Patch-based Natural Object Detection using CF*IRF

Wanjun Jin, Rongrong Wang, Lide Wu

Department of Computer Science and Engineering, Fudan University, Shanghai, P.R. China
{jwj, rrwang, ldwu}@fudan.edu.cn

Abstract
In this paper, we propose a patch-based approach for detecting natural objects on keyframes of video shots. We apply it on the extraction of semantic Feature "Vegetation" and "Animal", and on some search tasks in TRECVID2003. Our detection method is based on color and texture features, and considers the spatial information as well. TRECVID Evaluation shows that our approach works effectively and could deal with the special situation when target object only occupies a small portion of the whole image.

The main contribution of our approach is as follows: First, we devise a novel color weighting scheme which is named CF*IRF. Second, we use a patch-based detection method for Feature extraction task, and test it in an open large video corpus. Finally, spatial constraints of patches are defined in image tessellation, which provides more flexibility.

1. Introduction
In content-based image retrieval (CBIR), more and more researchers are looking beyond low-level color, texture and shape features in pursuit of more effective searching methods. For example, Naphade and Huang[1] use a list of semantic objects, including sky, snow, rock, water, and forest, in a factor graph-based framework for semantic indexing and retrieval of video. Smith and Li[2] assumed that a blue extended patch at the top of an image is likely to represent clear sky. More recently, Vailaya and Jain[3] presented an exemplar-based approach that uses a combination of color and texture features to classify sub-blocks (16*16 pixels) in an outdoor scene as sky or vegetation, assuming correct image orientation.

TRECVID 2003 is the third running of a TREC-style video retrieval evaluation, the goal of which remains to promote progress in content-based retrieval from digital video via open, metrics-based evaluation. The evaluation used a large video corpus. Well-segmented shots served as the predefined units of evaluation for the feature extraction tasks. (This year, 35067 shots for development and 32318 for test). One of the main tasks in TRECVID is Feature extraction, i.e. to automatically identify the occurrence of various semantic features which occur frequently in video information.

Some semantic features have close relationship with natural object detection. Since most of the semantic features can be extracted on keyframes in video shot. We exploit natural object detection on the keyframe of the video shot. The belief for the objects potentially present in an image eventually gives the score of the corresponding shot.

Since the video collection of TRECVID2003 is quite large, a good balance between speed and performance is really needed. In this scenario, most of the former pixel-based methods are not practical on the account of expensive time cost, so we develop a patch-based fast approach for detecting natural object. We utilize it in extracting Feature “vegetation” and Feature “animal”. The results of these natural object detectors are also helpful for extracting other features, e.g. “vegetation” detector can be used to improve the accuracy when extracting Feature “Outdoors”.

Our main contributions are as follows. First, A novel weighting scheme, called Color Frequency and Inverse Region Frequency (CF*IRF), is devised for evaluating color importance of certain natural object. Second, we use a patch-based detection method for Feature extraction task, and test it in an open large video corpus instead of a collected small image set. Finally, spatial constraints of patches are defined in image tessellation, which provides more flexibility.

2. Overview of our system
The structure of our natural object detection system is as follows. Basic natural object detector is first constructed by fusion of two classifiers. Color classifier is based on our proposed CF*IRF and texture classifier is based on Gabor feature. Then, we apply the basic detector in 8*6 image tessellation and get the patch belief map. Finally the analysis of patch belief map gives the possibility that the object potentially present in the image, which is also the ranking score of the corresponding video shot.
2.1 Basic natural object detector

In this section, we introduce the basic detector of our detection system. We specifically explain the color and texture features and how they are used for training the classifier.

2.1.1 CF*IRF. Enlightened by the idea of TF*IDF (Term Frequency and Inverse Document Frequency) weighting in text retrieval, we designed a CF*IRF (Color Frequency * Inverse Region Frequency) weighting scheme. It uses the region information to estimate the color importance of all positive regions (regions which contain target object). The basic assumption is that important colors should appear more times in the positive regions and fewer times in all the regions of the image.

