Oxford-IIIT TRECVID 2010 – Notebook Paper

Mayank Juneja, Siddhartha Chandra, Omkar M. Parkhi, C. V. Jawahar
Center for Visual Information Technology,
International Institute of Information Technology, Gachibowli, Hyderabad, India
Andrea Vedaldi, Marcin Marszalek, Andrew Zisserman
Visual Geometry Group,
Department of Engineering Science, University of Oxford, United Kingdom

Abstract

Our team participated in the "light" version of the semantic indexing task. All runs used a combination of an image-level dense visual words classifier and an object-level part based detector. For each of the ten features, these two methods were ranked based on their performance on a validation set and associated to successive runs by decreasing performance (we also used a number of different techniques to recombine the scores). The two methods yielded a significantly different performance depending on the feature, as expected by their design: The χ^2-SVM can be used for all feature types, including scene-like features such as Cityscape, Nighttime, Singing, but is outperformed by the object detector for object-like features, such as Boat or ship, Bus, and Person riding a bicycle.

Our team did not participate in the collaborative annotation effort. Instead, annotations were carried out internally for all the ten features to control quality and keyframe extraction, and to obtain region-of-interest annotations to train the object detectors. Compared to last year, the image-level classifier was significantly faster due to the use of a fast dense SIFT feature extractor and of an explicit feature map to approximate the χ^2 kernel SVM.

1 Introduction

Our team participated in the "light" version of the semantic indexing task. We extracted our own keyframes for every shot of both the TRECVID 2010 DEVEL and TEST data sets. The DEVEL set was subdivided into two halves denoted TRAIN and VAL and used for training and validation, respectively. This subdivision respects movie boundaries to guarantee the statistical independence of the keyframes in the two subsets. Ground truth labels for the DEVEL keyframes were obtained by an internal team of annotators. New developments this year include: Complete annotation of all the keyframes of the DEVEL set (Sect. 2); Fast dense SIFT descriptor computation and PEGASOS with explicit feature map for the image-level classifier (Sect. 3.1); discriminatively trained part based model for the object-level classifier (Sect. 3.2).

2 Annotations

Annotations were carried out (only for the DEVEL subset) at the frame level for each of the ten features. For some of the selected object-like features, Region of Interest (ROI) annotations were also carried out. After obtaining the first set of ground truth labels, multiple rounds of refinement were carried out to remove the errors in the annotation.

The refinement of annotations of the VAL set was carried out by using a weak classifier as follows :

1. Train a classifier on the TRAIN set.

2. Re-rank all the images in the VAL set based on the classifier output.

3. Refine the annotations of the top 1000 ranked frames and the bottom 1000 ranked frames.

Similarly the refinement of annotations of the TRAIN set was done by using a classifier trained on the VAL set.

3 Method

Our classification scheme combines three ideas: (i) For scene-like categories. we use a non-linear χ^2 SVM based on an approximated feature map and trained by using stochastic gradient (PEGASOS, Sect. 3.1); (ii) For object-like categories, we have used the discriminatively trained part model described in Sect. 3.2; (iii) We then combine the classification scores of (i) and (ii) by a number of different techniques (Sect. 3.3).

3.1 Pegasos SVM with Explicit χ^2 kernel Feature Map

The image-level classifier is a non-linear SVM on top of a bag of dense visual words (Sect. 3.1.1). To train a large-scale SVM efficiently we use [5] (as implemented in the VLFeat library [6]). While PEGASOS is a linear SVM solver, we use the explicit feature map for χ^2 kernel [7] to extend it efficiently to use a χ^2 (non-linear) kernel. The whole setup is fast and efficient compared to traditional SVM techniques that do not use the feature map idea. For example, on our framework training a SVM using 100K frames requires only 2 minutes and classifying 100K frames requires only 1 minute on an Intel Xeon CPU clocked at 1.86 GHz.

3.1.1 Feature Descriptors

We used Pyramid Histogram of Visual Words [1] to represent an image. The Pyramid Histogram of Visual Words (PHOW) descriptors consist of visual words which are computed on a dense grid. Here visual words are vector quantized SIFT descriptors [4] which capture the local spatial distribution of gradients.

Local appearance is captured by the visual words distribution. SIFT descriptors are computed at points on a regular grid with spacing M pixels. We have used gray level representations for each image. At each grid point, the descriptors are computed over circular support patches with radii r. Thus, each point is represented by four SIFT descriptors. These dense features are vector quantized into visual words using K-means clustering. Here, we have used a vocabulary of 1000 words. Each image is now represented by a histogram of these visual words occurrences.

We have used $M = 5$, $K = 1000$ and radii $r = 10, 15, 20, 25$. To deal with the empty patches, we zero all the SIFT descriptors with L2 norm below a threshold (200).

