Unsupervised Learning for Image Classification based on Distribution of Hierarchical Feature Tree

Thach-Thao Duong(1), Joo-Hwee Lim(2)
(1)Faculty of Information Technology
Ho Chi Minh University of Science
Ho Chi Minh City, Vietnam

Hai-Quan Vu(1), Jean-Pierre Chevallet(2)
(2)French-Singapore IPAL Joint Lab
Institute for Infocomm Research
Singapore

Abstract—**The classification image into one of several categories is a problem arisen naturally under a wide range of circumstances. In this paper, we present a novel unsupervised model for the image classification based on feature's distribution of particular patches of images. Our method firstly divides an image into grids and then constructs a hierarchical tree in order to mine the feature information of the image details. According to our definition, the root of the tree contains the global information of the image, and the child nodes contain detail information of image. We observe the distribution of features on the tree to find out which patches are important in term of a particular class. The experiment results show that our performances are competitive with the state of art in image classification in term of recognition rate.**

***Keywords-** image classification, hierarchical tree, unsupervised learning, distribution*

I. INTRODUCTION

Advance in multimedia technologies such as image digitization, storage and transmission along with the growth of the World Wide Web, mobile device, cameras have lead to the proliferation of online digital images. Content-based image classification has been an interesting subject of many researchers in recent years. There are many great efforts in developing the classification approaches and techniques to improve the classification accuracy.

Recently, many advanced classification approaches, such as artificial neural networks, fuzzy-sets, and expert systems, have been widely applied for the problem of image classification. In general, image classification can be classified into two major approaches: supervised and unsupervised. In this paper, we introduce an unsupervised learning method for this problem.

There are some classification methods [1, 2, 3, 16, 17, 18, 19] based on segmentation of image into regions which are considered as objects of image. They then extract the features from the objects such as size, shape, texture, etc to do classification. This kind of method is based on objects and the features extracted from them. However, these methods depend mainly on the quality of segmentation and obviously, the segmentation issue is a difficult and a complex problem in computer vision. In fact, a general universal algorithm for segmenting images certainly does not exist [4]. Our method does not rely on segmentation so that it could reduce inefficiency caused by wrong-segmented regions.

Images in the same classes usually have similar features at some locations. For example, images of a beach usually have the blue color in the top part of it or the image of the building usually has line structure in the middle. We divide images into a number of child-images and construct a hierarchical tree to store the information. The tree root is the original global image and other nodes are the child-images of parent image. The root and the child-nodes of tree express the global view of the whole image and the detail views respectively. Features of nodes are extracted to construct the feature tree. Each node of feature tree stores features at a particular location in the image. The hierarchical feature tree structure takes advantages in storing the global and local information of images as well as the spatial relation between patches of images.

At a particular location on the image, there are many sub-images of training images looks alike. This means that the distribution of feature at that location is dense, the deviation is small, and that location is important in classifying. Therefore, we observe the distribution of the features at each node through the deviation of features to compute the weight factor tree. This tree expresses the importance of nodes in tree. Weight factor tree expresses the importance of patches or locations within an image. This tree is the determinant in classifying an image to a class or a category.

II. PROPOSED ALGORITHMS

A. Image Representation

Some methods represent image as a collection of objects. Objects are considered as segmented regions. Links can be added between objects. Image representation becomes graph of objects. This image representation is close to the way presented by human but it is still a challenge in computer vision. The main difficulty is that the efficiency of the application depends too much on the segmentation quality, which is not very good so far. The second difficulty is complexity of graph algorithms.

We use a frame to split image into sub-images, which is a child-images of original image. If we use a frame of 2x2, we have a quad-tree representation [20]. An original image is split into child-images, and then these child-images are recursively split with the same way. This process repeats until the tree's height reaches a specific threshold or the size of leaf image reaches a specific threshold. As the result, we can construct a tree structure of the sub-images. The root is the original image; the child-nodes are the sub-images. Therefore, the root is the global view of image and the child-nodes are the detail views of image. At each node, features are extracted to construct a feature-tree. Feature-tree contains the features of the global view of image and the detail views of image. It is the important data tree which takes part into the learning process and the classification process.

