Objects localization in remote sensing images using local features clustering

Muhammad Farid, Khaled Badran, Gamal Elnashar
mfarid77@gmail.com , khaledbadran@hotmail.com , enjygamal@gmail.com

Electro-optics department, Military Technical College, Cairo, Egypt

ABSTRACT

There is an increasing trend towards object detection from aerial and satellite images. Most of the recent state-of the-art widely used object detection researches based on local features use the scanning of images by the sliding window. In this paper we propose an approach to localize the candidate objects by using the clustering of locations of the matched keypoints, this method has a benefits of minimizing the no of points to be processed by the classifier, and with more accuracy. In this paper, this approach is tested by SIFT and SURF local features detector and descriptor. This approach can be used as an object detection technique by itself or executed as a pre-step before apply the machine learning trained classifiers to achieve more precise results.

***Keywords**: Object detection, remote sensing, computer vision, local features, bag of visual word.*

Introduction

Object detection and classification in computer vision are very active research direction in the field of machine learning, they are widely used in many fields, including face recognition, pedestrian detection and tracking, intelligent video analysis, object recognition, and so on. Remote sensing images interpretation are one of the challenges in this field.

Remote sensing images became in the last decade a very interesting source of digital information to various scientific and military applications. These images have a resolution below 0.5 meters which give a level of details couldn't be achievable in the past, Image interpretation of this high resolution images are very tedious and time consuming process due to the size of the images and the high flow of the images. For example, a typical high-resolution image generated by WorldView-3 satellite (launched in 2014) has a size of 13 x 13 km with 0.35m resolution. This image produces 42000 x 42000 pixels. Searching for an object in such a huge image is very hard for human eyes, and very complex for any automated system. That's why we need to make a computer aid to help human eyes to automatically find locations in the images which is probably an interesting object.

In the last decades, a large number of methods have been developed for object detection from aerial and satellite images. We can generally divide them into four main categories: template matching-based methods, knowledge-based methods, OBIA-based methods, and machine learning-based methods. This paper focus on the machine learning approach of object detection. The related researches in object detection in remote sensing images like [1, 4, 5] using unsupervised methods which specifying region of interest (ROI) by grouping pixels into clusters and then using the shape features and spectral features to detect objects. Other researchers used supervised methods to train an object model with information extracted from training samples [2, 6-8]. The supervised learning methods can achieve more promising performance than the unsupervised approaches. Therefore, overwhelming object detection systems are usually based on the supervised learning techniques.

Object detection can be performed by learning a classifier that captures the variation in object appearances and views from a set of training data in a supervised or semi-supervised or weakly supervised framework. The input of the classifier is a set of regions (sliding windows or object proposals) with their corresponding feature representations and the output is their corresponding predicted labels, i.e., object or not. Feature extraction, feature fusion and dimension reduction (optional), and classifier training play the most important roles in the performance of object detection and hence we mainly focus on reviewing these three crucial steps.

The machine learning techniques classifiers process is a high cost process which needs to be run as minimum as possible, sliding window of the whole huge image (30000x30000 pixels as a typical example) is a very large repetition of running the classifier. For sliding window, multiple scan on the image must be used with different sizes of windows, there may be significant overlap on detected targets. To try to overcome this problem, an additional step of non-maximum suppression must be used as suggested in [3, 6, 9] to retain the sliding window with the highest score.

The method of find object proposals in the optical remote sensing images by using local features location clustering tries to solve the problems by train the system with a small number of positive and negative learning samples as a box contains an instance of the object without specifically draw the borders of this object to make the operation of labelling data more easy and fast. Using SIFT as a feature extraction algorithm, K-Means as a feature clustering and dimension reduction and Mean Shift to cluster matched features locations to segment and localize candidate objects.