For each color $C_i$, we define a measure of color frequency ($CF$), which reflects the extent it shows in positive regions. Intuitively, the larger the color frequency value, the more important this color is in representing the color feature of the object. The color frequency is defined in the following way.

$$CF(C_i) = \frac{|\{(x,y) | (x,y) \in \text{PositiveRegions}, color(x,y) = C_i\}|}{|\{(x,y) | (x,y) \in \text{PositiveRegions}\}|}$$  \hspace{1cm} (1)

where $color(x,y)$ is the color value at point $(x,y)$. On the other hand, a color becomes less important if it shows everywhere in the image. To reflect the distinguishing ability of a color, we define a measure of inverse region frequency ($IRF$) for color $C_i$:

$$IRF(C_i) = \log\left(\frac{|\{(x,y) | (x,y) \in \text{Image}\}|}{|\{(x,y) | (x,y) \in \text{Image}, color(x,y) = C_i\}|}\right)$$  \hspace{1cm} (2)

which is analogous to the IDF(inverse document frequency) in text retrieval.

Several things we have to pay attention to are:
1) Each color channel in RGB color space is quantized into 8 bins.
2) Any color with a very small $CF(C_i)$ will be omitted by setting $CF(C_i) = 0$.
3) Positive regions are segmented by hand and inputted into computer as a 0-1 mask. (As Fig.1 shows)
4) In the formula, Image means all training images, and PositiveRegions means all positive regions in all training images.

Based on the above preparations, we now come to the definition of the color importance.

$$CI(C_i) = \frac{CF(C_i) \cdot IRF(C_i)}{\sum_{i=1}^{n} CF(C_i) \cdot IRF(C_i)}$$  \hspace{1cm} (3)

Basically, the importance of a color is its color frequency weighted by the inverse image frequency, and normalized over all color such that the sum of all color importance weights is equal to 1.

2.1.2 Texture features. Texture classifier is trained by the Gabor feature of training samples. We choose 4 scales, 6 orientations for the Gabor filter bank. Prototype-based classifier is chosen as the classifier, since we find that other classifiers, such as SVM, are not appropriate for such an occasion.

After we browsed many keyframes in annotated development video set, we find that there are various types of “vegetation” in the news video. Some trees and vegetables in outdoor scene are high-textured, but some
artificial turfs in sports news are smooth and less textured. They are so various that we cannot find appropriate positive and negative samples to train the classifier like SVM.

We try to collect typical samples under different illumination to be our prototypes. Assuming we have collected k prototypes, they can be converted to points in texture space, named \( T_1, T_2, \ldots, T_k \). For a new patch, we first convolute it with Gabor filter bank and convert to a point \( T_{\text{patch}} \) in texture space. Then we define minimum distance:

\[
MINDIS(T_{\text{patch}}) = \min \{ ||T_{\text{patch}} - T_i||_i \}, i = 1, 2, \ldots, k
\]

(5)

If \( MINDIS(T) \) is smaller than a threshold \( \theta_2 \), we regard the patch as the positive patch. Here we choose \( k=5 \). Five prototypes of “Vegetation” are collected and shown in the figure below.

Fig.2. Texture prototypes of “Vegetation”

This distance-based scheme is more like a Query-by-example retrieval problem. We exploit it just to avoid the trouble to collect negative samples. Experimental results show this scheme is useful on this occasion.

2.2 Detection stage

In this section, we address the detection stage of our system. At the first step, keyframe is extracted on the basis of shot boundary segmentation. We take the middle frame of each shot as its key frame. And further natural object detection is performed on those key frames.

