7	0	0	1.67 (5.27)	0	16.67 (0)	0	81.67 (5.27)	0	0	0
8	0	5 (8.05)	0	5 (8.05)	0	0	0	90 (8.61)	0	0
9	0	0	6.67 (8.61)	0	0	10 (8.61)	0	0	83.33 (7.87)	0
10	0	0	0	0	0	0	0	0	0	100 (0)

(B)

Actual Class	Predicted Class 1	2	3	4	5	6	7	8	9	10
1	91.67 (14.1)	0	0	0	5 (8.05)	0	0	0	3.33 (7.03)	0
2	0	96.67 (7.03)	0	0	0	3.33 (7.03)	0	0	0	0
3	0	0	98.33 (5.27)	0	0	1.67 (5.27)	0	0	0	0
4	0	0	0	93.33 (8.61)	0	0	0	6.67 (8.61)	0	0
5	0	0	13.33 (7.03)	0	86.67 (7.03)	0	0	0	0	0
6	0	1.67 (5.27)	0	0	0	98.33 (5.27)	0	0	0	0
7	0	0	1.67 (8.05)	0	16.67 (7.03)	0	85 (5.27)	0	0	0
8	0	0	8.33 (8.78)	0	0	6.67 (8.60)	0	85 (5.27)	0	0
9	0	0	1.67 (5.27)	0	0	0	0	0	98.33 (5.27)	0
10	0	0	0	0	0	0	0	0	0	100 (0)

REFERENCES

[1] T. Weier, C. Stocking, and M. Barbour, *Botany: An introduction to plant biology,* John Wiley and Sons, 1974.

[2] Q. Wu, C. Zhou, and C. Wang, "Feature extraction and automatic recognition of plant leaf using artificial neural network," Advances in Artificial Intelligence, vol. 20, no. 3, pp. 3 – 10, 2006.

[3] Stuart J. Russell, Peter Norvig (2010) *Artificial Intelligence: A Modern Approach*, Third Edition, Prentice Hall.

[4] D. Guyer, G. Miles, and M. Schreiber,"Computer vision and image processing for plant identification," Microfiche collection, 1984.

[5] I. Yahiaoui, N. Hervé, and N. Boujemaa,"Shape-based image retrieval in botanical collections," In Pacific-Rim Conference on Multimedia, pp. 357-364, Nov. 2006.

[6] C. Zhao, S. S. F. Chan, W-K Cham, L.M. Chu: "Plant identification using leaf shapes -A pattern counting approach", Pattern Recognition, Vol. 48, pp3203-3215, 2015.

[7] J. Han and L. Guo, "A Shape-based Image Retrieval Method Using Salient edges," in Signal Processing: Image Communication, vol. 18, pp. 141-156, 2003.

[8] C. Im, H. Nishida, and T. Kunii, "Recognizing plant species by leaf shapes-a case study of the acer family," In Proceedings of 14th International Conference on Pattern Recognition, vol. 2, pp. 1171-1173, 1998.

[9] Z. Wang, Z. Chi, D. Feng, and Q. Wang,"Leaf image retrieval with shape features," In International Conference on Advances in Visual Information Systems, pp. 477-487, Nov. 2000.

[10] R. Haralick,"Statistical and structural approaches to texture," In Proceedings of IEEE, vol. 67, pp. 786 – 804, 1979.

[11] J. Chaki, R. Parekh, S. Bhattacharya, "Plant leaf recognition using texture and shape features with neural classifiers," In Pattern Recognition Letters, vol. 58, pp. 61-68, 2015.

[12] M. Hu, "Visual Pattern Recognition by Moment Invariants," In IEEE Transactions on Information Theory, vol. 8, no. 2, pp. 179-187, 1962.

[13] L. Hickey, A. Ash, B. Ellis, P. Wilf, K. Johnson, and S. Wing, "Manual of leaf architecture-morphological description and categorization of dicotyledonous and net-veined monocotyledonous angiosperms by Leaf Architecture Working Group," Smithsonian Institution. Washington, DC, pp. 26-27, 1999.

[14] H. Fu, Z. Chi, J. Chang, and C. Fu,"Extraction of leaf vein features based on artificial neural network – Studies on the living plant identification," In Chinese Bulletin of Botany, vol. 21, no. 4, pp. 429-436, 2003.