Proposed System

Training Phase

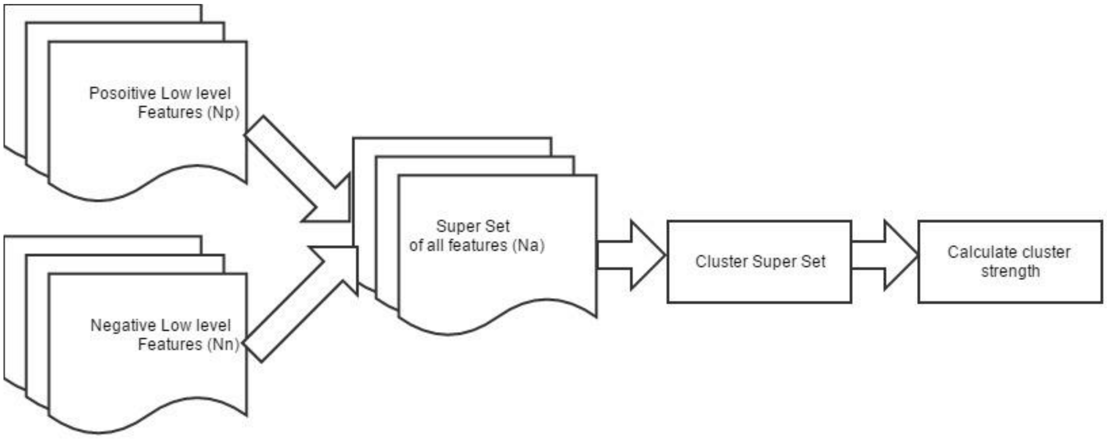
The training is done at the system initialization for the first time, the continuous training option assume that the human user of the system will review and feedback the system results to add the true positive detected objects features as a positive training data and false positive detected objects features as a passive training data. After each feedback the detected object features are added to either the positive features or the negative features according to if the object was detected correctly or not. Then a re-clustering is needed for the feature set to add the new feature to the set. The original set of unreduced size of features is saved to be used in the feedback continuous training adaptation.

Features can be extracted by any local feature extractor like SIFT, SURF, FAST. Every feature must be described mathematically by a descriptor to be compared with the descriptors of the testing image features. We have several local feature descriptors can be used as a feature descriptor like SIFT and SURF.

We use local features descriptors to characterize the region around each key point in image patches. Due to its ability to handle variations in terms of intensity, rotation, scale, and affine projection, the SIFT descriptor [10] is adopted in the proposed algorithm as the low-level descriptor to detect and describe the key points. According to existing work [2, 9, 11], the SIFT descriptor has been demonstrated to outperform a set of existing descriptors and widely used in analysing remote sensing images.

Key point detection done using the Scale Invariant Feature Transform “SIFT” capabilities, which output is a small circle on the locations of key points function finds the key point in the images. Each key point is a special structure which has many attributes like its (x,y) coordinates, size of the meaningful neighbourhood, angle which specifies its orientation, response that specifies strength of key points etc.

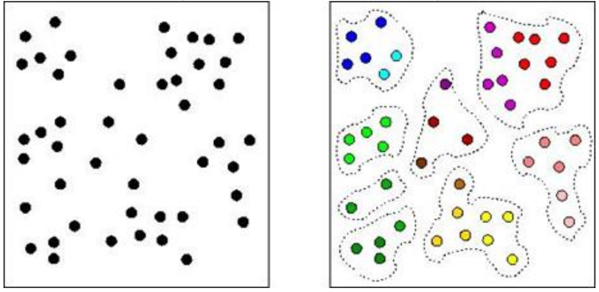
Figure 1 Training Phase


Posoitive Low level Features (Np)
Negative Low level Features (Nn)
Super Set of all features (Na)
Cluster Super Set
Calculate cluster strength

After several tens of training samples, training data can result of thousands of features, after thousands of training samples we can result to millions of features. Matching of such huge amount of training data is very complex and need a lot of processing specially it has a lot of repetitions. So we reduce the no of features by classifying these features to a limited no of feature classes, then we use the mean of the features of each class as a feature to be matched with the testing image features. We use K-means classification algorithm to do the task of minimizing the dimension of training feature space, The K is the no of classes that will be result from the clustering operation. Increasing K implies a higher precision and higher processing time. Decreasing K will simplify the model and increase detection performance but will affect the performance. K is a parameter of the model.

