QMAS: Querying, Mining And Summarization of Multi-modal Databases*

Robson L. F. Cordeiro[1], Fan Guo[2], Donna S. Haverkamp[3], James H. Horne[3], Ellen K. Hughes[3], Gunhee Kim[2], Agma J. M. Traina[1], Caetano Traina Jr.[1], Christos Faloutsos[2]

November 2010
CMU-CS-10-144

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

[1] University of São Paulo, São Carlos SP 13560-970, Brazil.
[2] School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA.
[3] Science Applications International Corporation, McLean, VA 22102, USA.

*This is an extended version of a paper to appear in the ICDM'10 conference proceedings by the same co-authors.

Work performed during Mr. Cordeiro's visit to Carnegie Mellon University. This material is based upon work supported by FAPESP (São Paulo State Research Foundation), CAPES (Brazilian Coordination for Improvement of Higher Level Personnel), CNPq (Brazilian National Council for Supporting Research), the National Science Foundation under Grants No. DBI-0640543 and IIS-0970179, Microsoft Research, a Google Focused Research Award, and a NRO/SAIC grant. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, or other funding parties.

Keywords: attention routing, low-labor labeling, multi-modal databases

Abstract

Given a large collection of images, very few of which have labels given *a priori*, how can we automatically assign the labels of the remaining majority, and make suggestion for images that may need brand new labels distinct from existing ones? Popular automatic labeling techniques usually scale super linearly with the size of the image set, and/or their performances degrade if limited images bear initial labels. In this paper, we propose *QMAS*, an efficient solution to the following problems: (i) low-labor labeling (L3) – given a collection of images, very few of which are already labeled with keywords, find the most suitable labels for the remaining ones; and (ii) mining and attention routing – with the same input set, output a number of top representative images and top outliers. We present experimental evaluation on three data sets of proprietary and public satellite images up to a size of 2.25GB. *QMAS* scales linearly with the number of images, obtaining better or equal accuracy while being up to 40 times faster than its baseline algorithm. With limited numbers of initial labels available, *QMAS* achieves a significant accuracy margin over the baseline approach. The application of *QMAS* to recommend representatives and spot outliers is also illustrated. The proposed framework could be generalized to solve similar content-based annotation and mining problems on other multi-modal databases.

1 Introduction

The problem of automatically analyzing, labeling and understanding large collections of images appears in numerous fields. Our driving application is related to satellite imagery, involving a scenario in which a topographer wants to analyze the terrains in a collection of satellite images. We assume that each image is divided into tiles (say, 16x16 pixels). Such a user would like to label a small number of tiles ('water', 'concrete' etc), and then the ideal system would automatically find labels for all the rest. The user would also like to know what strange pieces of land exist in the analyzed regions, since they may indicate anomalies (e.g., de-forested areas, potential environmental hazards, etc.), or errors in the data collection process. Finally, the user would like to have a few tiles that best represent each kind of terrain.

Such requirements appear in several other settings, like, e.g., medical image and biological image applications: A doctor wants to find tomographies or x-rays similar to the images of his/her patient's as well as a few examples that best represent both the most typical and the most strange image patterns. [5] [11] In biology, given a collection of fly embryos [15] or protein localization patterns [10] or cat retina images [3] and their labels, we want a system to answer the same types of questions.

Our goals are summarized in two research problems:

Problem 1 *low-labor labeling (L3) –* **Given** *a collection I of N_I images,* very few *of which are labeled with keywords,* **find** *the most suitable labels for the remaining ones.*

Problem 2 *mining and attention routing –* **Given** *a collection I of N_I partially labeled images,* **find** *clusters, the N_R images that best represent the data patterns and the top-N_O outliers.*

Figure 1 illustrates the research problems and the *QMAS* results. Figure 1(a) is a sample satellite image from the city of Annapolis, MD, USA[1]. We decomposed it into $1,024$ (32x32) tiles, very few (4) of which were manually labeled as "City" (red), "Water" (cyan), "Urban Trees" (green) or "Forest" (black). Figure 1(b) shows labeling results from our *QMAS* algorithm. Notice two observations: (a) the vast majority of tiles are correctly labeled and (b) there are few outlier tiles (marked in yellow) that *QMAS* judges as too different from the labeled ones, and thus are returned to the user as outliers that potentially deserve a new label of their own. Closer inspection shows that the outlier tiles tend to be on the border of, say, "water" and "city" (because they contain a bridge).

With the same input set (Annapolis), the problem of *mining and attention routing* consists in finding clusters of tiles, the N_R best representatives for the data patterns and the top-N_O outliers. Figure 1c and Figure 1d provide *QMAS*' summarized description of the dataset by pointing out the 3 tiles that best represent the data patterns and the top-2 outliers. Notice that the representatives actually cover the 3 major keywords ("City", "Urban Trees", and "Water"), while the top outliers are hybrid tiles, like the bottom right which is a bridge (both "Water" and "City").

QMAS can go even further by summarizing the results: Besides the representatives and top outliers, *QMAS* finds clusters in the data, ignoring the user-provided labels. This has two advantages: The first is that it indicates to the user what, if any, changes have to be done to the labels:

[1] The image is publicly available at 'geoeye.com'.

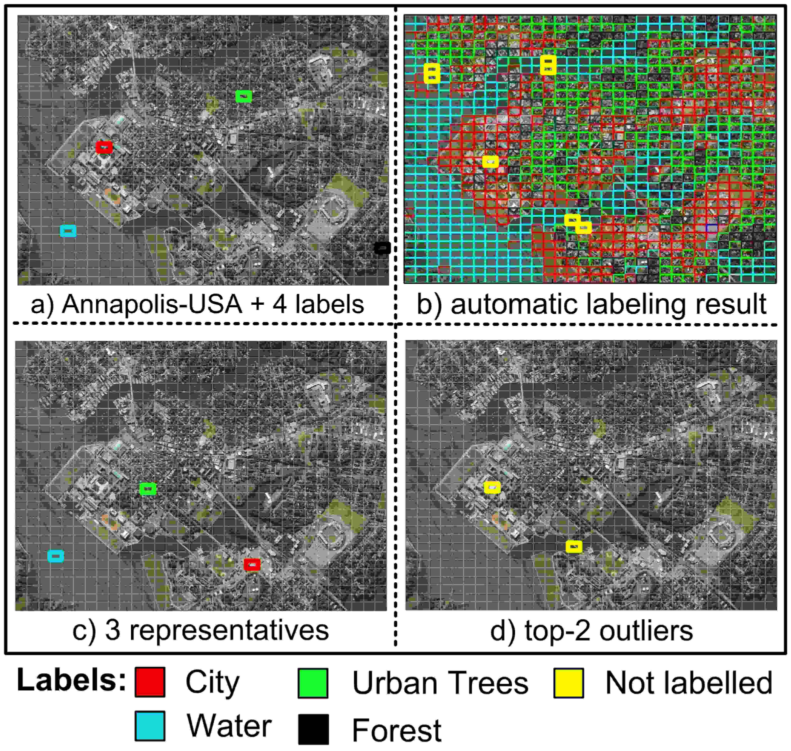


Figure 1: Our solution to the problems of *low-labor labeling (L3)* (Problem 1) and *mining and attention routing* (Problem 2). Best viewed in color - Top Left: the input satellite image of Annapolis (MD, USA), divided in $1,024$ (32x32) tiles, only 4 of which are labeled with keywords ("City" in red, etc). Top Right: the labels that *QMAS* proposes; yellow indicates outliers. Bottom Left: the 3 tiles that best represent the data, which actually cover the 3 major keywords. Bottom Right: the top-2 outlier tiles, where appropriate keywords do not exist (hybrid tiles, like the bottom right which is a bridge = "Water" and "City").