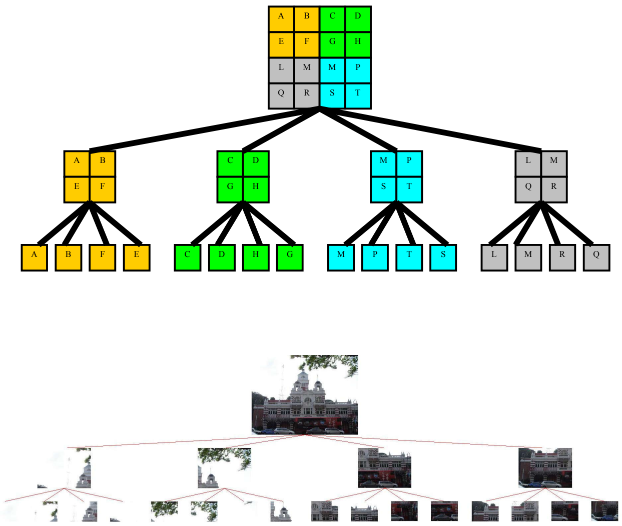

Figure 1: Image representation of sub-image tree

We define a feature-tree recursively. A feature-tree is denoted by $FT = \{N^F_{root}\}$ which has the root node N^F_{root}. A node at location S in feature tree $N^F_S = \{F_S; N^F_{S,l} \mid l = 1,..., L\}$ has itself data F_S and a list of child nodes. N^l_S is the $l-th$ child-node. L is the number of children. F_S is the set of features $F_S = \{f^k_S \mid i = 1,..., K\}$, K is the number of features extracted. f^k_S is the k-th feature

B. *Learning process*

The target of the classification is to classify the input image into a class and the target of the learning is to compute the specific characteristics of classes. Therefore, we have to define what the class representative is or what can characterize for the class. According to the class representative, we can define an appropriate method to match the input image and the class representatives.

Our idea is to emphasize the important nodes of tree structure when classifying image into classes. The node is important when there are many similar sub-images at that node location in the training set of a particular class. The similarity between sub-images is computed based on the distances between feature values. The more similar between sub-images is, the less distance between their features is. Therefore, the more important a node is, the denser the feature distribution is.

The notion "distribution of feature" is mentioned in [7] and has good performances. Thus, we compute mean features and a standard deviation to characterize the density of the distribution at each node of the tree.

1) Mean feature tree (MFT)

The mean feature tree of class C is the $FT = \{\overline{N^F_{root}}\}$, whose root node is $\overline{N^F_{root}}$. At the node location S in tree structure, all training images of class C have feature sets at the node location are $\{F_{S,j}; j = 1..N\}$ where N is number of training images of class C.

Feature set contains many features $F_{S,j} = \{f^k_{S,j} \mid k = 1..K\}$ where K is number of features.

The mean feature at the node location S in tree structure is

$$\overline{N^F_S} = \{\overline{F_S}, \overline{N^F_{S,l}} \mid j = 1..N; l = 1..L\}$$

$$\overline{F_S} = \{\overline{f^k_S} \mid k = 1..K\}$$

$$\overline{f^k_S} = \underset{j=1\to N}{avg}\left(f^k_{S,j}\right)$$

Where $\underset{j=1\to N}{avg}\left(f^k_{S,j}\right)$ is the average of N feature value $f^k_{S,j}$; j=1..N. If type of feature is just a scalar value

$$\underset{j=1\to N}{avg}\left(f^k_{S,j}\right) = \frac{1}{N}\sum_{j=1}^{N} f^k_{S,j}$$

If type of feature is vector or histogram in the other word, $f^k_{S,j} = \{h^{k,m}_{S,j} \mid m = 1..M\}$ where M is the number of bins

$$\overline{f^k_S} = \underset{j=1\to N}{avg}\left(f^k_{C,S,j}\right) = \{\overline{h^{k,m}_{C,S,j}} \mid m = 1..M\}$$

$$\overline{h^{k,m}_{C,S,j}} = \underset{m=1\to M}{avg}\left(h^{k,m}_{C,S,j}\right) = \frac{1}{N}\sum_{k=1}^{N} h^{k,m}_{C,S,j}$$

It means that we compute the average on each bin value to construct the mean histogram.

2) Standard deviation tree (SDT)

The standard deviation of a distribution is the value to measure the spread of distribution. We have to compute the standard deviation at each node on the tree structure to characterize the spread of feature distributions. Each node contains a set of features. We can consider the distributions of each separate feature or the distribution of the whole set of features.

If we consider the distributions of separate features, we call it partial distribution. At each node of SDT, there are K standard deviation values of K features. The notation of global standard deviation tree is $SDg = \{N^{SDg}_{root}\}$. A node at location S in standard deviation tree is $N^{SDg}_S = \{\sigma^k_S, N_{S,l} \mid k = 1,..., K; l = 1..L\}$, where K is the number of features, L is the number of children. σ^k_S is the standard deviation of the k-th feature at the node location S.

