Knowledge Discovery in Databases (KDD) with Images: A Novel Approach toward Image Mining and Processing

Pardeep Kumar
Deptt. of CSE & ICT
Jaypee University of Information Technology
Waknaghat,H.P, India

Vivek Kumar Sehgal
Deptt. of ECE
Jaypee University of Information Technology
Waknaghat, H.P, India

Durg Singh Chauhan
Uttrakhand Technical University
Dehradun ,Uttrakhand, India

ABSTRACT
We are in an age often referred to as the information age. In this information age, because we believe that information leads to power and success, from the technologies such as computers, satellites, etc., we have been collecting tremendous amounts of information. Our ability to analyze and understand massive datasets lags far behind our ability to gather and store data. Image and video data contains abundant, rich information for data miners to explore. On one hand, the rich literature on image and video data analysis will naturally provide many advanced methods that may help mining other kinds of data. On the other hand, recent research on data mining will also provide some new, interesting methods that may benefit image and video data retrieval and analysis. Today, a lot of data is available everywhere but the ability to understand and make use of that data is very less. Whether the context is business, medicine, science or government, the datasets themselves are of little value. What is of value is the knowledge that can be inferred from the data and put to use. We need systems which would analyze the data for us. This paper basically aims to find out important pixels of an image using one of the classification technique named as decision tree (ID-3). Our aim is to separate the important and unimportant pixels of an image using simple rules. Further one of the compression techniques named as Huffman algorithm is applied for image compression. Finally, resultant image is stored with lesser space complexity.

Keywords
KDD, Data Mining, Image Processing, Compression Ratio, Information Gain.

1. INTRODUCTION
It has been popularly recognized that the rapid development of computer and information technology in the last twenty years has fundamentally changed almost every field in science and engineering, transforming many disciplines from data poor to increasingly data-rich, and calling for the development of new, data-intensive methods to conduct re-search in science and engineering. Thus the new terms like, data science [1] or data engineering can be used to best characterize the data-intensive nature of today's science and engineering.

Data mining deals with the discovery of hidden knowledge, unexpected patterns and new rules from large databases. Data mining is regarded as the key element of a much more elaborate process called Knowledge Discovery in Databases (KDD) which is defined as the non – trivial process of identifying valid, novel, and ultimately understandable patterns in large databases [2]. Data mining is a multidisciplinary field, drawing work from areas including database technology, information retrieval, pattern recognition, data visualization, neural networks etc [6-7]. It is a particular data analysis technique that focuses on modeling and knowledge discovery for predictive rather than purely descriptive purposes. Data mining has attracted a great deal of attention in the information industry in recent years. This is due to the wide availability of huge amounts of data and the imminent need for turning such data into useful information and knowledge. Over recent years data mining has been establishing itself as one of the major disciplines in computer science with growing industrial impact. Undoubtedly, research in data mining will continue and even increase over coming decades. Fusion of data mining with image processing is a promising research frontier [7-9].

Image processing is a physical process used to convert an image signal into a physical image. The image signal can be either digital or analog. The actual output itself can be an actual physical image or the characteristics of an image. The most common type of image processing is photography. In this process, an image is captured using a camera to create a digital or analog image. In order to produce a physical picture, the image is processed using the appropriate technology based on the input source type. Advances in image acquisition and storage technology have led to tremendous growth in very large and detailed image databases. These images, if analyzed, can reveal useful information to the human users. Image mining deals with the extraction of implicit knowledge, image data relationship, or other patterns not explicitly stored in the images. Image mining is more than just an extension of data mining to image domain. It is an interdisciplinary endeavor that draws upon expertise in computer vision, image processing, image retrieval, data mining, machine learning, database, and artificial intelligence.

There is lot of research going in the machine learning and statistics communities on algorithms for effective and efficient transmission of huge data over channels. Jiawei Han suggests Data Mining for Image/Video Processing as a research frontier [1]. C. Ordonez and E. Omiecinski , Kun-Che Lu and Don-Lin Yang have done excellent work in such a research promising domain [4,5]. In this paper, we are going to give an approach to compress an image effectively using fusion of data mining with image processing. Section 2 describes the algorithms in use. Section 3 describes the existing and proposed schemes. Section

4 and section 5 describe the implementation and results. Section 6 describes the conclusion.

2. ALGORITHMS

Huffman Algorithm: It is an algorithm used for lossless data compression developed by David A. Huffman as a PhD student at MIT in 1952, and published in A Method for the Construction of Minimum-Redundancy Codes [10].