new labels may need to be created (to handle some clusters or outliers), and/or labels may need to be merged (e.g., “Forest” and “Urban trees”), and/or labels that are too general may need to be divided in two or more (“Shallow Water” and “Deep Sea”, instead of just “Water”). The second advantage is that these results can also be used for group labeling, since the user can decide to assign labels to entire clusters rather than labeling individual tiles one at a time.

In this paper we propose **QMAS: Querying, Mining And Summarization of Multi-modal Databases**. Our method is a fast ($O(N)$) solution to the problems of *low-labor labeling (L3)* (Problem 1) and, *mining and attention routing* (Problem 2). Our main contributions are summarized as follows:

- **Speed**: *QMAS* is a fast solution to the presented problems that scales linearly on the database size, being up to 40 times faster than top competitors (GCap);
- **Quality**: Our system can do *low-labor labeling (L3)*, providing results with better or equal quality when compared to the top competitors;
- **Non-labor intensive**: Our method works even when we are given very few labels – *it can still extrapolate from tiny sets of pre-labeled data*;
- **Functionality**: Contrasting to other methods, *QMAS* includes other mining tasks such as clustering and outlier and representatives detection as well as summarization. It also spots tiles that potentially require new labels;

The rest of the paper is organized as follows: the proposed strategies are presented in Section 3; the experimental results are presented in Section 4. In Section 2, we review the related work; and we conclude the paper in Section 5.

2 Related work

2.1 Labeling methods

There is an extensive body of work on the classification of unlabeled regions from partially labeled images in the computer vision field, such as image segmentation and region classification [17, 9, 19, 12]. The main learning algorithms of them are Conditional Random Fields (CRF) and boosting [17], Random Walk [9], KNN classifier [19] and Empirical Bayes [12]. The CRF based approach [17] shows the competitive accuracy for multi-class classification and segmentation, but it is relatively slow and requires a lot of training examples. The Random walk segmentation [9] is the one of methods close to our algorithm, but scalability is not discussed. It considered the segmentation of a single image. The KNN classifier [19] may be the fastest way for region labeling, but it is not robust against outliers. The Empirical Bayes approach [12] proposes a method to learn contextual information from unlabeled data. However, it may be difficult to learn the context from our image sets consisting of satellite images.

Graph-based methods provide a flexible tool for automatic image captioning. Images and caption keywords can be represented as multiple layers of nodes in a graph. Image content similarities are captured by edges between image nodes, and existing image captions become links between corresponding image and keyword. Such techniques have been previously applied in GCap [16], in which a tri-partite graph was constructed based on captioned images that were further segmented into regions. Given an image node of interest, the random walk with restart (RWR) algorithm was applied to perform proximity query to automatically assign the best annotation keyword for each region. The computation could be performed with either the power iteration method, which usually converges in no more than a few dozen iterations, or a more sophisticated approximation method such as FastRWR [18].

To create edges between similar image nodes, most previous work searches for nearest neighbors in the image feature space. However, this operation is super-linear even with the speed up offered by many approximate nearest-neighbor finding algorithms. Given millions of image tiles in satellite image analysis, greater scalability is almost mandatory.

2.2 Clustering

Several clustering algorithms exist in literature. Two common approaches are density based clustering and k-means based clustering. Density based methods assume the following cluster definition: a cluster is a region in the data space in which the objects are dense. This region may have an arbitrary shape and the points inside it may be arbitrarily distributed. Each cluster is separated from the other clusters by regions of low object density (noise). The algorithms define their own heuristics to distinguish between dense and non dense space regions, but they usually rely on user defined density thresholds. Some recent examples of density based algorithms are COPAC [1], STATPC [14] and MrCC [5].

k-means like methods start by picking k space positions as cluster centers (centroids) through a random process or by applying some specific heuristics for this task. The clustering is made possible by an iterative process that assigns objects to their closest centroids, constantly improving the centroids according to the objects assigned to each cluster. The computation stops when a quality criterion is satisfied or when a maximum number of iterations is achieved. Examples of k-means like methods are: K-Harmonic Means [21], CURLER [20], LAC [6] and LWC/CLWC [4].

The Visual Vocabulary (ViVo) [3] is a novel approach that uses Independent Component Analysis (ICA) to group image tiles into a set of visual terms, avoiding subtle problems (such as non-Gaussianity) which hurt other clustering and dimensionality reduction methods. It was developed for use with classification of biomedical images.

2.3 Feature Extraction

Feature extraction is generally considered to be a low-level image processing task and is closely related to feature detection. Histogram-based features are perhaps the most simple and popular type of features. Texture-based features such as wavelets and fractals are able to capture more subtle spatial variations such as repetitiveness. Local feature descriptors such as SIFT[13] and SURF[2] have also been widely used. Generalized Balanced Ternary (GBT) [8] is a hexagonal mathematical system used for feature extraction. A recent example of its usage in target recognition can be found in [7].

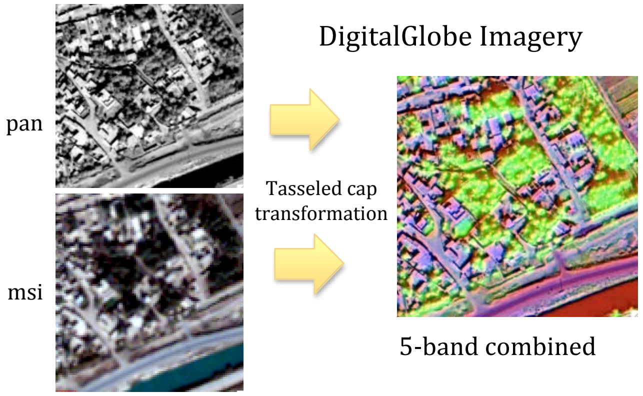


Figure 2: Pre-processing applied to multi-band satellite imagery. Best viewed in color. Left: sample input multi-band image; Right: the resulting 5- band composite image for which features are computed.

The choice of candidate features is usually domain-specific and may also be subject to scalability constraints in large scale analysis. The feature extraction procedures applied in our study will be introduced in Section 3.1.

3 Proposed Method

In this section we describe *QMAS*, our proposed solution to the problems of *low-labor labeling (L3)* (Problem 1) and *mining and attention routing* (Problem 2).