If we consider the distribution of the whole set of features, we call it global distribution. At each node of SDT, there is only one standard deviation value. The notation of partial standard deviation C is $SDf = \{N_{root_C}^{SDf}\}$. A node at location S in standard deviation tree is $N_S^{SDf} = \{\sigma_S; N_{S,l} \mid l = 1..L\}$, where L is the number of children.

Distance between two set of feature values with the same format R and S is defined as Equation (1)

$$D(R,Q) = \frac{1}{K}\sum_{k=1}^{K} D\left(f^k(R), f^k(Q)\right) \qquad (1)$$

Equation (1) means that the distance between two sets of features is just the average the distances between individual features. If the kind of feature is scalar, the distance will be the absolute subtraction of two features. If the kind of feature is vector or histogram, we can use the histogram distance

3) Weight factor tree (WT)

In this approach, we construct two kinds of tree: mean feature tree and weight tree. Mean feature tree has the same structure of the feature tree of all training images. We compute the average feature values of all the training images at the same location of the tree. Mean feature tree is considered as the class representative. Each input image will be compute distance to mean feature tree. The smaller the distance is, the higher the probability that input image belongs to the class is. Beside the mean feature value at each node, we also have the standard deviation of the feature distribution. The denser distribution is, the more important node is. The node is important in tree structure when the training sub-images at that node are similar. That means the distances between feature values at that node are small, and the distribution of features at that node is dense. Therefore, the standard deviation of that node is small in that case.

The smaller standard deviation value is, the larger weight factor is. Therefore, the weight is the reverse proportional to the standard deviation. The class representatives are mean feature tree and the weight tree. We can notice which part of the images is important by the weight factor. The important parts will attach much attention in matching new input image to class.

We propose two ways to compute the weight factor. The first one, we compute the weight as the inverse of the standard deviation

$$w = \frac{1}{\sigma} \qquad (2)$$

The second one, we traversal the standard deviation tree and get the minimum and maximum. The weight is the inverse of the standard deviation but it is scaled in range $[\varepsilon, E]$ where small positive value ε is larger then zero. To normalize the weight factor, we use E = 1.0 ($\varepsilon = 0.1$ or 0.01). With this way, we rearrange the standard deviation's inverse to fit with the range of minimum and maximum. It means that, we rearrange the weight value on WT tree to suitable with the distribution of standard value on STD.

$$\frac{E - w}{E - \varepsilon} = \frac{\sigma - \min}{\max - \min} \qquad (3)$$

Therefore, the formulation to compute the weight is

$$w = E - (E - \varepsilon)\frac{\sigma - \min}{\max - \min} \qquad (4)$$

If we consider the distribution of each separate feature, there are K standard deviation values of K features at each node of SDT. The partial weight tree is $WpT_C = \{N_{root}^{Wp}\}$. A node at location S in weight tree $N_S^{Wp} = \{w_S^k, N_{S,l}^{Wp} \mid k = 1,...,K; l = 1..L\}$, where K is the number of features, L is the number of children, w_S^k is the weight of the k-th feature at the node location S.

If we consider the distribution of the whole feature set, at node of WT we will have only one weight value. The global weight tree of class C is $WgT_C = \{N_{root_C}^{Wg}\}$. The node location S in weight tree $N_S^{Wp} = \{w_S; N_{S,l}^{Wp} \mid l = 1..L\}$; L is the number of children

C. Classification

The input image I is split into sub-images. A tree of sub-images is constructed and extracted features to build a feature-tree $FT(I) = \{N_{root}^{I}(I)\}$. We denote D(.) is the distance function. If we use the global weight tree, distance between image I and class C is $D_g(I,C)$, which is compute recursively

$$D_g(I,C) = D_g\left(N_{root}^F(I), \overline{N_{root}^F}(C), N_{root}^{Wg}(C)\right) \qquad (5)$$

At the node location

$$D_g\left(N_S^F(I), \overline{N_S^F}(C), N_S^{Wg}(C)\right) = D\left(F_S(I), \overline{F_S}(C)\right) * w_S + \sum_{l=1}^{L} D_g\left(N_{S,l}^F(I), \overline{N_{S,l}^F}(C), N_{S,l}^{Wg}(C)\right) \qquad (6)$$

Equation (6) means that we summarize the distance of nodes in feature tree of input image to nodes in the mean feature tree of class with the weight values in weight factor tree. If the weight in a node is large, it will make distance value larger. The distance at that node does not effect too much on the distance value. In the other word, if the weight is large, it will decrease the possibility an image belongs to class.

