Real-time Cascade Template Matching for Object Instance Detection

Chengli Xie[1], Jianguo Li[2], Tao Wang[2],
Jinqiao Wang[1], Hanqing Lu[1],

[1] National Laboratory of Pattern Recognition,
Institute of Automation,Chinese Academy of Sciences
100190 Beijing, China
{clxie, jqwang, luhq}@nlpr.ia.ac.cn

[2] Intel Labs China
{jianguo.li, tao.wang}@intel.com

Abstract. Object instance detection finds where a specific object instance is in an image or a video frame. It is a variation of object detection, but distinguished on two points. First, object detection focused on a category of object, while object instance detection focused on a specific object. For instance, object detection may work to find where toothpaste is in an image, while object instance detection will work on finding and locating a specific brand of toothpaste, such as Colgate toothpaste. Second, object instance detection tasks usually have much fewer (positive) samples in training compared to that of object detection. Therefore, traditional object instance detection methods are mostly based on template matching.

This paper presents a cascade template matching framework for object instance detection. Specially, we propose a 3-stage heterogeneous cascade template matching method. The first stage employs dominate orientation template (DOT) for scale and rotation invariant filtering. The second stage is based on local ternary patterns (LTP) to further filter with texture information. The third stage trained a classifier on appearance feature (PCA) to further reduce false-alarms. The cascade template matching (CTM) can provide very low false-alarm-rate comparing to traditional template matching based methods and SIFT matching based methods. We demonstrate the effectiveness of the proposed method on several instance detection tasks on YouTube videos.

Keywords: cascade, template matching, object instance, detection

1 Introduction

Object detection is a hot-topic in the field of computer vision. There are a lot of researches, such as [1, 5~7, 10]. However, with the advance in camera phone and mobile computing, people want to not only detect objects in images anywhere they took, but also go beyond to the identity of the object. That is to say, they want to know what it is specifically of the detected objects. For instance, when we walk on

street and find one beautiful handbag, but we do not know what brand and type is it and where to buy it. This case does not belong to object detection. Thus, a new research topic arises, namely object instance detection, which becomes more and more important in nowadays mobile computing scenario. There are some typical applications in smart phones, such as Google's goggle.

Concretely, object instance detection aims at finding and locating specific instances of objects, such as Colgate toothpaste boxes, Coca-cola cans, or Canon DLSR camera, in an image and video frames. It is quite different to existing techniques such as object detection, object category recognition, image retrieval.

Given an example of handbag recognition, these four techniques have different goals. Object detection tries to find and locate general handbags in images. Object instance detection attempts to find and locate a specific handbag (for instance a specific type of Louis Vuitton handbag) in images. Object category recognition just wants to know whether the image contains a handbag or not. And in image retrieval, people just find images in a large database which has similar global appearance to the given query handbag image.

In this paper, we propose a 3-stage cascade template matching framework to object instance detection as shown in Figure 1. Specially, the first stage employs DOT [1] for fast, scale and rotation invariant non-target filtering. The second stage is based on local ternary patterns (LTP) to further filter non-target with texture information. The third stage trained a neural-network (MLP) classifier on appearance feature (PCA) to further reduce false-alarms. The 3-stage cascade is heterogeneous.

The contributions of this work could be summarized as follows:

(1) We present a cascade template matching framework for object instance detection.
(2) Through effective combination the DOT, LTP and PCA-MLP matches, the cascade-template-matching not only provides high hit-rate and low false alarms, but also yields real-time processing speed.
(3) The approach is scale and rotation invariant to the object instance.

In the reminder of this paper, we first discuss some related works on instances detection. And then our approach is represented in details. Following that, comparisons between ours and the state-of-art ones are shown with some discussion and analysis. And finally, the whole work is concluded.

2 Related Works

There exist two typical types of works in object instance detection: first is template matching based methods; second is the learning based methods.

In template matching, SIFT [9] features were widely used. SIFT describes scale and rotation invariant local features in image, and has been applied to object instance recognition. However, SIFT cannot locate more than two instances in one query image. Besides, SIFT is a bit slowly in computing. SURF[11] is proposed to replace SIFT features with fast processing speed.

Triplet of feature descriptors [3] is proposed for detection and recognition. These triplets are labeled via modified K-means clustering algorithm, which is followed by inverse lookup matching and triplet votes.

DOT (dominant orientation templates) [1] is proposed for texture less instances detection or tracking, which assumes a simple and slow motion environment. While in object instance detection, scale and lighting may change greatly, scenes in adjacent frames may transit quickly or suddenly. And occlusion is not the major limitation. In the case of general videos, to take advertisement clip as an example, the scene and content change largely and sometimes quickly. Thus, one can't assume continuity of object or near constant of background.