3.1 Feature extraction

A feature extraction process is first applied by *QMAS* over the input set of images. Two different approaches to feature extraction were utilized and separately tested. The type of features used for datasets *GeoEye* and *SAT1.5GB* (see Section 4, Experimental Results) was Haar wavelets in 2 resolution levels, plus the mean value of each band of the images.

For dataset *SATLARGE* (see Section 4), we extracted features using a different approach. First, pre-processing of multi-band satellite imagery is applied, resulting in a 5-band composite image for which features are computed. The first four bands are the 4-band tasseled cap transformation (TCT) of 4-band multi-spectral data, and the fifth band is the panchromatic band. The TCT results in enhanced object class separation for subsequent processing. See Figure 2 for an example.

This second approach to feature generation uses a variety of characteristics, including statistical measures, gradients, moments, and texture measures. For multi-scale image characterization, which is crucial for finding patterns at various resolutions, we use GBT. We map the raster pixel data into GBT space and calculate a set of moments-based features over the multi-scale hierarchy of GBT cells. The GBT structure is such that any cell or aggregate at a given layer in the hierarchy is composed of 7 hexagonally grouped aggregates or hexagons (if at the pixel level) in the layer below it. The cells form a hexagonal tiling of the pixels at a variety of scales, effectively

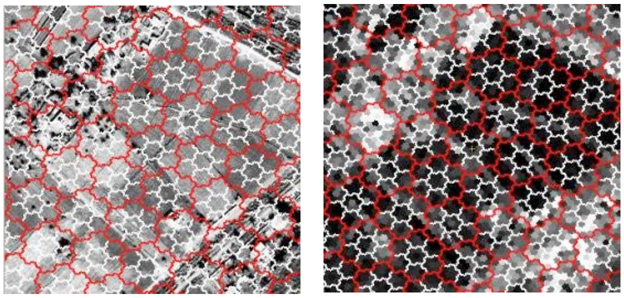


Figure 3: GBT structure illustrated. Notice the hexagonal shape. Left: two levels of GBT cells with 343 pixels (L3, outlined in white) and 2401 pixels (L4, outlined in red) overlaid on an image. Right: output values were assigned according to the variance of the next lower level of cells (at L2, consisting of 49 pixels each). Bright areas have greater variance, dark areas less.

describing image content at multiple resolutions. For every cell, we derive a complete feature set, with features such as mean, variance, gradient direction and magnitude, and texture. A sample of GBT structure and simple computations is shown in Figure 3. GBT has a natural addressing structure that supports rapid analysis of spatial relationships within the data and its hierarchical, hexagonal representation of the data has foundations in computational neuroscience. In addition, the features can be quickly computed, taking roughly a minute for a 400MB image on a standard laptop computer.

Image features such as mean, variance, and GBT texture are calculated for GBT aggregates in each of the five bands of data. The final feature set comprises a 30-dimensional feature vector per aggregate: mean, variance, and GBT texture of the Ln aggregate in each of the five data bands plus the mean, variance, and GBT texture of the Ln+1 aggregate centered at that Ln position in each of the five data bands.

Following this feature extraction, we utilize ViVo to group image tiles into a set of visual terms. ViVo's basic processing steps were modified slightly to incorporate and work with GBT aggregate features. If a tile cannot be represented by the vocabulary already known to ViVo, then it will automatically devise new types of tiles (represented by new vocabulary), as needed. The new types represent natural groupings of tiles in feature space and indicate where new labels can greatly improve the accuracy of *QMAS*. ViVo can also help identify which features are most important for labeling, and thus helps to guide the selection of features in the data.

3.2 Mining and Attention Routing

In this section we present our solution to the problem of *mining and attention routing* (Problem 2). First we do clustering on the set of images I; then we find (a) the subset of images R that best represent I, and (b) the top-N_O outliers O, sorted according to the confidence degree of it being an outlier. Algorithm 1 provides a general view of our solution to Problem 2. The details are described as follows.

Algorithm 1 : *QMAS-mining*.

```
Input: collection of images $I$;
       desired number of representatives $N_R$;
       desired number of top outliers $N_O$.
Output: clustering result $C$;
        set of representatives $R$;
        top-$N_O$ outliers $O$, in sorted order.
do soft clustering on $I$, let the result be $C$;
$R$ = random $N_R$ images from $I$;
$error = E_{QMAS}(I, R)$; // from Equation 2
repeat
    improve the representatives in $R$;
    $old\_error = error$;
    $error = E_{QMAS}(I, R)$; // from Equation 2
until $error == old\_error$
$O =$ the $N_O$ images of $I$ worst represented by $R$;
return $C$, $R$ and $O$;
```

3.2.1 Clustering

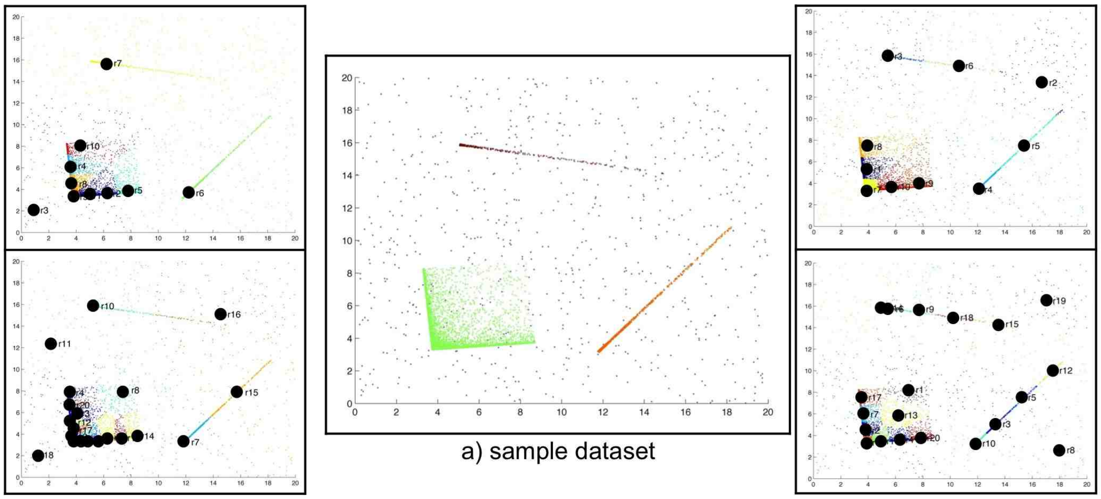
The clustering step over the set of images I is performed by a slightly modified version of the MrCC algorithm. As described in Section 2, MrCC is a fast clustering algorithm designed to look for clusters in large collections of medium-dimensionality data. We ignore MrCC's merging (third step) and use the clusters found so far as a soft clustering result, where a single tile can belong to one or more clusters with equal probabilities. This modified version of MrCC is used in our work to find clusters in the set of images I.

3.2.2 Finding Representatives

Now we focus on the problem of selecting a set of elements R, $N_R = |R|$, to represent a given set of images I. First, we discuss the desirable properties for a set of representatives, then we work on two possible approaches to actually find the representatives.