Huffman Codes" are widely used applications that involve the compression and transmission of digital data, such as: fax machines, modems, computer networks, and high-definition television (HDTV), etc. Data compression is the necessity to reduce the space required to store an image on intermediate processing devices like routers etc and the time to transmit large files. The basic idea behind this algorithm is that it uses a variable-length code table for encoding a source symbol (such as a character in a file) where the variable-length code table has been derived in a particular way based on the frequency of occurrence for each possible value of the source symbol.

Pseudo code1:

```
HUFFMAN(C)
1 n← |C|
2 Q n← C
3 for i ←1 to n - 1
4  do ALLOCATE-NODE(z)
5 left[z]← x ←EXTRACT-MIN(Q)
6 right[z]← y ←EXTRACT-MIN(Q)
7 f[z]←f[x] + f[y]
8 INSERT(Q, z)
9 return EXTRACT-MIN(Q) [10-11]
```

C is a set of n characters and that each character c. C is an object with a defined frequency f[c]. A min-priority queue Q, keyed on f, is used to identify the two least-frequent objects to merge together. The result of the merger of two objects is a new object whose frequency is the sum of the frequencies of the two objects that were merged.

The running time complexity of Huffman's algorithm assumes that Q is implemented as a binary min-heap. For a set C of n characters, the initialization of Q in line 2 can be performed in O(n) time using the BUILD-MIN-HEAP procedure.

The for loop in lines 3-8 is executed exactly |n| - 1 times. Each heap operation requires time O(logn).The loop contributes (|n| - 1) * O(log n) = O(nlog n). Thus, the total running time of HUFFMAN on a set of n characters is given by equation 1

$$O(n) + O(n\log n) = O(n \log n) \quad (1)$$

ID-3 Algorithm: ID3(Iterative Dichotomizer 3) algorithm is a process for the classification and analysis of information hidden in large data sets/databases, which retrieves useful information in the form of a decision tree i.e. a flowchart like tree structure. The algorithm adopts a greedy approach in which the decision trees are constructed in a top-down recursive divide and conquer manner on the basis of a training set employing an attribute selection measure. To get the fastest decision-making procedure, one has to arrange attributes in a decision tree in a proper order- the most discriminating attribute first. The most discriminating attribute can be defined in precise terms as the attribute for which the fixing its value changes the entropy of possible decisions at most. Let S be a set consisting of s data samples. Suppose the class label attribute has m distinct values defining m distinct classes, C_i (for i=1,….,m). Let s_i be the number of samples of S in class C_i. The expected information needed to classify a given sample is given by equation 2

$$I(s_1, s_2, \ldots, s_m) = -\sum_{i=1}^{m} p_i \log(p_i) \quad (2)$$

where p_i is the probability that an arbitrary sample belongs to class C_i and is estimated by s_i/s. Note that a log function to the base 2 is used since the information is encoded in bits. Let attribute A have v distinct values,$\{a_1, a_2, \ldots\ldots, a_v\}$. Attribute A can be used to partition S into v subsets, $\{s_1, s_2, \ldots..,s_v\}$, where S_j contains those samples in S that have value a_j of A. If A were selected as the test attribute (i.e. the best attribute for splitting), then these subsets would correspond to the branches grown from the node containing the set S. Let s_{ij} be the number of samples of class C_i in a subset. The entropy, or expected information based on the partitioning into subsets by A, is given by equation 3

$$E(A) = \sum_{j=1}^{v} \frac{s_{1j} + \cdots + s_{mj}}{s} \cdot I(s_{1j}, s_{2j}, \ldots, s_{mj}) \quad (3)$$

The term $\frac{s_{1j} + \cdots + s_{mj}}{s}$ acts as the weight of the jth subset and is the number of samples in the subset (i.e., having value a_j of A) divided by the total number of samples in S. The smaller the entropy value, the greater the purity of the subset partitions. Note that for a given subset s_i, $I(s_{1j}, s_{2j}, \ldots, s_{mj}) = -\sum_{i}^{m} p_{ij} \log(p_{ij})$ where and is the probability that a sample in Sj belongs to class Ci. The encoding information that would be gained by branching on A is given by equation 4

$$\text{Gain(A)} = I(s_1, s_2, \ldots, s_m) - \text{E(A)} \quad (4)$$

In other words, Gain (A) is the expected reduction in entropy caused by knowing the value of attribute A. The algorithm computes the information gain of each attribute. The attribute with the highest information gain is chosen as the test attribute for the given set S. A node is created and labeled with the attribute, branches are created for each value of the attribute, and the samples are partitioned accordingly [6, 12- 15].

Pseudo code2:

D_T //Generate a decision tree from the training tuples of data partition P.