Mustafa Ozuysal [8] proposed FERNS, which belong to the learning based method. It formulates feature point recognition in object instance detection in a Naïve Bayesian classification framework through non-hierarchical point-pair feature groups (these groups are called ferns). This yields simple, scalability and efficient results in terms of number of classes. From the point of feature selection, ferns can be regards as some kind of variety of random forest.

In some old works, object instance detection can be divided into two steps. First is detecting object class in test images or frames. Second is recognizing instances of object for the detection results.

Nevertheless, the proposed solution attempts to deal with the problem in a whole, with the cascade template matching framework.

3 Casade Template Matching

This section we will describe the cascade template matching framework (CTM) in details. And explain how they are built and trained at real-time speed.

The CTM approach consists of the DOT stage, the pyramid LTP stage and the PCA-MLP stage. It takes gradient information, texture information and appearance features into consideration, separately.

3.1 Framework overview

For the purpose of robust and rapidly discovering specified instances of object class, coarse to fine framework is adopted (Figure 1). First stage makes use of multi-angle multi-scale gradient template to extract candidates based on a variation of DOT [1]. Then a local texture descriptor, histogram of pyramid LTP, is calculated to further filter false positive samples from first stage. Finally, machine learning method (currently used multi-layer perception neuron network) is employed to pick positive from negatives in the results of stage 2. Therefore, accurate detection results (category instances) are found effectively.

Given an input image or a video frame, CTM proceeds in three steps:

(1) A merged version of the DOT detector [1] was applied: For all candidate windows C_i (not using tracking information), we use 3-step merge algorithms to preliminary reduce some overlap ones.

(2) Pyramid LTP template matching is utilized to quickly filter false alarms.

(3) PCA-MLP layer as the last stage tells the most difficult negative samples from positives, with the combination of MLP classifiers and PCA features.

Note that the proposed method is not restricted to specific descriptors or classifiers in each layer. Taking the 3rd stage as an example, MLP can be replaced with SVM or other classifiers.

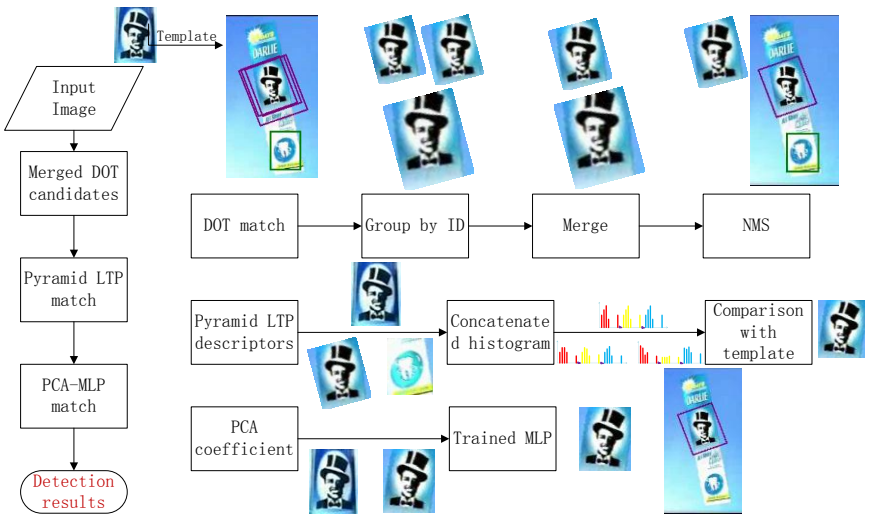

Figure 1. Framework Overview.

3.2. Merged DOT based template matching

In the original, only maximum response for each template is accepted, or ESM tracking lib to filter, which is rather slow. In this paper, we apply a 3-step merge algorithm to DOT candidates.