A good set of representatives R for the images in I must have the following property: *there is a big similarity between every image $I_i \in I$ and its most similar representative R_r*. Obviously, the set of representatives that best represent I is the full set of elements, $N_R = N_I \Rightarrow R = I$. In this case, the similarity is maximal between each image I_i and its most similar representative R_r, which is the image itself, $I_i = R_r$. However, when $N_R < N_I$ the goodness computation needs further evaluation.

A simple way to evaluate the goodness of a given representatives collection is to sum the squared distances between each image I_i and its closest representative R_r. This gives us an error function that should be minimized in order to achieve the best set of representatives R for a given set of images I. Not by a coincidence, this is the error function minimized by the classic clustering algorithm K-Means, which is formally defined as:



a) sample dataset

b) K-Means representatives

c) QMAS representatives

Figure 4: Examples of representative picking on synthetic data. Center: sample dataset with three clusters following skewed distributions; Borders: representatives selected by K-Means (left) and *QMAS* (right), for $N_R = 10$ (top) and 20 (bottom). These are the results with the smallest error over 50 runs.

$$E_{KM}(I, R) = \sum_{I_i \in I} MIN\{\|I_i - R_r\|^2 \mid R_r \in R\}, \tag{1}$$

where $\|I_i - R_r\|$ is the distance between I_i and R_r, and MIN is a function that returns the minimum value within its input set of values. Without loss of generality, the Euclidean distance L_2 is considered here.

Based on this idea, when we ask K-Means for N_R clusters, the clusters' centroids are good indicators of the data space positions where we should look for representatives. By finding the images of I which are the closest ones to each centroid, we have a set of representatives for K-Means.

Figure 4a shows a sample synthetic dataset containing three clusters. The clusters and their sizes follow skewed distributions. The sizes are $30,000$, $3,000$ and $1,000$ for the clusters in the bottom left, bottom right and top of the data space respectively. Additionally, 500 points are uniformly distributed through the data space in order to represent noise.

Figure 4b shows the representatives selected for our sample dataset by using K-Means and considering N_R as 10 (top) and 20 (bottom). The presented results are the best ones over 50 runs, i.e., the ones with the smallest error, computed by Equation 1. Notice that, in all cases, the selected representatives are excessively concentrated in the bottom right cluster, the biggest one, while the other two clusters are poorly represented, having only a few representatives each. These results indicate that K-Means is very sensitive to the data distribution, not presenting satisfactory results, especially for skewed data distributions.

We propose to use the K-Harmonic Means clustering algorithm in *QMAS*, since it is very insensitive to skewed distributions, data imbalance, and bad seeds initialization. Thus, it provides us a more robust way to look for representatives, again by asking for N_R clusters and picking the closest image of I to each cluster centroid as a representative. The minimized error function is:

$$\begin{aligned} E_{QMAS}(I,R) &= \sum_{I_i \in I} HAR\{\|I_i - R_r\|^2 \mid R_r \in R\} \\ &= \sum_{I_i \in I} \frac{N_R}{\displaystyle\sum_{R_r \in R} \frac{1}{\|I_i - R_r\|^2}}, \end{aligned} \tag{2}$$

where $\|I_i - R_r\|$ is the distance between I_i and R_r, and HAR is a function that returns the harmonic mean of its input values. The Euclidean distance L_2 is used once more, without loss of generality.

Figure 4c shows the representatives selected by *QMAS* for our sample dataset, again considering N_R as 10 (top) and 20 (bottom). Once more, the presented results are the best ones over 50 runs, this time considering the error function in Equation 2. Notice that the chosen representatives are now well distributed among the three clusters, providing to the user a summary that better describes the data patterns.

3.2.3 Finding the Top-N_O Outliers

The final task related to the problem of *mining and attention routing* is to find the top-N_O outliers O for the set of images I. In other words, O contains the N_O images of I that diverge the most from the main data patterns. The outliers must be sorted in a way that we identify the top $1st$ outlier, the top $2nd$ outlier and so on, according to the confidence degree of it being an outlier.

In order to achieve this goal, we take the representatives found in the previous section as a base for the outliers definition. Assuming that a set of representatives R is a good summary of I, the N_O images from I worst represented by R are said to be the top-N_O outliers. Consider again the error function in Equation 2. Notice that the minimized error is the summation of individual errors for each image $I_i \in I$, where the individual error with respect to I_i is given by:

$$IE_{QMAS}(I_i, R) = \frac{N_R}{\displaystyle\sum_{R_r \in R} \frac{1}{\|I_i - R_r\|^2}}, \tag{3}$$

the harmonic mean of the squared distances between I_i and each one of the representatives in R. The image $I_i \in I$ with the greatest individual error is the one which is worst represented by R, being considered the top $1st$ outlier of I. The top $2nd$ outlier is the image with the second greatest individual error, and so on. Thus, the top-N_O outliers O are defined as:

$$O = MAX(N_O,\ I,\ R,\ IE_{QMAS}), \tag{4}$$

where MAX is a function that returns the first N_O images of I, when they are sorted in descending order according to their corresponding individual errors, IE_{QMAS}.

Figure 5 shows the top-10 outliers for the sample dataset in Figure 4a, considering $N_O = 10$ and $N_R = 10$. As we can see, the top outliers are actually the most extreme cases for this data.

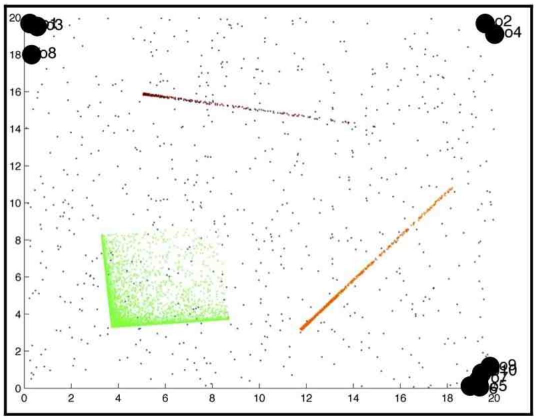


Figure 5: Top-10 outliers for the sample dataset in Figure 4a, considering the QMAS representatives from Figure 4c (top). As we can see, the top outliers are actually the most extreme cases for this data.

3.3 Low-labor Labeling (L3)

In this section we discuss our solution to the problem of *low-labor labeling (L3)* (Problem 1). In order to solve this problem, we first represent the input images and labels as a graph G, which we name as the *Knowledge Graph*. Then, random walks with restarts over G allow us to find the most suitable labels for each unlabeled image. Algorithm 2 provides a general view of our solution to Problem 1. The details are described as follows.