Input: Data partition, P, which is a set of training tuples along with their class labels; att_list, the set of candidate attributes; and att_sel_measure, a procedure to determine the division criteria.

Output: A decision tree.

Steps: (1) Create a node N

(2) If tuples in P are all of the same class C; return N as a leaf node labeled with class C;

(3)If att_list = NULL, then return N as leaf node labeled with the majority class in P //majority voting

(4)apply att_sel_measure(P,att_list) to find the best splitting point.

(5)label node N with best splitting point.

(6) If split_att is discrete valued then att_list←att_list-splitting att.

(7) for each outcome j of splitting point// partition the tuples and grow subtrees for each partition

(8) Let Pj be the set of data tuples in P satisfying outcome j; // a partition

(9) if Pj = NULL, then attach a leaf node labeled with the majority class in P to node N.

(10)else attach the node returned by D_T(Pj, att_list) to node N.

(11) endfor

(12) return N [3]

The running time complexity of the Pseudo code2 with given training set P is given by equation 5

$$O(n*|P|\log|P|) \quad (5)$$

where n is the number of attributes describing the tuples in P and |P| is the number of training tuples in P. This signifies that the computational cost of growing a tree grows at most n* |P|*log(|P|) with |P| tuples.

Proposed Algorithm: The pseudo code for the proposed algorithm is given below:

Pseudo code3:

Hybrid_Algo// Generate image with less memory requirement.

1. A(x,y,l)←Image

2. Call Pseudo code2(A(x,y,l)

3. Output← Pseudo code2// return from Pseudo code 2 and computation is stored in Output.

4. Call Pseudo code1(Output)

5. Output1← Pseudocode1// return from Pseudo code1 and computation is stored in Output1.

6. Compute compression ratio for Output1

The running time complexity of the proposed scheme is given by equation 6 as

$$O(n*|P|\log|P|)*O(n\log n) \quad (6)$$

as Pseudo code1 and Pseudo code2 are called from Pseudo code3 in a sequential manner.

3. EXISTING AND PRPOSED APPROACH

3.1 Existing Approach

Figure 1 describes the existing compression scheme. Pixel representation of image becomes the input for Huffman algorithm. Compressed image is the output.

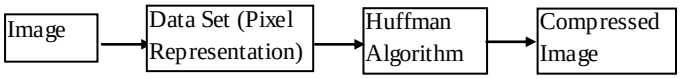

Fig 1: Existing Compression Approach

3.2 Proposed Approach

Figure 2 describes the proposed compression scheme. Pixel representation of image becomes the input for Decision tree algorithm (ID-3). Image with relevant pixels becomes the input for Huffman algorithm. Compressed image is the output.

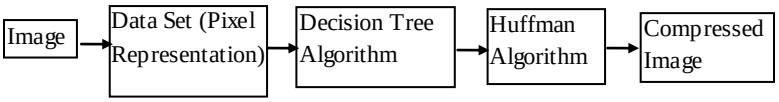

Fig 2: Proposed Compression Approach

The proposed approach for the evaluation of important and unimportant pixels follows the following model as shown in figure3.

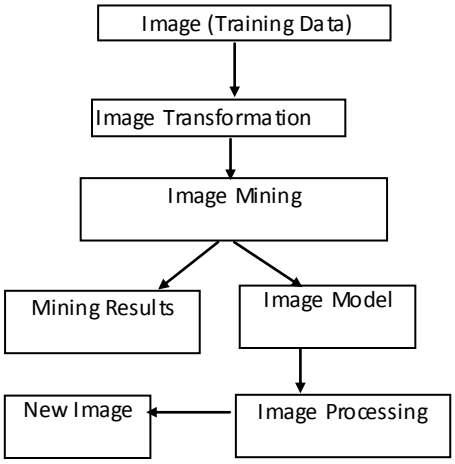

Fig3: Evaluation of important and Unimportant Pixels [5].

Image transformation deals with transformation of an input image data-set into a data base like table. The raw image is a d-dimensional light-intensity function, denoted by $R(c_1, c_2, \ldots, c_d)$, where the amplitude (or value) of R at spatial coordinates $(c_1, c_2, \ldots, c_d)$ gives the intensity of the raw image at that point (or pixel). The database-like table $X = \{x_1, x_2, \ldots, x_t\}$ is a set of records, where each record $x_r \in \Re_k$ is a vector with elements $<a_1, a_2, \ldots, a_k>$ being the value of attributes (or features) of X [5].

After having obtained such a database-like table in accordance to the desired input image dataset, mining algorithms can then be used on it. Due to the higher level comprehensibility of the decision trees (ID-3 in our case), we are using them for our

proposed method of image compression. Attributes with pixels of database like table X satisfying a particular entropy level are permitted to appear in the resultant image. Such an image model will be used for further image processing as compression in our proposed approach.