We then group the extracted descriptors into 1% of clusters. We use MiniBatchKMeans, a variation of K-Means that uses a random sample of the instances in each iteration. As it computes the distances to the centroids for only a sample of the instances in each iteration, MiniBatchKMeans converges more quickly but its clusters' distortions may be greater. In practice, the results are similar, and this compromise is acceptable

Fig.4 Feature set clustering



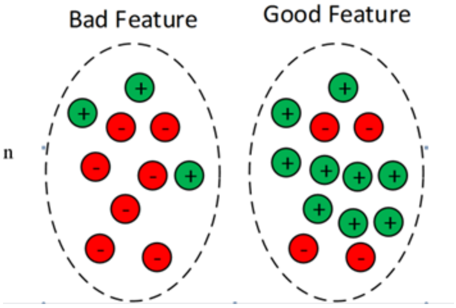
When we make the training we collect all features from both positive and negative sample images, then we make the classification on the whole feature set. Then we calculate a strength factor depending on the ratio between the no of positive features and the no of negative features per class. As no of positive features increases relative to the negative features the strength is increases to detect our interesting object (Fig. 5). We used a simple equation to calculate the values represent the strength of the feature Eq (1).

$$str(c) = \frac{N_p(c)/N_n(c)}{\sum_{c=1}^{K} N_p(c)/N_n(c)} \qquad \text{Eq. (1)}$$

Where $N_p(c)$ is the no of the positive features in cluster c, $N_n(c)$ is the no of negative feature in cluster c, K is the no of feature clusters, and $str(c)$ is the strength of feature cluster c.

The threshold of the strength can be adjusted as a parameter to increase precision by increasing minimum acceptable strength value, or to increase recall by decreasing strength. We selected threshold to be 0.5.

Figure 2 Bad Feature Vs Good Feature

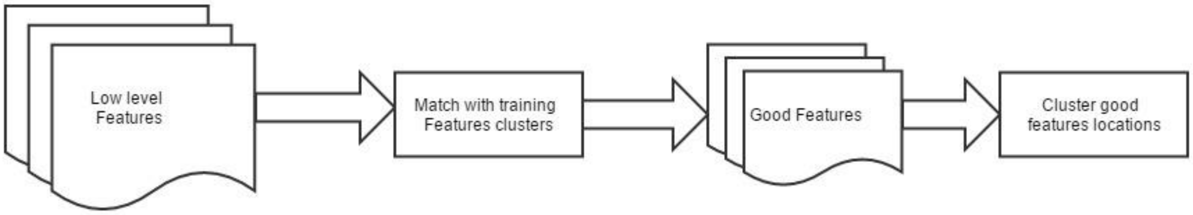


During training phase, we train the system with the average size of the class objects and use it during the mean shift clustering. Due to different resolutions and pixel size from image to image, the bandwidth is calculated from the projection information of the image read by the GDAL GIS library, this is very suitable to georeferenced projected raster images of remote sensing, which is the target of the application. So the training images and the testing images needs to be georeferenced.

Detection Phase

Feature extraction is executed on the testing image by the same algorithm used in the training phase, if the test image is too big, the image might be partitioned into smaller tiles of size 1000x1000 pixels, because large files will increase the complexity which need a lot of processing time and computing resources, and may crash the normal systems.

Figure 3 Detection procedure


Low level Features
Match with training Features clusters
Good Features
Cluster good features locations

The extracted features of the testing image is matched with the selected training features resulted from the training phase. Matching process generate a score of similarity between the two features on from the training and the other from the testing image.

The threshold of the accepted score to be considered a good match is calculated as David Lowe's paper, we use the K Nearest Neighbour Matching algorithm to find the best two matches for every feature in the testing image, if the score of the best match is higher than the second best match by more than a matching threshold 85% then this match considered good match, else it is considered a not matched feature. The matching threshold value can be decreased to increase precision and increasing the threshold will increase the recall.

After feature matching phase we got a set of good match features scattered throughout the image. Objects are dense regions in the feature space, separated by regions of lower feature density. The problem here is how to classify these features into classes represents detected objects, so we need to cluster these features to label each set of features as a candidate object without any information about how many objects in the image and if it exists altogether.

Our proposal to use density based clustering which can handle such a problem. The Clusters are dense regions in the data space, separated by regions of lower object density, we use Mean Shift clustering algorithm to cluster the sparse matched good key points into clusters representing the candidate objects, Mean shift algorithm is a non-parametric feature-space analysis technique for locating the maxima of a density function, a so-called mode-seeking algorithm. The physical size of the candidate object is controlled by the bandwidth parameter (B) passed to the mean shift algorithm to specify the maximum size of the class and accordingly the max size of the object. During training phase, we train the system with the average size of the class objects and use it during the mean shift clustering. Due to different resolutions and pixel size from image to image, we use the bandwidth calculated from the projection information of the image read by the GDAL GIS library.