Algorithm 2 : *QMAS-labeling*.

```
Input: collection of images $I$;
       collection of known labels $L$;
       restart probability $c$;
       clustering result $C$. // from Algorithm 1
Output: full set of labels $LF$.
use $I$, $L$ and $C$ to build the Knowledge Graph $G$;
for each unlabeled image $I_i \in I$ do
    do random walks with restarts in $G$, using $c$ and
    always starting at the vertex $V(I_i)$;
    compute the affinity between each label of $L$ and $I_i$, let $L_l$ be the one with the biggest
    affinity;
    set in $LF$: $L_l$ is the appropriate label for image $I_i$;
end for
return $LF$;
```

G is a tri-partite graph composed of a set of vertexes V and a set of edges E, i.e., $G = (V, E)$. To build the graph, the clustering results obtained in Section 3.2, the provided sets of images I, and the known labels L are used. V consists of one vertex for each image, cluster, and label, and

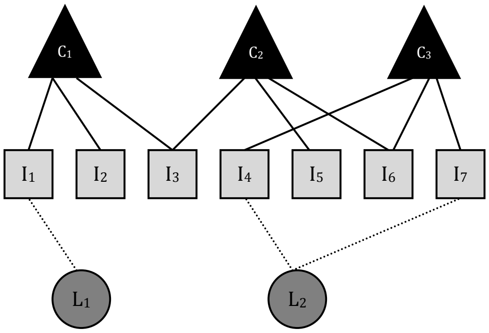


Figure 6: The *Knowledge Graph G* for a sample dataset. Nodes with shape of squares, circles, and triangles represent images, labels, and clusters respectively. The edges link images to their corresponding groups and known labels.

the edges link images to their respective clusters and labels. In our notation, $V(I_i)$ and $V(L_l)$ represent the vertexes of G related to image I_i and label L_l respectively. Provided the clustering results for the images in I, the process of building G is very simple, having linear time and memory complexities relative to the number of images, labels and clusters.

Figure 6 shows the *Knowledge Graph G* for a small sample dataset with seven images, two labels, and three clusters. In this figure, images, labels, and clusters are represented by nodes with shape of squares, circles, and triangles, respectively. As we can see, the graph indicates that cluster C_1 contains the images I_1, I_2, and I_3. Image I_3 also belongs to cluster C_2 in the example. In addition, the graph shows that image I_1 has the known label L_1, while the images I_4 and I_7 have the known label L_2.

In order to look for the most suitable label for an unlabeled image I_i, we use random walks with restarts over the graph G. This process is described as follows: a random walker starts from vertex $V(I_i)$. At each step, the walker either goes back to the initial vertex $V(I_i)$, with probability c, or to a randomly chosen vertex that shares an edge with the current vertex, with probability $1-c$. The value of c is user defined, and may be determined by cross validation. It is set to 0.15 in our experiments. And the probability of choosing to a neighboring vertex is proportional to the degree of that vertex, *i.e.*, the walker favors smaller clusters and more specific labels in this process. The affinity between I_i and a label L_l is given by the steady state probability that our random walker will find himself at vertex $V(L_l)$. Finally, the label L_l with the biggest affinity with image I_i is considered the most suitable label for I_i.

The intuition behind this procedure is that the steady state probability that a random walker will find himself in vertex $V(L_l)$, starting the walk from vertex $V(I_i)$, is a way to measure the closeness between $V(I_i)$ and $V(L_l)$. If the computed probability is high, the vertexes are probably linked by short paths. On the other hand, if the probability is low, it's likely that no short path links them.

This idea can be better understood through our example in Figure 6. Let's assume that we want to find the most appropriate label for image I_2. There is a high probability that a random walker will reach L_1 when starting the walk from I_2 because of an existing three-step path linking I_2 and

L_1. On the other hand, the probability that the walker will find himself at L_2 when starting the walk from I_2 is low because the shortest path between I_2 and L_2 has seven steps. This fact leads us to conclude that, in our example, the most appropriate label for I_2 is L_1.

4 Experimental Results

We did experiments to support our claimed contributions stated in Section 1 regarding speed, quality, non-labor intensive capability, and functionality. Our experiments related to each one of these items are presented in the following subsections.

The datasets we used are real satellite images, summarized in Table 1, which may be described as follows:

- *GeoEye*[2] – this dataset contains 14 high quality satellite images in jpeg format extracted from famous cities around the world, such as the city of Annapolis (MD, USA), illustrated in Figure 1a. The total data size is about 17 MB. We divided each image into equal-sized rectangular tiles and the entire dataset contains $14,336$ tiles, from which the used features were extracted;

- *SAT1.5GB* – this proprietary dataset contains 3 satellite images of around 500 MB each in the GeoTIFF data format. The total data size is about 1.5 GB. Each image was divided into equal-sized rectangular tiles. The 3 images combined form a set of $721,408$ tiles, from which we extracted the used features;

- *SATLARGE* – this proprietary dataset contains a pan QuickBird image of size 1.8 GB, and its matching 4-band multi-spectral image of size 450 MB. These images were combined as described previously, and 2,570,055 hexagonal tiles generated, from which we extracted the used features.

Table 1: Summary of datasets.

Dataset	# of Tiles	File Size
GeoEye	$14,336$	17 MB
SAT1.5GB	$721,409$	1.5 GB
SATLARGE	2,570,055	2.25 GB

All experiments were made in a machine with 4 GB of RAM, using a single 2.8 GHz processor and the Linux operational system. We compared *QMAS* to one of the best competitors, the GCap method, which was tested in two versions: with the original quadratic nearest neighbors (GCap) and with approximate nearest neighbors (GCap-ANN), the number of nearest neighbors for both

[2] The dataset is publicly available at: ‘geoeye.com’.

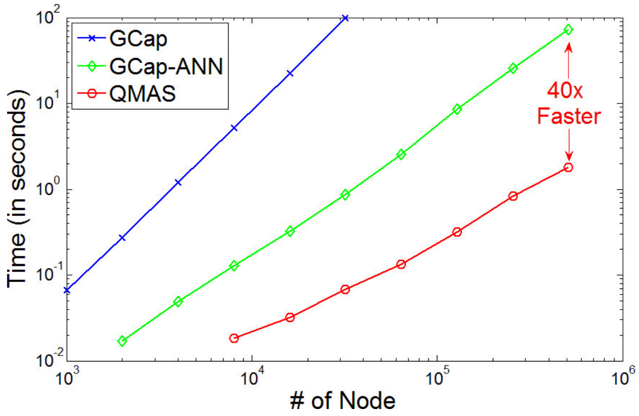


Figure 7: Time vs. # of tiles for random samples of *SAT1.5GB*. *QMAS*: red circles; GCap: blue crosses; GCap-ANN: green diamonds. *QMAS* scales linearly on the data size, while the slope of log-log curves are 2.1 for GCap and 1.5 for GCap-ANN. For the full data, *QMAS* is 40 times faster than GCap-ANN, and running GCap is prohibitive. Timing results are averaged over 10 runs; error bars are too small to be shown.

of which are set to 7. The three competitors were configured with the default restart parameter $c = 0.15$ for the random walks with restarts, which was implemented using the *power iteration method*.