1) Grouping candidates from the same template by overlap ratio and filter out noise (groups with candidates less than a group threshold T_g). Here, N_c stands for number of candidates. C_{ic} means the ic^{th} candidate. And function Group(C_{ic}, C_{jc}) puts the ic^{th} and jc^{th} candidates into one merge pool. Similarly, N_g stands for number of groups. While function Filter_out_group(G_{ig}) deletes the ig^{th} group.

```
Step1: group
  For ic = 0:Nc-1
      For jc = i+1:Nc
       If not_in_one_group(Cic,Cjc)
            If is_overlap_enough(Cic,Cjc)
              Group(Cic, Cjc);
  For ig = 0:Ng-1
  If num_of_group(Gig)< Tg
       Filter_out_group(Gig);
  Update(Ng);
```

2) Merging rectangles in each group G_i by average with Laplace estimation, if the number is more than the merge threshold T_m. In this section, function num_of(G_{ig}) computes the number of candidates in ig^{th} group. And MC_{ig} stands for the ig^{th} merged candidate.

```
Step 2: merge
  For ig = 0:Ng-1
      num_ig  = num_of( Gig);
      Gig.confidence /= num_ig;
      If num_ig > Tm
  MCig.coor=(sum(Gig.cand.coor)*2+num_ig)/(2*num_ig);
```

3) Suppressing non-maximum rectangles of different templates at same location of current video frame. Function is_overlap_enough(MC_{im}, MC_{in}) computes the overlap area of the im^{th} and the in^{th} merged candidates. While function Amensalism(a,b) means we mark a and b as exclusion, and discard one by some criterion.

```
Step 3: non-max suppress
  Nc = num_of(MC);
  For im = 0:Nc;
        For in = 0:Nc
        If is_overlap (MCim, MCin) > Threshold_overlap
            Amensalism(MCim, MCin);
```

Note that, the original DOT suppresses the non-maximum responses of the same template in order to reduce false alarm. (Or it use ESM lib to track these candidates, which slows down the process). While in the 3-step merging algorithm, non-maximum responses belonging to different templates at same location are suppressed, so that missing rate can be cut down at this stage and false positive rate can be maintained at a low level by following stages.

3.3. Pyramid LTP histogram match

LBP [4] is a general used texture descriptor, while sensitive to light changes. LTP [2] is a generalization of LBP, which split local ternary pattern into positive and negative LBP parts. LTP is much robust to illumination by adding a threshold to comparison center pixel with its neighbors.

To reflect differences in different resolution, we propose pyramid LTP. Given a normalized candidate, we calculate LTP at 3 different resolutions, and then concatenate them into a histogram (Figure 2) after a normalization operation to each histogram separately.

After histogram calculation and normalization, the distance D(H1, H2) between histograms of candidate and template is compared with corresponding threshold T_h. Candidates passed the T_h pass the second stage.

Though there are several methods to calculate distance between two histograms, such as correlation, chi-square (CHISQR) and histogram intersection. It seems that CHISQR performs similar with histogram intersection, while better than correlation.

Please refer to section 4.2 for details. D(H1, H2) with CHISQR is adopted in this work.

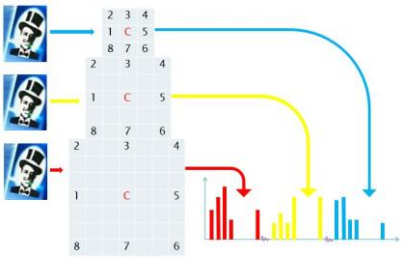

Figure 2. Pyramid LTP descriptor

3.4. PCA-MLP match

After processed by the first two stages, the left false positives are hard to distinguish from true positives. Therefore, non-linear classifier and appearance features are adopted to further remove false from truth.

In this paper, we adopted PCA features to reduce the dimension of appearance feature, and adopted multiple layer perception (MLP) as the non-linear classifier.

Before extracting the PCA (principal component analysis) feature, candidates reach this stage are first scale-rotation normalized with respect to template, according to match index within previous stage. As computing appearance descriptor is a bit time cost, it is hoped that the number of input is as small as possible.

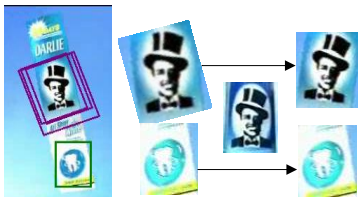

Figure 3. Scale and orientation normalization of merged-candidates with respect to template.

4 Experiment Validation

4.1 Experiment dataset

We crawled the YouTube web site and gather 2 types of videos, containing toothpaste category with several different instances, and Canon DLSR camera EOS D7. Please refer to below table for details. Each video clip contains around 750 frames (about 30 seconds at 25fps). The training and testing number of video clips varies as some of

the classes have more frames and some are less. We just split them equally into training and testing sets.

There are 2 Darlie instances, since we would like to see whether the cascade template matching framework can discriminate different instances in the same object class. (In this experiment, we would see the differentiation of instance Darlie-2 and other toothpaste instances, including Darlie-1).

