**Introduction**

While significant progress has been made in information retrieval in text documents, tools for indexing multimedia data are still primitive. One of the most difficult but important components of such systems is Automatic Image Annotation. Automatic Image Annotation (also known as automatic image tagging) is the process of the computer system automatically assigning keywords to a given digital image, and this is used in image retrieval systems to locate images that are queried. In this work, we propose a new ad hoc algorithm based on machine learning to perform automatic image annotation. Scale Invariant Feature Transform (SIFT) algorithm is employed to extract the key points and feature vectors of each image. We use the k-means clustering technique to construct visual words. A set of benchmark training and testing images from Visual Object Class Challenge 2006 (VOC2006) database has been used to assess the proposed image annotation algorithm. The numerical experiments confirm the effectiveness of the primitive algorithm. Two possible improvements are given, and the numerical experiments suggest the success in improving the performance.

**Image Representation Using Visual Words**

**Key Points and Feature Vectors**

Scale Invariant Feature Transform (SIFT) [1] is a method to extract distinctive features from an image which is invariant under scaling, rotation, addition of noise, and illumination change etc. The first step of SIFT is to find the key points of the image. The basic idea of extracting the keypoints is to apply Gaussian filters at different scales of the image. Then the extreme points are taken from the difference of successive Gaussian-blurred images (Difference of Gaussians - DoG), and this is what we call keypoints. Given an image, \( I(x, y) \), and a variable-scale Gaussian, \( G(x, y) \), the space-scale of an image is defined as [1]

\[
L(x, y) = G(x, y) \ast I(x, y),
\]

where * is the convolution operator in x and y, and

\[
G(x, y) = e^{-(x^2+y^2)/2\sigma^2},
\]

and the Difference of Gaussian function can be calculated by

\[
G(x, y) = G(x, y) - G(x, y - \theta x, \theta y) \ast I(x, y) = L(x, y) - \theta L(x, y - \theta x, \theta y).
\]

where \( \theta \) is a constant multiplicative factor which separates two nearby scales. Figure on the right illustrates how this algorithm works. Interest points (called keypoints in the SIFT framework) are identified as local maxima or minima of the DoG images across scales. SIFT assigns a feature vector with 128 elements to reflect the local properties.

**Visual Word Representation**

We used the public image data sets from VOC2006 database [2] including predefined training and testing sets. The training set, which is used for training the image annotation machine, includes 1473 images and the testing set includes 2065 images. There are 10 object classes (1) bicycle, bus, cat, cow, dog, horse, motorbike, person, sheep. Every image has one or more labels.

For every image, the SIFT algorithm extracts the grey-scale intensity matrix and gives a collection of feature vectors. The bag of feature vectors collected from all the training images is then sent to the k-means algorithm. The k-means algorithm clusters all the feature vectors into k clusters. In this work, k = 1000 is chosen. Now, among the feature vectors belonging to one training image, we can have a count of feature vectors in each cluster. Each cluster can be conceptually understood as a visual word. Therefore, for each training image, there’s a visual word distribution associated to this image. This process is shown in Figure on the left. In fact, this visual word distribution is the representation of the content of this image. In addition to the visual word distribution, there’s also a collection of bags assigned to this image.

**An Ad Hoc Image Annotation Algorithm**

**Building Visual Word Distribution for Each label**

Once the visual word distributions for all the training images are built, it's ready for the labels to learn their visual word distributions. We use an example to illustrate this process.

**Image Tagging**

When a testing image is given to our image annotation engine, a confidence that each label should be assigned to this image is calculated, namely \( f_j \). Previously, we make the final labeling decision is:

\[
\text{if } f_j \geq \text{threshold } \Rightarrow \text{label } j \text{ is assigned to this image.}
\]

To improve the performance, we can use the following decision:

\[
\text{if } \sum f_j \geq \text{min}f_j + \text{threshold } \Rightarrow \text{max}f_j \text{ is assigned to this image.}
\]

**Improvement Through Multiple-Label Scheme**

Another way to make the final labeling decision is to compare the visual word distribution of the testing image with that of a single label, or a fixed label. This means that we assume an image consists of 1 or 2 labels, then we build the ideal visual word distribution for such images, and then compare the SIFT feature of the visual word distribution of images with one label or two labels. The decision making step is:

\[
\text{if } \sum f_j \geq \text{min}f_j + \text{threshold } \Rightarrow \text{max}f_j \text{ is assigned to this image.}
\]

**Conclusion**

We have built an ad hoc automatic image annotation engine and also proposed two possible ways to improve the primitive design. The performance has been tested on a set of benchmark training and testing images. The numerical experiments confirm the effectiveness of the primitive algorithm and the improvements.

**Literature**