4.1 Speed

QMAS is a fast solution to the problems of *low-labor labeling (L3)* (Problem 1) and *mining and attention routing* (Problem 2). Our method scales linearly on the database size, being several times faster than our competitors GCap and GCap-ANN. Figure 7 shows how the compared methods scale with increasing dataset sizes. Random samples from our *SAT1.5GB* dataset were used. *QMAS* scales linearly on the data size, while the slope of log-log curves are 2.1 for GCap and 1.5 for GCap-ANN. Notice that, for the full *SAT1.5GB* dataset, *QMAS* is 40 times faster than GCap-ANN, while running GCap will take hours long (not shown in the figure).

4.2 Quality

Our system can do *low-labor labeling (L3)* much faster than the competitors, with similar quality. In order to support this claim, we manually assigned labels to 256 tiles from the *SAT1.5GB* dataset, disclosed a small number of ground truth labels randomly selected from each class as the input, and used the remaining ones to compare with the most likely label given by each technique. Figure 8 illustrates averaged prediction accuracy from 10 repetitions in box plots. As we can see, although *QMAS* is up to 40 times faster than the competitors, prediction accuracies are no worse, and even better when pre-labeled data size is limited. Additional experiments have shown that compared with GCap-ANN with the number of nearest neighbors set to 3 and given 10 pre-labeled examples from each class, QMAS is around 10% more accurate, still being 1.75 times faster on the full data.

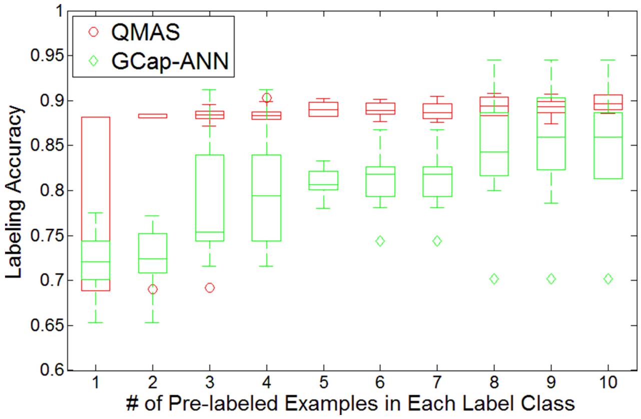


Figure 8: Comparison of approaches in box plots – Quality vs. Size of pre-labeled data. Top left is the ideal point. *QMAS*: red circles; GCap-ANN: green diamonds. Accuracy values of *QMAS* are barely affected by the size of the pre-labeled data. Results are obtained over 10 runs.

4.3 Non-labor Intensive

Our method works even when we are given very few labels – it can still extrapolate from tiny sets of pre-labeled data. The aforementioned Figure 8 presents results to support this claim: accuracies of *QMAS* are barely affected by the number of the pre-labeled examples per class, when it goes above 5.

4.4 Functionality

In contrast to the competing methods, *QMAS* includes other mining tasks such as clustering, outlier and representatives detection, and summarization. In other words, *QMAS* solves both the problem of *low-labor labeling (L3)* (Problem 1) and the problem of *mining and attention routing* (Problem 2), while its competitors address only the former. In order to support this claim, we analyzed the functionality of our method with respect to the following aspects:

1 clustering;

2 representatives;

3 top-N_O outliers;

Figure 9 and Figure 10 show some screenshots of *QMAS*'s clustering results over the *GeoEye* and the *SAT1.5GB* datasets, respectively. The results are shown by coloring each tile according to its cluster. A few tiles belong to more than one cluster, since *QMAS* does soft clustering. These were colored according to one of their assigned clusters, chosen at random. Yellow tiles represent outliers. As we can see, the clustering results really represent the main patterns apparent in the analyzed images.

Figure 11 presents *QMAS*'s results over the *GeoEye* dataset with respect to data representatives. $N_R = 6$ representatives are shown, which were colored according to their clusters. By comparing

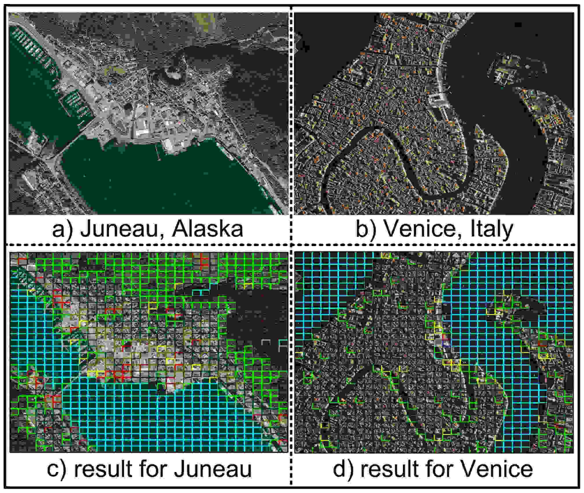


Figure 9: *QMAS*'s clustering on the *GeoEye* dataset (best viewed in color). Top: the real satellite images; Bottom: the corresponding results, shown by coloring each tile after its cluster. Yellow tiles represent outliers. Notice that the clusters actually represent the main data patterns.

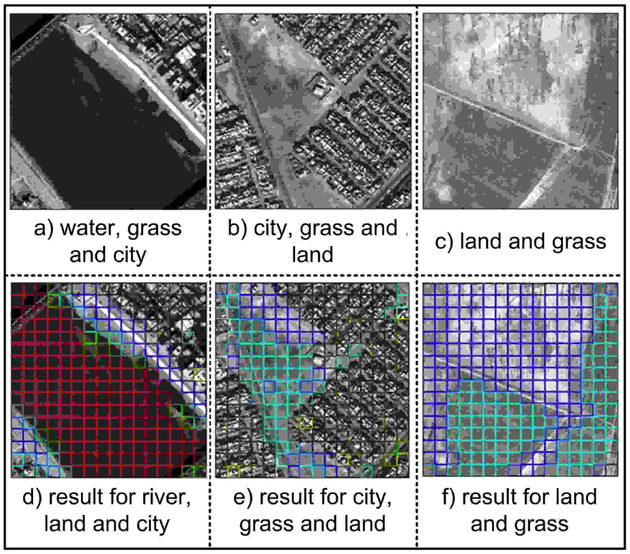

a) water, grass and city
b) city, grass and land
c) land and grass
d) result for river, land and city
e) result for city, grass and land
f) result for land and grass

Figure 10: *QMAS*'s clustering on the *SAT1.5GB* dataset (best viewed in color). Top: the real satellite images; Bottom: the corresponding results, shown by coloring each tile after its cluster. Yellow tiles represent outliers. Notice that the clusters actually represent the main data patterns.