Table 1. Training and test data amount (in number of video clips)

Category	Toothpaste					Canon EOS camera
Subclass	Colgate	Darlie-1	Darlie-2	Aquafresh	Crest	
Train	10	8	9	8	10	6
Test	10	7	8	8	10	5
Template image						

Figure 4 below are overall detection results. Thin red and green windows are first and second stages outputs. And bold blue boxes denote the final results. It is clear that, the cascade template matching framework can filter false alarms while maintaining true positives.

Our approach can also deal with slight occlusion; see the middle two of Crest at row 5. And due to pyramid LTP, influences by illuminate changes are minimized (last example of Crest at row 5).

In Figure 4, Darlie-2 at row 3 could find other instances of Darlie class at the cost of accuracy down by decreasing the threshold (last two at row 3), or just can't find other kinds of instances (e.g. Darlie-1 at the third column of this row). In other words, our approach has the ability to distinguish between different instances of same object category.

For Canon camera, it shows that cascade template matching is suitable to not only flat object, but also to 3D instances.

Besides, our approach takes 29ms on the average for each frame on Core2Duo 2.8G, 4G RAM, which satisfy the real-time application.

4.1 Performance at each single stage

Figure 5 shows performances of each single stage: merged DOT stage, the pyramid LTP stage, and the PCA-MLP stage. Merged DOT can be considered as original DOT though the performance improves. From the ROC curves, we can see that every stage is useful since different information is used: gradient information, texture information and color appearance information.

Theoretically, the product of these 3 stages reflects the capability of the whole cascade. The experimental results of the whole CTM cascade will be shown at section 4.3.

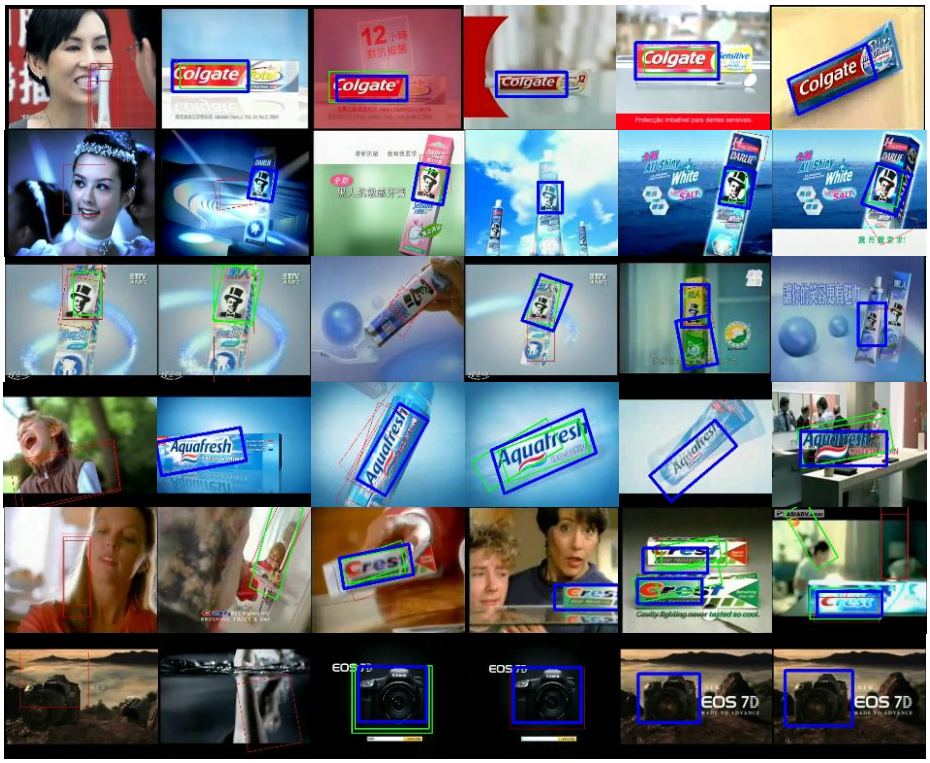

Figure 4. Some detection results, where thin red window indicates first stage candidate, green window stands for the pyramid stage output, and bold blue denotes the final result. Each line is for an object instance. Specifically, they are, from top to bottom, Colgate toothpaste, Darlie-1 toothpaste, Darlie-2 toothpaste, Aquafresh toothpaste, Crest toothpaste and Cannon EOS camera.

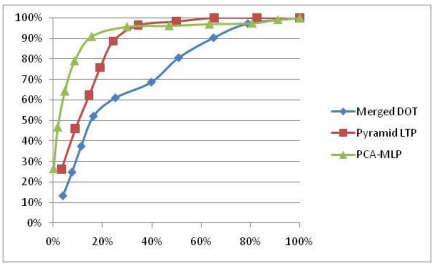

Figure 5. Single stage performances comparison.