Second, color and texture classifier introduced above are exploited in 8x6 image tessellation. The size of each patch in tessellation is 44x44 for ABC and CNN news, and 44x32 for C-SPAN news. (As Fig. 3(a) shows)

Fig.3. (a) The image grid (b) Positive patch map

The features of each patch are fed to the color classifier and texture classifier. Two values, Number of positive patch (NPP) and sum of average color importance (SACI), are acquired by fusion of the outputs of color classifier and texture classifier. The binary positive-patch map of Fig.3(a) is shown by square with different color in Fig.3(b)

\[
NPP = \sum_{\text{patch/image}} (ACI(\text{patch}) > \theta_1) \& \& (MINDIS(T_{\text{patch}}) < \theta_2)
\]

(6)

\[
SACI = \sum_{\text{patch/image}} ACI(\text{patch})
\]

(7)

Finally, a weighted sum of NPP and SACI gives the score of the keyframe, which is used to rank the video shots.

\[
SCORE = NPP + \beta \times SACI
\]

(8)

In practice, initial values of three threshold \( \theta_1, \theta_2, \beta \) are selected empirically. But \( \theta_1 \) and \( \theta_2 \) can change dynamically. i.e. If a patch gets a high score from the color classifier, the texture restriction upon it could become looser, and verse visa.

2.3 Spatial constraints based on image tessellation

Generally, the belief for the objects potentially present in an image is in proportion to the total number of positive patches in it. It is true for the position-free feature, such as “vegetation”. But some features have spatial bias. They often show up in some place while seldom show up in other place, e.g. sky often shows at the top of an image. In this case, imposing some spatial constraints on the positive patches is quite necessary. We consider two kinds of spatial constraint, the spatial distribution of patches and the inter-relationship between patches.

For the first constraint, instead of a position classifier used by Vailaya [1], we use a distribution map to represent the distribution of positive patches in the image. The distribution map achieved from the development data can be used as the weight of patch when detecting in test data.

A possibly more appropriate case is to learn the narrower concept, such as some search tasks in TRECVID 2003. Look at the Fig.4, which shows how distribution map works in search topic 0102 (Find shots from behind the pitcher in a baseball game as he throws a ball that the batter swings at).

Fig.4. One example of pitch scene and the distribution map of vegetation patches in the pitch scene.

Besides this, we also consider the inter-relationship between different types of patches. This kind of spatial restraint is important especially when positive patches are constellated but could appear anywhere in the image. Luo et.al. develop a probabilistic spatial context model to represent that inter-relationship between patches [4]. while we develop a patch correlogram to represent that inter-relationship between patches. More details about our scheme about spatial constraint could be found in [5]. In TRECVID2003, we just use a simple version of this
scheme when detecting feature “animal”. After constructing patch belief map of “animal”, we update it by increasing the belief value of a positive patch if it is adjacent to other positive patches.

3. Experiments and evaluations

We apply it on the extraction of semantic Feature "Vegetation" and "Animal" and on some search tasks in TRECVID2003. The average time cost for one keyframe is 0.85s at a Pentium3 866Hz PC. The evaluation results given by NIST are shown in Fig.5 and Fig. 6(a), where pink bar represents the hits of our best run and red bar represents the hits of run using this approach. This year, not using multi-modal may deteriorate the performance of all our runs.

![Fig.5. (a) Hits at depth 100 for feature “vegetation” (b) Hits at depth 2000 for feature “vegetation”](image)

![Fig.6. (a) Hits at depth 100 for feature “animal” (b) Hits at depth 100 for feature “animal” with our three systems.](image)

From the evaluation results, we could see our proposed approach performs well for feature “vegetation”. Some hits and false alarms are shown in Fig.7 and Fig.9. By its nature, our approach could also deal with the special situation when vegetation only occupies a small portion of the whole image. (Fig. 8) Although the evaluation only shows the hits at depth up to 2000, we believe our approach could perform better at deeper depth. As to feature “animal”, the performance of this approach is not satisfying. Although the detection performance is not bad (Fig. 10), there are too many false alarms. Although most runs in TRECVID2003 don’t have satisfying results for this feature, we could see that it is wiser to find the environment where animal often appears than to find the animal itself, and ASR information is quite useful here. (Fig. 6(b))

4. Conclusion

In this paper, we propose a patch-based approach for detecting natural objects on keyframes of video shots. We apply it on the extraction of semantic Feature "Vegetation" and "Animal" and on some search tasks in TRECVID2003. Our detection method is based on color and texture features, and considers the spatial information as well. TRECVID Evaluation shows that our approach works effectively and could deal with the special situation when target object only occupies a small portion of the whole image.

5. References