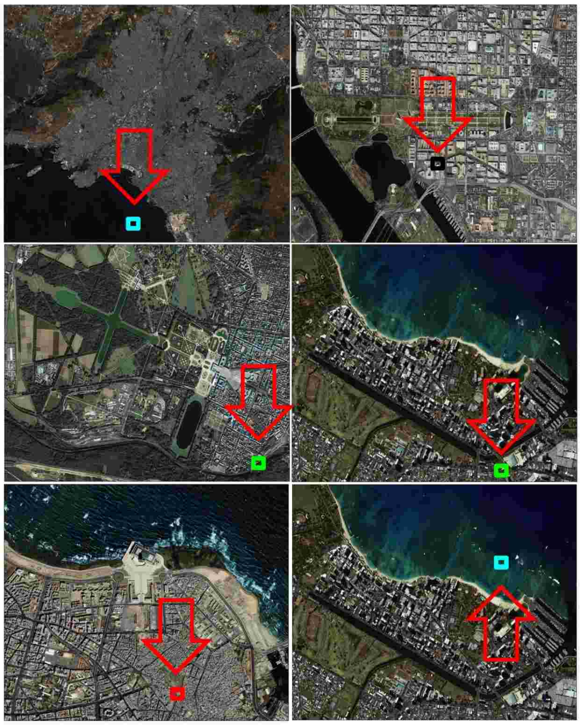


Figure 11: *QMAS*'s $N_R = 6$ representatives for the *GeoEye* dataset, colored according to cluster (best viewed in color). By comparing them to the clusters presented in Figure 9 it's easy to see that these few representatives nicely cover the main clusters.

these results to the clusters presented in Figure 9 it's easy to see that these few representatives nicely cover the main clusters.

Figure 12 presents *QMAS*'s results over the *GeoEye* dataset with respect to the top outliers. The top-3 outliers were obtained based on the 6 representatives of Figure 11. Closer inspection shows that these outlier tiles tend to be on the border of areas like “water” and “city” (because they contain a bridge). Comparing to the clusters presented in Figure 9, notice that these 3 outliers together with the 6 representatives, only 9 tiles in total, nicely summarize the *GeoEye* dataset, which contains more than 14 thousand tiles.

4.5 Experiments on the *SATLARGE* dataset

Here we present results for the *SATLARGE* dataset, related to *query by examples* experiments, i.e. given a small set of tiles (examples), manually labeled with one keyword, query the unlabeled tiles to find the ones most likely related to that keyword. Figures 13 to 18 exhibit results for several categories (water, houses, trees, etc) to show that *QMAS* returns good results, being almost insensitive to the kind of tile given as example. Figures 16 and 17 show that *QMAS*'s results are good even for tiny sets of pre-labeled data. The sizes vary from as many as 50 samples to as few as 3 samples. Varying the amount of labeled data allowed us to observe how the system responds

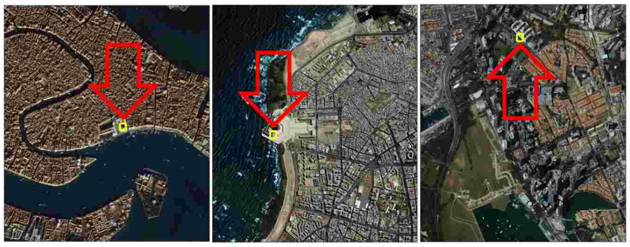


Figure 12: *QMAS*'s top-3 outliers for the *GeoEye* dataset based on the 6 representatives of Figure 11 (best viewed in color). The outlier tiles tend to be on the border of areas like "water" and "city" (because they contain a bridge). Notice that these 3 outliers together with the 6 representatives, only 9 tiles in total, nicely summarize the *GeoEye* dataset, which contains more than 14 thousand tiles.

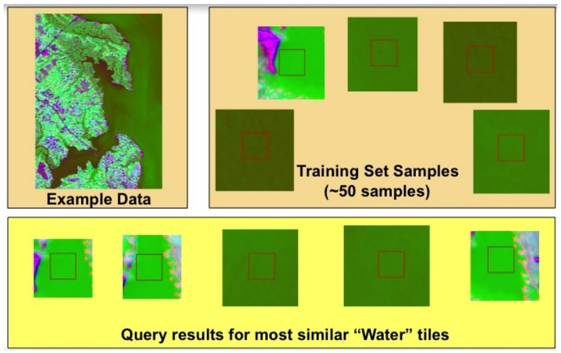


Figure 13: Example Water: Labeled Data and Results of Water Query.

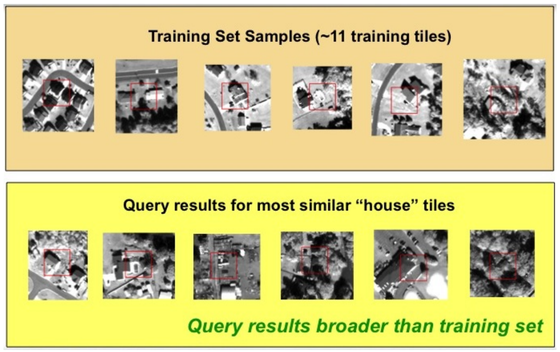

Training Set Samples (~11 training tiles)
Query results for most similar "house" tiles
Query results broader than training set

Figure 14: Example House: Labeled Data and Results of House Query.

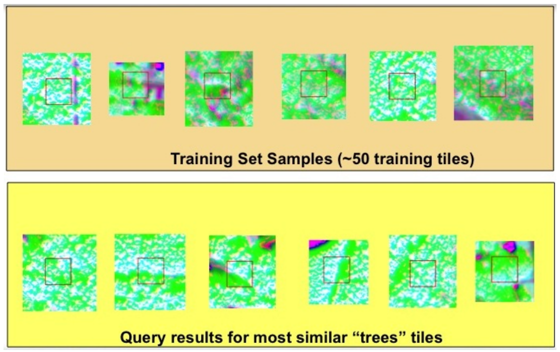

Training Set Samples (~50 training tiles)
Query results for most similar "trees" tiles

Figure 15: Example Tree: Labeled Data and Results of Trees Query.

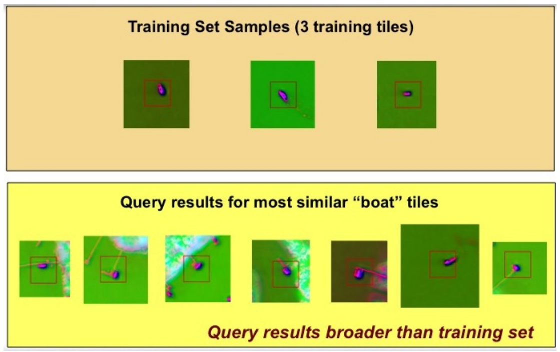

Training Set Samples (3 training tiles)
Query results for most similar "boat" tiles
Query results broader than training set

Figure 16: Example Boats: Labeled Data and Results of Boat Query.

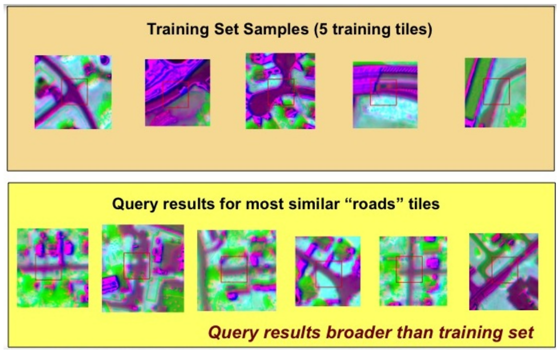

Training Set Samples (5 training tiles)
Query results for most similar "roads" tiles
Query results broader than training set

Figure 17: Example Roads: Labeled Data and Results of Road Query.