[15] A. Caglayan, O. Guclu, and A. Can, "A plant recognition approach using shape and colour features in leaf images," In International Conference on Image Analysis and Processing, pp. 161-170, Sept. 2013.

[16] F. Zhang and H. Zhang, "Content Based Image Retrieval of Standard Tobacco Leaf Database," In Computer Engineering and Applications, vol. 7, p. 67, 2002.

[17] O. Mzoughi, I. Yahiaoui, N. Boujemaa, E. Zagrouba,"Advanced tree species identification using multiple leaf parts image queries," In International Conference on Image Processing (ICIP2013), pp. 3967-3971, Sept. 2013.

[18] S. Wu, F. Bao, E. Xu, Y. Wang, Y. Chang, and Q. Xiang,"A leaf recognition algorithm for plant classification using probabilistic neural network," In IEEE International Symposium on Signal Processing and Information Technology, pp. 11-16, Dec. 2007.

[19] P. Belhumeur, D. Chen, S. Feiner, D. Jacobs, W. Kress, H. Ling, and L. Zhang, "Searching the world's herbaria: A system for visual identification of plant species," In European Conference on Computer Vision, pp. 116-129, Oct. 2008.

[20] J. Du, X. Wang, and G. Zhang, "Leaf shape based plant species recognition," Applied Mathematics and Computation, vol. 185, pp. 883 – 893, 2007.

[21] S. Hang, A. Tatsuma, and M. Aono,"Bluefield (KDE TUT) at LifeCLEF 2016 plant identification task." Working notes of CLEF 2016 conference, 2016.

[22] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for LargeScale Image Recognition," in CVPR, Sept. 2014.

[23] S. H. Lee, C. S. Chan, S. J. Mayo and P. Remagnino: "How deep learning extracts and learns leaf features for plant classification", Pattern Recognition, Vol. 71, pp1-13, 2017.

[24] R. Hu, W. Jia, and D. Huang, "Multiscale distance matrix for fast plant leaf recognition," IEEE transactions on image processing, vol. 21, no. 11, pp. 4667-4672, 2012.

[25] Bair, E. Semi-supervised clustering methods. Wiley Interdisciplinary Reviews: Computational Statistics, 5(5), 349-361, 2013.

[26] S. Haykin, *Neural Networks and Learning Machines*. Pearson Education Inc, 2009.

[27] S. Zhong: "Efficient online spherical k-means clustering". In Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on (Vol. 5, pp. 3180-3185), 2005.

[28] Flavia Dataset, http://flavia.sourceforge.net/., last accessed 2019/01/07.

[29] A. K. Jain and F. Farrokhnia: "Unsupervised texture segmentation using Gabor filters," Pattern Recogn. 24, 12 (December 1991), pp1167-1186.

[30] Hugo Aguirre-Ramos, J. G. Avina-Cervantes, I. Cruz-Aceves, José Ruiz-Pinales, S. Ledesma:"Blood vessel segmentation in retinal fundus images using Gabor filters, fractional derivatives, and Expectation Maximization", Applied Mathematics and Computation, Volume 339, 15 December 2018, pp568-587.

[31] M. N. Sumaiya, R. Shantha Selva Kumari: "Gabor filter based change detection in SAR images by KI thresholding", Optik, Volume 130, February 2017, pp114-122.

[32] E. R. Davies: "Introduction to texture analysis", in Handbook of Texture Analysis, edited by M. Mirmehdi, et al, Imperial College Press, pp. 9-13, 2008.

[33] D. Marr and E. Hildreth, "Theory of Edge Detection," Proceedings of the Royal Society of London. Series B, Biological Sciences, 207 (1167): 187–217, 1980.

[34] H. Kong, H. C. Akakin and S. E. Sarma, "A Generalized Laplacian of Gaussian Filter for Blob Detection and Its Applications," in IEEE Transactions on Cybernetics, vol. 43, no. 6, pp. 1719-1733, Dec. 2013.

[35] Y. LeCun, Y. Bengio and G. Hinton,"Deep Learning", Nature, 521, pp436–444, 2015.