4. IMPLEMENTATION

The existing and proposed hybrid approach has been implemented using java based WEKA tool. Lena image is used for implementation purpose.

5. RESULT

Table 1 shows the result with and without data mining approach. After discretization, image dimensions are of the order of 239 X 239 pixels.

Table 1. Result with and without data mining approach

Scheme	Input(Discretized)	Output
Existing Approach	239 X 239 pixels	6,99,508 bits
Proposed Approach	13 X 239 pixels	36,729 bits

Compression Ratio =Output (Bits required without ID3)/ Output (Bits required after ID3)

6. CONCLUSION

With the advanced development in Internet, teleconferencing, multimedia and high-definition television technologies, the amount of information that is handled by computers has grown exponentially over the past decades. Hence storage and transmission of the digital image component of multimedia systems is a major problem. Here is where the real application of this paper lies. Our paper provides one of the possible solutions to such a problem by merging data mining in image compressions. The information which can be available from our research could be used in image storage and their transmission through channels used in communication systems. In this paper decision tree based data mining approach is used to obtain important and unimportant pixels from a given image. Image attributes which constitute the decision rules are to be stored for further image compression. Although compression ratio of the proposed scheme is very good yet quality of the image may not be good due to huge redundancy in the given image pixels. The proposed model requires label information of image pixels in advance; however, in some situations this label information may be unavailable or undetermined. Sometimes, it might be necessary to find the actual hidden label properties, and our future work is to refine the proposed model to an unsupervised one, which can automatically analyze and determine the label information for further use. On the other hand, we are trying to tailor this general framework to a particular case.

Further work is going on to retrieve important pixels from the rules discovered in ID3 - **Inverse Decision Tree Learning –** it is the process to retrieve a continuous image from the rules generated. Recently, machine-learning-based inverse modeling techniques have been proposed.

So finally, there is a huge potential to integrate data mining and image data retrieval/analysis and much joint research effort is needed in this promising direction.

7. REFERENCES

[1] Jiawei Han. 2008. Data Mining for Image/Video Processing: A promising research frontier.Department of Computer Science, University of Illinois at Urbana-Champaign.

[2] U.M. Fayyad, G. P. Shapiro and P. Smyth. 1996. The KDD process for extracting useful knowledge from volumes from data. Communication of ACM, Vol. 39(11), pp.27 – 34

[3] U. Fayyad, D. Haussler, and P. Stolorz.1996. Mining scientific data. Communications of the ACM, Vol. 39 pp. 51-57.

[4] C.Ordonez and E. Omiecinski. 1998. Image mining: A new approach for data mining. Technical Report GIT-CC-98-12, College of Computing, Georgia Institute of Technology.

[5] Kun-Che Lu and Don-Lin Yang. 2009. Image processing and image mining using decision trees. Journal of Information Science and Engineering, vol 25,pp. 989-1003

[6] J.Han and M. Kamber. 2006. Data Mining: Concepts and Techniques (2nd Ed.). Morgan Kaufmann.

[7] Jiawei Han. 2009. Research challenges for data mining in science and engineering.Department of Computer Science and Engineering, University of Illinois at Urbana-Champaign.

[8] J.Gray and A.Szalay. 2002. The world wide telescope: An archetype for online science. Comm. ACM, pp50-54.

[9] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. 1996. From data mining to knowledge discovery: An overview. In Advances in Knowledge Discovery and Data Mining, Eds. AAAI/MIT Press, Cambridge, Mass.

[10] Huffman, D.A. 1952. A Method for the construction of Minimum Redundancy Codes, Proceedings of the IRE, pp1098-1110.

[11] Rcnato M. Capocelli, Alfredo De Santis. 1991. A Note on D-ary Huffman Codes, IEEE Transaction of Information Theory, Vol 17. No I.

[12] J.R.Quinlan.2003.Induction in decision trees.Journal of Machine Learning, Vol.1, Issue 1, pp81 –106.

[13] M. Ankerst, C. Elsen, M. Ester, and H.-P. Kriegel.1999. Visual classification: An interactive approach to decision tree construction. In Proc. Int. Conf. Knowledge Discovery and Data Mining (KDD'99), pp 392-396, San Diego, CA.

[14] Langley, P., and Simon, H.A. 1995. Applications of machine learning and rule induction. Commun. ACM 38, Issue11, pp55-64.

[15] M.S.Chen, J. Han, and P. S. Yu. 1996. Data mining: An overview from a database perspective. IEEE Trans. Knowledge and Data Engineering, pp 866-883.