We propose a score function of each detected object base on the radial basis kernel function as in Eq. 2.

$$Score(c) = \frac{1}{n}\sum_{i=1}^{n} e^{-\frac{\|x_i - x`\|^2}{2}} \qquad \text{Eq. 2}$$

Where n is the number of keypoints which are members of the detected object cluster, x_i is the keypoint i location, $x`$ is the cluster centre location.

Another post filter analysis of the results of the clustering is to filter weak classes by the number of the detected features in the class (n), Higher n increase precision and decrease recall, and vice versa.

Experimental results

We test our proposal on the aircraft object class. This proposal is concentrates on the remote sensing images with projection and georeferenced information, so all our researches and testing is done on a self-built dataset extracted from the World Imagery map service by Esri, This map service presents satellite imagery for the world and high-resolution imagery for the United States and other areas around the world , It provides one meter or better satellite and aerial imagery in many parts of the world and lower resolution satellite imagery worldwide. The map includes 15m TerraColor imagery at small and mid-scales (~1:591M down to ~1:72k) and 2.5m SPOT Imagery (~1:288k to ~1:72k) for the world. The map features 0.3m resolution imagery in the continental United States and parts of Western Europe from DigitalGlobe.

The main idea of the experiments is to compare the proposed procedure to detect objects locations vs the sliding windows method with various window overlap size. The experiments include using of SIFT and SURF local feature detector and descriptors, compare their results with various parameters of the model. We tested with several values of the number of clusters of the mid-level features. The following figures shows the Precision recall graphs and the Receiver operating characteristic (ROC) for the results of the experiments, which illustrate the comparison of the cases for using SIFT and SURF algorithms.

We found that the usage of the overlap factor affects the precision heavily because the overlap causes the single object is detected multiple times in the adjacent windows, the non-usage of overlap factor increases the precision but decrease the recall because objects may reside in the middle between two windows which results in a miss detection. In our model we decide to use no overlap, with a window size equal to the max size of the training objects, this solution risks the detection of small adjacent objects as a single object but this happened rarely and is acceptable. Table 1 shows the detection rate of the proposed method and the sliding window method with different overlap size.

Table 1 The recall and precession for different methods

	Proposed SURF Accuracy 85%	Proposed SIFT Accuracy 85%	S. Window 0% overlap	S. Window 30% overlap	S. Window 50% overlap
Recall	99.4%	99.8%	99.6%	100%	100%
Precession	8.4%	10.3%	7.8%	3.9%	2.1%
Echo	173.3%	124.4%	172.2%	455.6%	972%

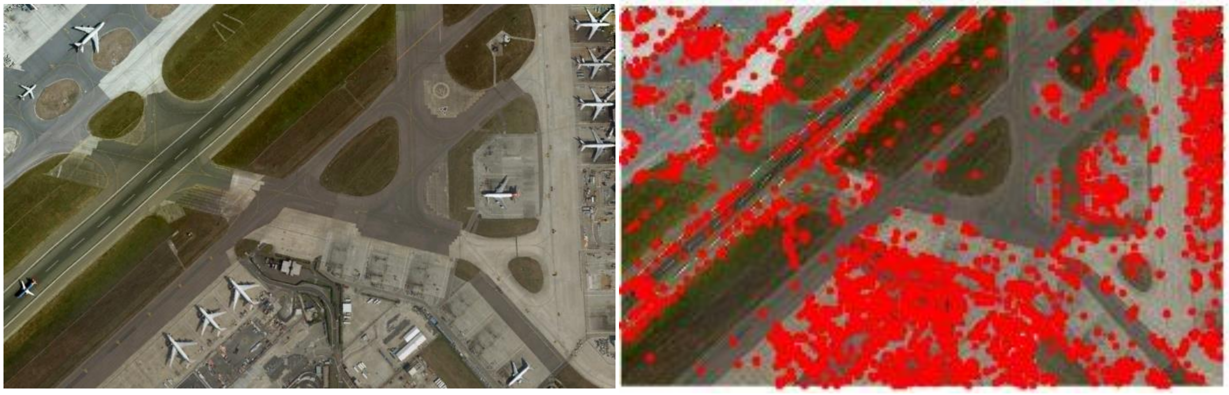
	S. Window 0% overlap 2 passes	S. Window 30% overlap 2 passes	S. Window 50% overlap 2 passes	S. Window 0% overlap 3 passes	S. Window 30% overlap 3 passes	S. Window 50% overlap 3 passes
Recall	100%	100%	100%	100%	100%	100%
Precession	3.8%	1.7%	0.9%	2.1%	3.9%	2.1%
Echo	366.2%	997.3%	2013.3%	583.9%	1652.4%	3121.2%

