ISSN:0975-9646

Sukhman Kaur et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 4514-4518

Comparative Study on Content Based Image Retrieval With Various Approaches

Sukhman Kaur[#], Rajesh Sharma[*]

[#]*Computer Science, Punjab Technical University*
C.T. Engineering and Technology College Jalandhar, India

[*]*Asstt.Prof. In Computer Science*
C.T. Engineering and Technology College Jalandhar, India

***Abstract*— Content Based Image Retrieval (CBIR) has become one of the most active research areas in the past few years and is still a demanding research topic in the field of Digital Image Processing. This paper starts with the introduction about CBIR and then we have studied about various efficient methods based on Content Based Image Retrieval. In this we have studied about various approaches like SURF (Speeded up Robust Features), SVM(Support Vector Machine) and NN(Neural Network).we have study the features for Image Retrieval like color, texture and shape. We briefly examine the likeness measures based on which matches are made and images are retrieved**

Keywords:- **Matching, Image processing, Surf, Neural Network and SVM**

I. INTRODUCTION

At present, the image matching methods can be roughly divided into two classes; one is the image matching based on image matching and feature matching. Matching method is directly use the image grey value to determine the space geometry transform between the images, this method can make full use of the information of the image, so it is also known as the matching method based on integral image content; it has no feature detection steps; in the feature matching stage; the fixed size window and even whole image matching are adopted in estimation; so the calculation is simple and also easy to be performed. In recent years, very large collections of images and videos have grown rapidly. In parallel with this growth, con-tent based retrieval and querying the indexed collections are required to access visual information. Therefore two of the main components of the visual information are texture and colour. The history of the content-based image retrieval can be divided into three phases:

1. The retrieval based on artificial notes.
2. The retrieval based on vision character of image contents.
3. The retrieval based on image semantic features.

The image retrieval that is based on artificial notes labels images by using text firstly, in fact it has already changed image retrieval into traditional keywords retrieval. Problem with the approach is that, it brings heavy workload and on the other hand, it still remains subjectivity and uncertainty. Because the image retrieval that is based on artificial notes still remains insufficiency, the farther study that adapts vision image features has been come up and become the main study. The character of this method is image feature extraction impersonally; whether the retrieval is good or not depends on the accuracy of the features extraction. Therefore the research based on vision features is becoming the focus in the academic community. The feature of vision can be classified by semantic hierarchy into middle level feature and low- level feature. Low-level feature includes colour, texture and inflexion. Middle level involves shape description and object feature. Content based Image Retrieval systems try to retrieve images similar to a user-defined specification or pattern (e.g., shape sketch, image example). And their goal is to support image retrieval based on content properties (e.g., shape, colour, texture), usually encoded into feature vectors [4,5,7]. One of the main advantages of the CBIR approach is the possibility of an automatic retrieval process; instead of the traditional keyword-based approach; which usually requires very laborious and time-consuming previous annotation of database images.

II. OVERVIEW OF CBIR

As processors become increasingly powerful; and memories become increasingly cheaper; the deployment of large image databases for a variety of applications have now become realisable. And databases of art works; satellite and medical imagery have been attracting more and more users in various professional fields — for example; geography; medicine; architecture; advertising; design; fashion; and publishing. Therefore effectively and efficiently accessing desired images from large and varied image databases is now a necessity. ***CBIR*** or ***Content Based Image Retrieval*** is the retrieval of images based on visual features such as colour; texture and shape. And reasons for its development are that in many large image databases; traditional methods of image indexing have proven to be insufficient; laborious; and extremely time consuming. Therefore these old methods of image indexing; ranging from storing an image in the database and associating it with a keyword or number; to associating it with a categorized description; have become obsolete [8,9,12]. This is not ***CBIR***. In CBIR; each image that is stored in the database has its features extracted and compared to the features of the query image. This involves two steps:

1. Feature Extraction: The first step in the process is extracting image features to a distinguishable extent.

2. Matching: The second step involves matching these features to yield a result that is visually similar.

A. CBIR Systems

Several CBIR systems currently exist; and are being constantly developed. The examples are:

1. ***QBIC*** or ***Query By Image Content*** was developed by IBM; Almaden Research Centre; to allow users to graphically pose and refine queries based on multiple visual properties such as colour, texture and shape. It supports queries based on input images; user-constructed sketches; and selected colour and texture patterns .
2. ***VIR Image Engine*** by Virage Inc; like QBIC; and enables image retrieval based on primitive attributes such as colour; texture and structure. This examines the pixels in the image and performs an analysis process; deriving image characterization features .
3. ***VisualSEEK*** and ***WebSEEK*** were developed by the Department of Electrical Engineering; Columbia University. The both these systems support colour and spatial location matching as well as texture matching.
4. ***NeTra*** was developed by the Department of Electrical and Computer Engineering; University of California. This supports colour; shape; spatial layout and texture matching; as well as segmentation.
5. ***MARS*** or ***Multimedia Analysis and Retrieval System*** was developed by the Beckman Institute for Advanced Science and Technology; University of Illinois. This supports colour; spatial layout; texture and shape matching.
6. ***Viper*** or ***Visual Information Processing for Enhanced Retrieval*** was developed at the Computer Vision Group; University of Geneva. This supports colour and texture matching [7].

III. SPEEDED UP R0BUST FEATURE (SURF)

SURF (Speeded up Robust Features) is a robust local feature detector; first presented by Herbert Bay et al in 2006; that can be used in computer vision tasks like object recognition or 3D reconstruction. This is partly inspired by the SIFT descriptor. Therefore standard version of SURF is several times faster than SIFT and claimed by its authors to be more robust against different image transformations than SIFT. And SURF is based on sums of 2D Haar wavelet responses and makes an efficient use of integral images. This uses an integer approximation to the determinant of Hessian blob detector; which can be computed extremely quickly with an integral image (3 integer operations). Therefore For features; it uses the sum of the Haar wavelet response around the point of interest. These can be computed with the aid of the integral image. SURF used in this approach to extract relevant features and descriptors from images. This approach is preferred over its predecessor due to its succinct descriptor length i.e. 64 floating point values. In SURF, a descriptor vector of length 64 is constructed using a histogram of gradient orientations in the local neighborhood around each key point. Modified SURF (Speeded up Robust Features) is one of the famous feature-detection algorithms [11,17]. The panorama image stitching system which combines an image matching algorithm; modified SURF and an image blending algorithm; multi-band blending. This process is divided in the following steps: first; get feature descriptor of the image using modified SURF; secondly; find matching pairs; using correlation matrix; and remove the mismatch couples by RANSAC(Random Sample Consensus); then; adjust the images by bundle adjustment and estimate the accurate homographic matrix; lastly; blend images by Alpha blending. And comparison of SIFT (Scale Invariant Feature Transform) and Harris detector are also shown as a base of selection of image matching algorithm. And according to the experiments; the present system can make the stitching seam invisible and get a perfect panorama for large image data and it is faster than previous method. SURF approximates or even outperforms previously proposed schemes with respect to repeatability; distinctiveness; and robustness; yet can be computed and compared much faster. And this is achieved by relying on integral images for image convolutions; by building on the strengths of the leading existing detector sand descriptors specially, using a Hessian matrix-based measure for the detector; and a distribution-based descriptor and by simplifying these methods to the essential [18,20]. This leads to a combination of novel detection; description; and matching steps. It approximates or even outperforms previously proposed schemes with respect to repeatability; distinctiveness; and robustness; yet can be computed and compared much faster. And this is achieved by;

1. Relying on integral images for image convolutions
2. Building on the strengths of the leading existing detectors and descriptors (using a Hessian matrix-based measure for the detector; and a distribution based descriptor).
3. Simplifying these methods to the essential.

This leads to a combination of novel detection; description; and matching steps.

IV. NEURAL NETWORKS

Neural network is set of interconnected neurons. This is used for universal approximation. Artificial neural networks are composed of interconnecting artificial neurons (programming constructs that mimic the properties of biological neurons). And artificial neural networks may either be used to gain an understanding of biological neural networks; or for solving artificial intelligence problems without necessarily creating a model of a real biological system. Therefore real; biological nervous system is highly complex: artificial neural network algorithms attempt to abstract this complexity and focus on what may hypothetically matter most from an information processing point of view. Good performance (e.g. as measured by good predictive ability; low generalization error); or performance mimicking animal or human error patterns; can then be used as one source of evidence towards supporting the hypothesis that the abstraction really captured something important from the point of view of information processing in the brain. And another incentive for these abstractions is to reduce the amount of computation required to simulate artificial neural networks.