With the partial weight tree, distance between image I and class C is $D_p(I,C)$

$$D_p(I,C) = \frac{1}{K} D_p^k\left(N_{root}^F(I), \overline{N_{root}^F}(C), N_{root}^{Wp}(C)\right) \qquad (7)$$

At the node location

$$D_p^k\left(N_S^F(I), \overline{N_S^F}(C), N_S^{Wp}(C)\right) = D\left(f_S^k(I), \overline{f_S^k}(C)\right) * w_S^k + \sum_{l=1}^{L} D_p^k\left(N_{S,l}^F(I), \overline{N_{S,l}^F}(C), N_{S,l}^{Wp}(C)\right) \qquad (8)$$

Likewise with the global distance but we compute the distance on each separate kind of features and then get the average value. The idea is the same with the previous formulation but we just observe each kind of features and because we have the separate weight values, therefore we have to compute the separate distance and get the average for the final distance between image I and class C. The smaller distance between image and class, the higher probability the image belongs to that class.

To classify an image I to one of M classes C_i , i = 1..M. We compute the distance $D(I,C_i)$ between image I and class C_i . The class whose the distance $D(I,C_i)$ is smallest is the rightful class of image I.

III. EXPERIMENTS

We use the color histogram HSL and Edge orientation histogram [9] for the color and shape feature. HSV histogram is extracted with 64 bins for each channel Hue, Saturation, Value. Edge is detected using Canny operator, quantized into 80 bins of 4.5^0 . HSL and Edge Orientation Histogram are naturally invariant to translations and rotations in the image.

We test on color databases which are STOIC90[10], WANG[15], and ZuBuD[8]. ZuBuD is the database of 201 buildings at Zurich. Each building is trained by 5 images at 5 arbitrary viewpoints. The test set contains 115 images. The performances measured in [12, 13, 14] are respectively 93%, 89.6%, and 59.13% of classification recognition rate. STOIC90 is the Singapore Tourist Object Identification Collection for scenes in Singapore. All the images are resized 320x240. Training set is 90 class and 5 training images for each class. Test set is 110 images. The STOIC is much more varied in content in comparing with ZuBuD, which only contains close-up of buildings. The object in the STOIC database can be smaller due to their physical sizes (e.g. sculptures) or acquisition distance (e.g. far view), or different viewing angles. The performance in [10] is ranging from 76% to 92%. WANG is the subset of the Corel database. It contains 10 classes of 100 images. This is a major advantage of this database because due to the given classification it is possible to evaluate retrieval performances. The images are of size 384×256 or 256×384 pixels. This database was created by the group of professor Wang from the Pennsylvania State University and is available for download [15]

TABLE I. CLASSIFICATION RATE ON STOIC

	Histogram distance	Eq. (2)	Eq. (4)
Global distribution	χ^2	66.36%	60.90%
	Fidelity	81.81%	80.90%
	Minkowski	73.63%	74.54%
	Relative Deviation	80.90%	83.63%
Partial distribution	χ^2	64.54%	66.36%
	Fidelity	66.36%	65.54%
	Minkowski	59.09%	59.09%
	Relative Deviation	74.45%	74.54%

Our performances use 4 types of histogram distance: χ^2 Distance[7], Fidelity Based Distance[6], Minkowski Distances[7] with p=2, Relative Deviation Distance[7]. We choose these because they are symmetric.

Table I shows that the Fidelity distance always gives the good performances comparing with the three others. The global distribution has larger recognition rate than the partial distribution. It is similar to the way human being looking image. People usually consider image similarity based on set of features rather than the separate features such as color and shape.

TABLE II. CLASSIFICATION RATE ON ZUBUD,STOIC, WANG USING FIDELITY HISTOGRAM DISTANCE

	Database	Eq. (2)	Eq. (4)
Global distribution	ZuBuD	93.91%	93.04%
	STOIC	81.81%	80.90%
	WANG	89.5%	88.50%
Partial distribution	ZuBuD	94.78%	94.78%
	STOIC	66.36%	65.54%
	WANG	83.5%	81.50%

Our approach uses the hierarchical tree to store the characterized features of image from global view to local views. The learning method takes advantages of feature distribution on hierarchical tree to evaluate key patches of images. Table II shows that our method is efficient comparing with three databases above.

IV. CONCLUSION AND FUTURE WORK

Our proposed algorithm focuses on solving the problem in image representation by using unsupervised learning for image classification phase. Though this method is still limited as the matching feature-tree node by node, it has efficient performances for the data in the experiments. The algorithm guarantees the order in matching feature-tree or in other words, it guarantees the order in matching between parts in image. In the future, we will construct the algorithm for non-order matching of feature-tree. Our method is not very complex but it has efficient performances for the data in the experiments. It is potential to continue especially in apply non-order matching algorithm. Finally, this method is suitable for large database not only in image classification but in image annotation and image retrieval also.