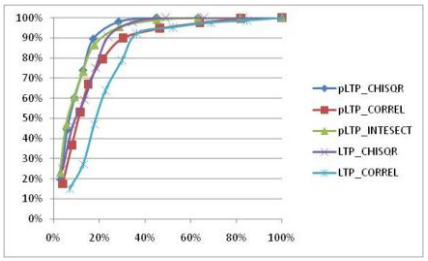

Figure 6. Pyramid LTP versus LTP with different histogram comparison schemes

4.2 Different LTP schemes

In this experiment, we compare performance of pyramid LTP (pLTP) and LTP with different histogram compute methods mentioned in section 3.3. Figure 6 shows the results of comparison.

It is obvious that for both pyramid LTP and LTP, CHISQR works better than correlation. And pyramid LTP (pLTP) with CHISQR outperforms other schemes.

4.3 Cascade Template Matching v.s. SIFT and DOT

In this part, we compared CTM with original DOT and SIFT. Table 2 shows the total true responses and false alarms on the whole test set (48 video clips).

As the first stage in CTM, our merged DOT is with a lower threshold than original to avoid filtering true responses. Another benefit is that, lower threshold results in more candidates, while more candidates means both more positive and negative training examples for following LTP and PCA-MLP stages.

As the videos are crawled from YouTube, there is no label information of object instance at all. Therefore, we just record the total true responses and false alarms of each method for comparison. True response means the output window just covers most of the right object instance. Others are considered as false alarms. Responses to partial instance samples (occluded more than 25% by others, or truncated more than 25% by image boundary, etc.) are not counted.

Table 2. Cascade templates matching versus SIFT & DOT

Methods	True responses	False alarms
SIFT	2627	631
CTM	2698	66
DOT	2013	2105

It is obviously, our approach can reduce false positives significantly; meanwhile keep a relative high detection rate.

Moreover, CTM and DOT have the ability to detect two or more instances in one image (e.g. the last but one at row 5 in Figure 4).

On another side, SIFT takes more than one second for a frame on average, much slower than ours' 29 milliseconds.

5 Conclusions

In this paper, we raise the problem of object instance detection, and propose a cascade template matching framework (CTM) for object instance detection. The framework consists of 3 heterogeneous stages: merged DOT, pyramid LTP and PCA-MLP, which utilize gradient information, pyramid texture histogram and color appearance

information separately. Experiments show that CTM yields fast and accurate instance detection on some sets of YouTube videos.

Acknowledgement

Jinqiao Wang and Hanqing Lu are supported by the National Natural Science Foundation of China (Grant No. 60833006 and 60905008), and 973 Program (Project No. 2010CB327905).

References

1. Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Pascal Fua and Nassir Navab: Dominant Orientation Templates for Real-Time Detection of Texture-Less Objects. CVPR, 2010.
2. Xiaoyang Tan and Bill Triggs: Enhanced Local Texture Feature Sets for Face Recognition under Difficult Lighting Conditions. IEEE Transactions on Image Processing, 2010
3. C. Lawrence Zitnick, Jie Sun, Richard Szeliski and Simon Winder: Object Instance Recognition using Triplets of Feature Symbols. Microsoft Research, Technical Report, 2007-53.
4. Heikkila M., Pietikainen M and Schmid C: Description of Interest Regions with Local Binary Patterns. Pattern Recognition, 42(3): 425-436, 2009.
5. Juergen Gall and Victor Lempitsky: Class-specific Hough Forests for Object Detection. CVPR 2009.
6. Christoph H. Lampert, Matthew B. Blaschko and Thomas Hofmann: Beyond Sliding Windows: Object Localization by Efficient Subwindow Search. CVPR 2008.
7. Subhransu Maji, Alexander C. Berg: Max-Margin Additive Classifiers for Detection. ICCV 2009.
8. Mustafa Ozuysal, Pascal Fua and Vincent Lepetit: Fast Keypoint Recognition in Ten Lines of Code. CVPR 2007.
9. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. Int'l J. of Computer Vision 60 (2004) 91-110.
10. P. Felzenszwalb, R. Girshick, D. McAllester and D. Ramanan: Object Detection with Discriminatively Trained Part Based Models. PAMI Vol.32, No.9, September 2010.
11. Herbert Bay, Tinne Tuytelaars and Luc Van Gool: Surf: Speeded up Robust Features. In Proceedings of the ninth European Conference on Computer Vision, 2006.