A. Architecture of artificial neural network

The basic architecture consists of three types of neuron layers: input; hidden; and output. And feed-forward networks; the signal flow is from input to output units; strictly in a feed-forward direction. Therefore data processing can extend over multiple layers of units; but no feedback connections are present. The recurrent networks contain feedback connections. The contrary to feed-forward networks; the dynamical properties of the network are important. In some cases; the activation values of the units undergo a relaxation process such that the network will evolve to a stable state in which these activations do not change anymore [12].

B. Artificial Neural Networks

Artificial neural networks are composed of interconnecting artificial neurons (programming constructs that mimic the properties of biological neurons). Therefore Artificial neural networks may either be used to gain an understanding of biological neural networks; or for solving artificial intelligence problems without necessarily creating a model of a real biological system. Therefore real; biological nervous system is highly complex: artificial neural network algorithms attempt to abstract this complexity and focus on what may hypothetically matter most from an information processing point of view. Good performance (e.g. as measured by good predictive ability, low generalization error), or performance mimicking animal or human error patterns, can then be used as one source of evidence towards supporting the hypothesis that the abstraction really captured something important from the point of view of information processing in the brain [20]. Another incentive for these abstractions is to reduce the amount of computation required to simulate artificial neural networks; so as to allow one to experiment with larger networks and train them on larger data sets. And application areas of ANNs include system identification and control (vehicle control; process control); game-playing and decision making (backgammon, chess, racing), pattern recognition (radar systems; face identification; object recognition); sequence recognition (gesture, speech, handwritten text recognition); medical diagnosis; financial applications; data mining (or knowledge discovery in databases, "KDD"); visualization and e-mail spam filtering.

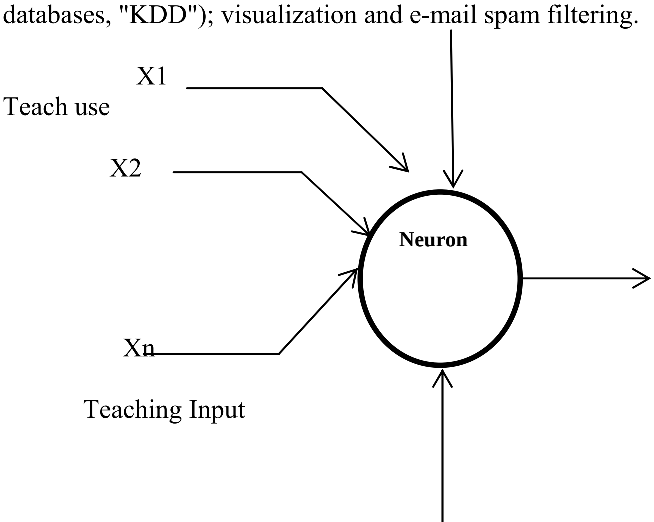

Figure 1: Neural Network

C. Delta Rule

The delta rule is a gradient descent learning rule for updating the weights of the artificial neurons in a single-layer perceptron. This is a special case of the more general back propagation algorithm. For a neuron j with activation function g(x); the delta rule for j';, ith weight is given by

$$\Delta W_{ij} = (t_j - y_j)\, g'\,(h_j) x_i \qquad (1)$$

Therefore delta rule is commonly stated in simplified form for a perceptron with a linear activation function as $\Delta W_{ij} = \alpha\, (t_j - y_j)\, x_i$; where α is known as the learning rate parameter.