REFERENCES

[1] S.R Avula, Jinshan Tang, S.T. Acton, “Image retrieval using segmentation”, Systems and Information Engineering Design Symposium , pp. 289 – 294,24-25 April 2003 IEEE

[2] Yinghua Lu, Qiushi Zhao, Jun Kong, Changhua Tang, Yanwen Li, “A Two-Stage Region-Based Image Retrieval Approach Using Combined Color and Texture Features”, Volume 4304/2006 P1010-1014. AI 2006: Advances in Artificial Intelligent.

[3] Ying Liu, Dengsheng Zhang, Guojun Lu, M. Wei-Ying, “A survey of content-based image retrieval with high-level semantics”. Pattern Recognition, Volume 40 , Issue 1, pp. 262-282 , January 2007

[4] L. Lucchese and S.K. Mitra, “Color Image Segmentation: A State-of-the-Art Survey”, Proceedings of the Indian National ScienceAcademy, 67(2):207–221 , March 2001.

[5] Stefan Wessel, Norbert Stoop, Emanuel Gull, Simon Trebst and Matthias Troyer, “Optimized broad-histogram ensembles for the simulation of quantum systems”, Journal of Statistical Mechanics : Theory and Experiments P12005 (2007).

[6] M. N¨olle, “Distribution Distance Measures Applied to 3-D Object Recognition – A Case Study”, Proc. DAGM 2003, Pattern Recognition, 25th DAGM Symposium, Vol. 2781 of Lecture Notes in Computer Science, pp. 84–91, Magdeburg, Germany, Sept. 2003. Springer Verlag.

[7] Y. Rubner, J. Puzicha, C. Tomasi, and J. M. Buhmann. “Empirical evaluation of dissimilarity measures for color and texture”, Computer Vision and Image Understanding, 84(1):25-43, October 2001

[8] ETH-Zurich. Zurich building image database. http://www.vision.ee.ethz.ch/showroom/zubud/index.en.html.

[9] A.K Jain and A. Vailaya, “Shape-Based Retrieval: A Case Study With Trademark Image Database”, Pattern Recognition, Vol. 31, No.9, page 1369-1390, 1998

[10] Joo-Hwee Lim, Yiqun Li, Yilun You, and Jean-Pierre Chevallet, “Scene Recognition with camera phones for tourist information access”, IEEE International Conference on Multimedia and Expo, pp. 100-103, 2007

[11] Rapha¨el Mar´ee, Pierre Geurts, Louis Wehenkel, “Content-based Image Retrieval by Indexing Random Subwindows with Randomized Trees” Proc. 8th Asian Conference on Computer Vision (ACCV), LNCS, Volume 4844, page 611—620, Nov 2007

[12] S. Obdrzalek and J. Matas, “Sub-linear indexing for large scale object recognition”, In Proc. British Machine Vision Conference (BMVC), pages 1–10, 2005.

[13] T. Deselaers, D. Keysers, and H. Ney, “Classification error rate for quantitative evaluation of content-based image retrieval systems”, In Proc. 17th International Conference on Pattern Recognition (ICPR), pages 505–508, 2004.

[14] H. Shao, T. Svoboda, V. Ferrari, T. Tuytelaars, and L. Van Gool, “Fast indexing for image retrieval based on local appearance with re-ranking”, In Proc. IEEE International Conference on Image Processing (ICIP), pages 737–749, 2003.

[15] James Z.Wangs ‘s Research Group. http://wang.ist.psu.edu/

[16] D. Geneletti and B.G.H. Gorte, “A method for object-oriented land cover classification combining Landsat TM data and aerial photographs”, International Journal of Remote Sensing, 2003, Vol. 24, no. 6., pp. 1273-1286

[17] C.J.Van der Sande, S.M. De Jong and A.P.J. De Roo, “A segmentation and classification approach of IKONOS-2 imagery for land cover mapping to assist flood risk and flood damage assessment”, International Journal of Applied Earth Observation and Geoinformation, 4, pp. 217–229., June 2003

[18] U.C. Benz , P. Hofmann , G. Willhauck, I. Lingenfelder, and M. Heynen, “Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GISready information”, ISPRS Journal of Photogrammetry & Remote Sensing, 58, pp. 239–258, 2004

[19] V. Walter, “Object-based classification of remote sensing data for change detection”, ISPRS Journal of Photogrammetry & Remote Sensing, 58, pp. 225–238, 2004

[20] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf, “Computational Geometry 2nd edition, Chapter 14: Quadtrees”, pp.291–306, Springer-Verlag, 2000