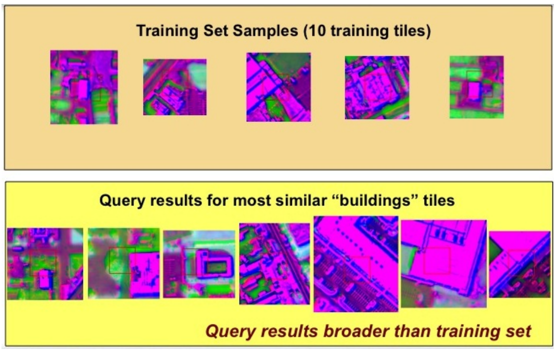

Training Set Samples (10 training tiles)
Query results for most similar "buildings" tiles
Query results broader than training set

Figure 18: Example Building: Labeled Data and Results of Buildings Query.

to these changes. In general, labeling only small numbers of examples (even less than 5) still leads to accurate results. Notice that correct returned results often look very different from the given samples. In other words, the system is able to extrapolate from the given examples to other, correct tiles that do not have significant resemblance to the pre-labeled set. Clearly, this is not a typical automated target recognition (ATR) approach. There are no "templates" and no specific object shapes, orientations, sizes, or patterns that are learned. Unlike a traditional ATR that typically fails when it encounters an object that does not fit the specified description, *QMAS* is able to correctly label an object that has a somewhat different appearance from the "known" set.

5 Conclusions

In this paper we proposed **QMAS: Querying, Mining And Summarization of Multi-modal Databases**. Our method is a fast ($O(N)$) solution to the problems of *low-labor labeling (L3)* (Problem 1) and *mining and attention routing* (Problem 2). Our main contributions, supported by experiments on real satellite images spanning up to 2.25 GB, are presented as follows:

- **Speed**: *QMAS* is a fast solution to the presented problems, and it scales linearly on the database size. It is up to 40 times faster than top competitors (GCap);
- **Quality**: Our system can do *low-labor labeling (L3)*, providing results better than or equal to the accuracy of the competitor;
- **Non-labor intensive**: Our method works even when we are given very few labels – *it can still extrapolate from tiny sets of pre-labeled data*;
- **Functionality**: In contrast to the other methods, *QMAS* spots tiles that potentially require new labels, and includes other mining tasks such as clustering and outlier / representatives detection, as well as summarization;

References

[1] Elke Achtert, Christian Böhm, Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. Robust, complete, and efficient correlation clustering. In *SDM, USA*, 2007.

[2] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359, 2008.

[3] Arnab Bhattacharya, Vebjorn Ljosa, Jia-Yu Pan, Mark R. Verardo, Hyung-Jeong Yang, Christos Faloutsos, and Ambuj K. Singh. Vivo: Visual vocabulary construction for mining biomedical images. In *ICDM*, pages 50–57. IEEE Computer Society, 2005.

[4] Hao Cheng, Kien A. Hua, and Khanh Vu. Constrained locally weighted clustering. *PVLDB*, 1(1):90–101, 2008.

[5] Robson L. F. Cordeiro, Agma J. M. Traina, Christos Faloutsos, and Caetano Traina Jr. Finding clusters in subspaces of very large, multi-dimensional datasets. In *ICDE*. IEEE Computer Society, 2010.

[6] Carlotta Domeniconi, Dimitrios Gunopulos, Sheng Ma, Bojun Yan, Muna Al-Razgan, and Dimitris Papadopoulos. Locally adaptive metrics for clustering high dimensional data. *Data Min. Knowl. Discov.*, 14(1):63–97, 2007.

[7] Laurie Gibson, James Horne, and Donna Haverkamp. Cassie: contextual analysis for spectral and spatial information extraction. In Ivan Kadar, editor, *Signal Processing, Sensor Fusion, and Target Recognition XVIII*, volume 7336. SPIE, 2009.

[8] Laurie Gibson and Dean Lucas. Spatial Data Processing Using Generalized Balanced Ternary. In *IEEE Conference on Pattern Recognition and Image Analysis*, June 1982.

[9] Leo Grady. Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(11):1768–1783, 2006.

[10] K. Huang and R. F. Murphy. From quantitative microscopy to automated image understanding. *J. Biomed. Optics*, 9:893–912, 2004.

[11] Flip Korn, Nikolaos Sidiropoulos, Christos Faloutsos, Eliot Siegel, and Zenon Protopapas. Fast nearest-neighbor search in medical image databases. *Conf. on Very Large Data Bases (VLDB)*, September 1996. Also available as Univ. of Maryland tech. report: CS-TR-3613, ISR-TR-96-13.

[12] Svetlana Lazebnik and Maxim Raginsky. An empirical bayes approach to contextual region classification. In *CVPR*, pages 2380–2387. IEEE, 2009.

[13] David G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999.

[14] Gabriela Moise and Jörg Sander. Finding non-redundant, statistically significant regions in high dimensional data: a novel approach to projected and subspace clustering. In *KDD*, pages 533–541, 2008.

[15] Jia-Yu Pan, Andre Guilherme Ribeiro Balan, Eric P. Xing, Agma J. M. Traina, and Christos Faloutsos. Automatic mining of fruit fly embryo images. *KDD*, pages 693–698, 2006.

[16] Jia-Yu Pan, Hyung-Jeong Yang, Christos Faloutsos, and Pinar Duygulu. Gcap: Graph-based automatic image captioning. In *CVPRW '04: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 9*, page 146, 2004.

[17] Jamie Shotton, John M. Winn, Carsten Rother, and Antonio Criminisi. *TextonBoost*: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In Ales Leonardis, Horst Bischof, and Axel Pinz, editors, *ECCV (1)*, volume 3951 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2006.

[18] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast random walk with restart and its applications. In *ICDM '06: Proceedings of the Sixth International Conference on Data Mining*, pages 613–622, Washington, DC, USA, 2006. IEEE Computer Society.

[19] Antonio B. Torralba, Robert Fergus, and William T. Freeman. 80 million tiny images: A large data set for non-parametric object and scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(11):1958–1970, 2008.

[20] Anthony K. H. Tung, Xin Xu, and Beng Chin Ooi. Curler: finding and visualizing nonlinear correlation clusters. In *SIGMOD*, pages 467–478, New York, NY, USA, 2005.

[21] Bin Zhang, Meichun Hsu, and Umeshwar Dayal. K-harmonic means - a spatial clustering algorithm with boosting. In John F. Roddick and Kathleen Hornsby, editors, *TSDM*, volume 2007 of *Lecture Notes in Computer Science*, pages 31–45. Springer, 2000.