V. SUPPORT VECTOR MACHINE

The Support Vector Machine (SVM) is a state-of-the-art classification method introduced in 1992 by Boser, Guyon, and Vapnik. The SVM classifier is widely used in bioinformatics (and other disciplines) due to its highly accurate; able to calculate and process the high-dimensional data such as gene expression and exibility in modeling diverse sources of data .SVMs belong to the general category of kernel methods. And a kernel method is an algorithm that depends on the data only through dot-products. This is the case; the dot product can be replaced by a kernel function which computes a dot product in some possibly high dimensional feature space. It has two advantages: First; the ability to generate non-linear decision boundaries using methods designed for linear classifiers. And second; the use of kernel functions allows the user to apply a classifier to data that have no obvious fixed-dimensional vector space representation. Thus prime example of such data in bioinformatics are sequence; either DNA or protein; and protein structure. Using SVMs effectively requires an understanding of how they work. When training an SVM the practitioner needs to make a number of decisions: how to preprocess the data, what kernel to use; and finally; setting the parameters of the SVM and the kernel [1]. Uninformed choices may result in severely reduced performance. Therefore we aim to provide the user with an intuitive understanding of these choices and provide general usage guidelines [7,13]. All the examples shown were generated using the PyML machine learning environment,;which focuses on kernel methods and SVMs.

A. PRELIMINARIES: LINEAR CLASSIFIERS

Support vector machines are an example of a linear two-class classifier. This section explains what that means. The data for a two class learning problem consists of objects labeled with one of two labels corresponding to the two classes; for convenience we assume the labels are +1 or -1. In what follows boldface x denotes a vector with components xi. Thus notation xi will denote the ith vector in a dataset, f(xi; yi)gni =1, where yi is the label associated with xi. The boundary between regions classified as positive and negative is called the decision boundary of the classifier. The decision boundary defined by a hyper plane is said to be linear because it is linear in the input examples. A classifier with a linear decision boundary is called a linear classifier. Conversely, when the decision boundary of a classifier depends on the data in a non-linear the classifier is said to be non-linear.

B. KERNELS: FROM LINEAR TO NON-LINEAR CLASSIFIERS

In many applications a non-linear classifier provides better accuracy. Yet; linear classifiers have advantages; one of them being that they often have simple training algorithms that scale well with the number of examples [9, 10]. This begs the question: Can the machinery of linear classifiers be extended to generate non-linear decision boundaries? Therefore furthermore; can we handle domains such as protein sequences or structures where a representation in a fixed dimensional vector space is not available? The naive way of making a non-linear classifier out of a linear classifier is to map our data from the input space X to a feature space F using a non-linear function.

The approach of explicitly computing non-linear features does not scale well with the number of input features: when applying the mapping from the above example the dimensionality of the feature space F is quadratic in the dimensionality of the original space. The result in a quadratic increase in memory usage for storing the features and a quadratic increase in the time required to compute the discriminant function of the classifier. The quadratic complexity is feasible for low dimensional data; but when handling gene expression data that can have thousands of dimensions; quadratic complexity in the number of dimensions is not acceptable. And Kernel methods solve this issue by avoiding the step of explicitly mapping the data to a high dimensional feature-space.

Gaussian kernel is defined by:

$$K(\mathbf{x_i}, \mathbf{x_j}) = \exp(-\frac{\|\mathbf{x_i} - \mathbf{x_j}\|^2}{2\sigma^2}) \qquad (2)$$

Where k > 0 is a parameter that control the width of Gaussian. It plays a similar role as the degree of the polynomial kernel in controlling the exibility of the resulting classifier. We saw that a linear decision boundary can be kernelized i.e. its dependence on the data is only through dot products. In order for this to be useful, the training algorithms need to be kernelizable as well [6]. It turns out that a large number of machine learning algorithms can be expressed using kernels | including ridge regression, the perceptron algorithm, and SVMs [16].

C. SVMS FOR UNBALANCED DATA

Many datasets encountered in bioinformatics and other areas of application are unbalanced; i.e. one class contains a lot more examples than the other. Therefore unbalanced datasets can present a challenge when training a classifier and SVMs are no exception see [13] for a general overview of the issue. A good strategy for producing a high-accuracy classifier on imbalanced data is to classify any example as belonging to the majority class; this is called the majority-class classifier. While highly accurate under the standard measure of accuracy such a classifier is not very useful [12]. When presented with an unbalanced dataset that is not linearly separable, an SVM that follows the formulation will often produce a classifier that behaves similarly to the majority-class classifier. The crux of the problem is that the standard notion of accuracy (the success rate; or fraction of correctly classified examples) is not a good way to measure the success of a classifier applied balanced data, as is evident by the fact that the majority-class classifier performs well under it. This problem with the success rate is that it assigns equal importance to errors made on examples belonging the majority class and errors made on examples belonging to the minority class. Therefore correct for the imbalance in the data we need to assign different costs for misclassification to each class.