In Figure 6 the steps of the method of specifying the interesting area of candidate objects is shown. In (a) the raw image, (b) the red points represent the detected features by SIFT keypoint detector, (c) after the filtering of the detected features by matching it with the training positive features stored from training phase and cluster the locations of thesis features (The red circle represent the cluster centre), (d) After running the filtering on the cluster centres of areas of interest and show the most probable only with its matched features.

Conclusion

As it can be observed form the test results the method of detecting objects using positive keypoints matches location clustering is effective in localizing the small objects in big images like remote sensing images with acceptable accuracy, but with a lot of false positives. So it may be more effective to use it as a method to generate a list of object proposals to the later steps of machine learning classifiers. The SIFT algorithm as the feature detector and descriptor is performing a lot better than SURF in this method, which is not capable to catch the details in the small objects in the remote sensing images. “K-means ++” with minibatch technique as the clustering algorithm to build the mid-level feature clusters is the proposed successful method to cluster the low level features into mid-level feature clusters.

Figure 4 Steps of feature location clustering method to localize the interesting area of candidate objects



(a) (b)

(c) (d)

References

[1] W. Liu, F. Yamazaki, and T. T. Vu, “Automated vehicle extraction and speed determination from QuickBird satellite images,” IEEE J. sel. Topics Appl. Earth Observ. Remote Sens., vol. 4, no. 1, pp. 75-82, 2011.

[2] G. Cheng, L. Guo, T. Zhao, J. Han, H. Li, and J. Fang, “Automatic landslide detection from remote-sensing imagery using a scene classification method based on BoVW and pLSA,” Int. J. Remote Sens., vol. 34, no. 1, pp. 45-59, 2013.

[3] J. Han, P. Zhou, D. Zhang, G. Cheng, L. Guo, Z. Liu, S. Bu, and J. Wu, “Efficient, simultaneous detection of multi-class geospatial targets based on visual saliency modeling and discriminative learning of sparse coding,” ISPRS J. Photogramm. Remote Sens., vol. 89, pp. 37-48, 2014.

[4] X. Li, S. Zhang, X. Pan, P. Dale, and R. Cropp, “Straight road edge detection from high-resolution remote sensing images based on the ridgelet transform with the revised parallel-beam Radon transform,” Int. J. Remote Sens., vol. 31, no. 19, pp. 5041-5059, 2010.

[5] G. Liu, Y. Zhang, X. Zheng, X. Sun, K. Fu, and H. Wang, “A New Method on Inshore Ship Detection in High-Resolution Satellite Images Using Shape and Context Information,” IEEE Geosci. Remote Sens. Lett., vol. 11, no. 3, pp. 617-621, 2014.

[6] G. Cheng, J. Han, L. Guo, X. Qian, P. Zhou, X. Yao, and X. Hu, “Object detection in remote sensing imagery using a discriminatively trained mixture model,” ISPRS J. Photogramm. Remote Sens., vol. 85, pp. 32-43, 2013.

[7] J. Han, S. He, X. Qian, D. Wang, L. Guo, and T. Liu, “An object-oriented visual saliency detection framework based on sparse coding representations,” IEEE Trans. Circuits Syst. Video Technol., vol. 23, no. 12, pp.2009 -2021, 2013.

[8] J. Leitloff, S. Hinz, and U. Stilla, “Vehicle detection in very high resolution satellite images of city areas,” IEEE Trans. Geosci. Remote Sens., vol. 48, no. 7, pp. 2795-2806, 2010.

[9] X. Bai, H. Zhang, and J. Zhou, “VHR Object Detection Based on Structural Feature Extraction and Query Expansion,” IEEE Trans. Geosci. Remote Sens., vol. PP, no. 99, pp. 1-13, 2014.

[10] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” Int. J. Comput. Vis., vol. 60, no. 2, pp. 91-110, 2004.

[11] Y. Yang, and S. Newsam, “Geographic image retrieval using local invariant features,” IEEE Trans. Geosci. Remote Sens., vol. 51, no. 2, pp. 818-832, 2013.