VI CONCLUSION

Content-Based Image Retrieval (CBIR) is a exigent task which retrieves the related images from the database. Many CBIR techniques have been proposed earlier but they were not good enough and can be temporarily tampered with so the task was not fulfilled. We proposed 'Image Matching Based on Improved SURF Algorithm using SVM Classifier and Neural Network'. Therefore most of the CBIR system uses the low-level features such as colour; texture and shape to extract the features from the images.

REFERENCES

[1] B Zitova, J Flusser. Image registration methods: A survey. Image Vis. Comput. 2003; 21(11): 977-1000.

[2] Dai X, Khorram S. Development of a Feature-Based Approach to Automated Image Registration for Multi temporal and Multi sensor Remotely Sensed Imagery. Proceedings of the IEEE Interactional Geoscience and Remoth Sensing Symposium. 1997; 1: 243-245.

[3] Goshtasby A. Mbolieally-assisted approach to digital image registration with application in computer vision. Ph. D. dissertation, Dept. of Computer Science, Miehigan State University, TR83-013. 1983.

[4] Ton J, Jain A. Stering Landsat images by point matehing. IEEE Transactions on Geoscience and Remote Sensing. 1989; 27(5): 642-651.

[5] Li H, Manjunath B, Mitra S. Ntour-Based Approach to Multisensor Image Registration. IEEE Transactions on Image Processing. 1995; 4(3): 320-334.

[6] Tham J, Ranganath S, Ranganath M, Kassim A. Vel Unrestricted Center-Biased Diamond Search Algorithm for Block Motion Estimation. IEEE Transactions on Circuits System and Video Technology. 1998; 8(4): 369-377.

[7] Yuan Z, Wu F, Zhuang Y. I-sensor image registration using multi-resolution shape analysis. Journal Of Zhejiang University Science A. 2006; (4): 549-555.

[8] Ziou D, Tabbone S. Dtection techniques-an overview. International Journal of Pattern Recognition and Image Analysis. 1988; (5): 537-559.

[9] Mount D, Netanyahu N, Moigne J. Ent Algorithms for Robust Feature Matching. Pattern Recognition, 1999; 32(1): 17-38.

[10] Bay H, Tuvtellars T, Gool L Van. SURF: speeded up robust features. Proceedings of the European Conference on Computer Vision. 2006: 404 417.

[11] Rong W, Chen H, et al. Icing of Microscope Images based on SURF. 24th International Conference Image and Vision Computing New Zealand (IVCNZ2009). 2009: 272-275.

[12] Hartley R, Zisserman A. View Geometry in Computer Vision, second. ed. Cambridge University Press, Cambridge. 2003.

[13] Tola E, Lepetit V. St local descriptor for dense matching. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Washington, DC: IEEE Computer Society. 2008: 1-8.

[14] Tola E, Lepetit VY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo. IEEE Transactions on Pattern Analysis & Machine Intelligence. 2010; 32(5): 815-830.
[15] Bracewell R. Fourier Transform and Its Applications. McGraw-Hill, New York. 1965.
[16] Xiaoli Li, Xiaohong Wang and Chunsheng Li: Image Matching Based on Unification, The Fourth International Joint Conference on Computational Science and Optimization (2011), p. 825-828.
[17] Herbert Bay, Andreas Ess, Tinne Tuytelaars and Luc Van Gool: Speeded-Up Robust Features (SURF), Computer Vision and Image Understanding (2008), 110(3), p. 346-359.
[18] David G. Lowe: Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision (2004), 60(2), p. 91-110.
[19] Luo Juan and Oubong Gwun: A Comparison of SIFT, PCA-SIFT and SURF, International Journal of Image Processing (2009), 3(4), p. 143-152.
[20] Dorigo M., Maniezzo V. and Colorni A.: Ant System: Optimization by A Colony of Cooperating Agents, IEEE Transaction on Systems, Man, and Cybernetics-Part B (1996), 26(1), p. 29-41.