In order to capture the spatial layout representation, which is inspired by the pyramid representation of Lazebnik et. al. [3] , an image is tiled into regions at multiple resolutions. A histogram of visual words is then computed for each image sub-region at each resolution level.

To summarize, the representation of an appearance descriptor for an image is a concatenation of the histograms of different levels into a single vector which are referred to as Pyramid Histogram of Visual Words (PHOW). We have used two levels for the pyramid representation. The distance between the two PHOW descriptors reflects the extent to which the images contain similar appearance and the extent to which the appearances correspond in their spatial layout.

3.1.2 Results

This method worked well for most of the scene like categories, (e.g. *Nighttime*, *Cityscape*, *Singing*, *Playing Instrument*, *Demonstration or Protest*). Results obtained on the TEST set for different features are shown in Fig. 1 for Airplane_Flying, Fig. 2 for Cityscape, Fig. 3 for Demonstration or Protest, and Fig. 4 for Nighttime. Quantitative results are reported in Table 1.

Category	**AP**	**xinfAP**
Training Set **Testing Set**	**TRAIN** **VAL**	**TRAIN+VAL** **TEST**
Cityscape	0.49	0.11
Nighttime	0.34	0.12
Demonstration Or Protest	0.27	0.07
Airplane	0.21	0.13
Boat Ship	0.21	0.15
Singing	0.17	0.05

Table 1: **Performance of the SVM image classifier.** The table reports the average precision of the method of 3.1 when trained on TRAIN and evaluated on VAL, and TRECVID extended inferred AP when trained on TRAIN+VAL. To compute average precision on TRAIN+VAL the complete and cleaned annotations were used. In several cases the difference in AP and xinfAP is very large, suggesting either that there is a significant statistical difference between the DEVEL and TEST data subsets, or that the accuracy of the xinfAP estimate is poor (xinfAP may still be adequate to rank different methods).

3.2 Object Detection with Discriminatively Trained Part Based Models

A category detector was trained for object-like features (Bus, Boat-Ship and Airplane-Flying).

3.2.1 Model

We used the part-based detector of [2], summarized next. Object detection is inherently a difficult task because of the significant variability in photometry, viewpoint and the intra-class variability. This model uses a sliding window to detect objects. HOG features in 8×8 pixel blocks are computed over the image. The model is a mixture of deformable part models, one for each object aspect, where each mixture component has a global template (a root filter with coarse resolution) and deformable parts (part filters with finer resolution). The global template is called the root filter which has a coarse resolution and the deformable parts are called the part filters which have a finer resolution. The root filter is denoted as F_0 and the n part models are denotes (F_i, v_i, d_i) and include a part filter F_i, an fixed offset v_i and an the parameters of an elastic deformation d_i.

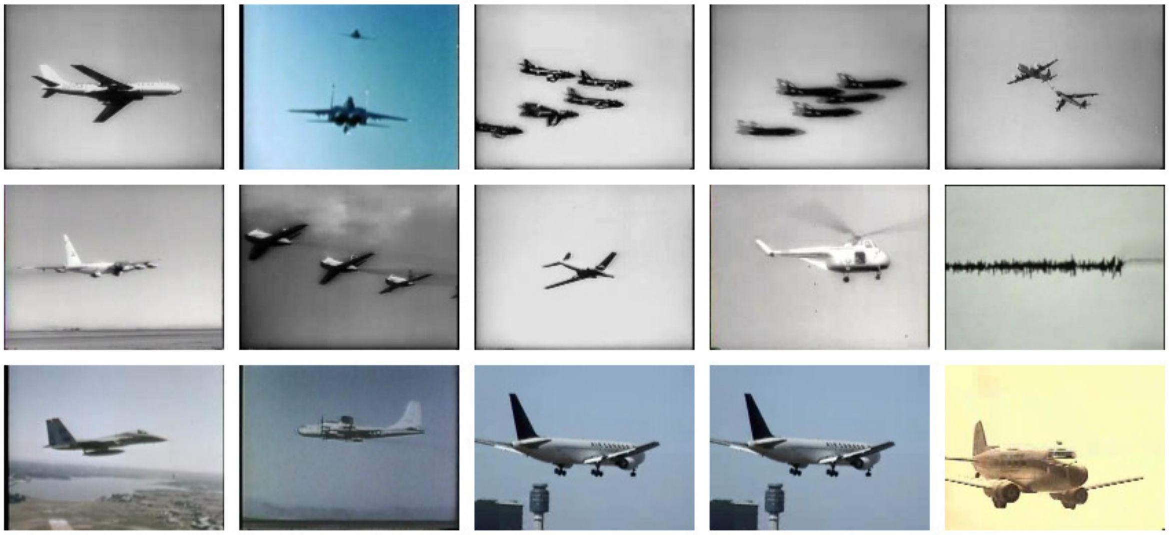

Figure 1: Top 15 shots from the TEST set (shown by keyframes) using the scene classifier for Airplane_Flying category.

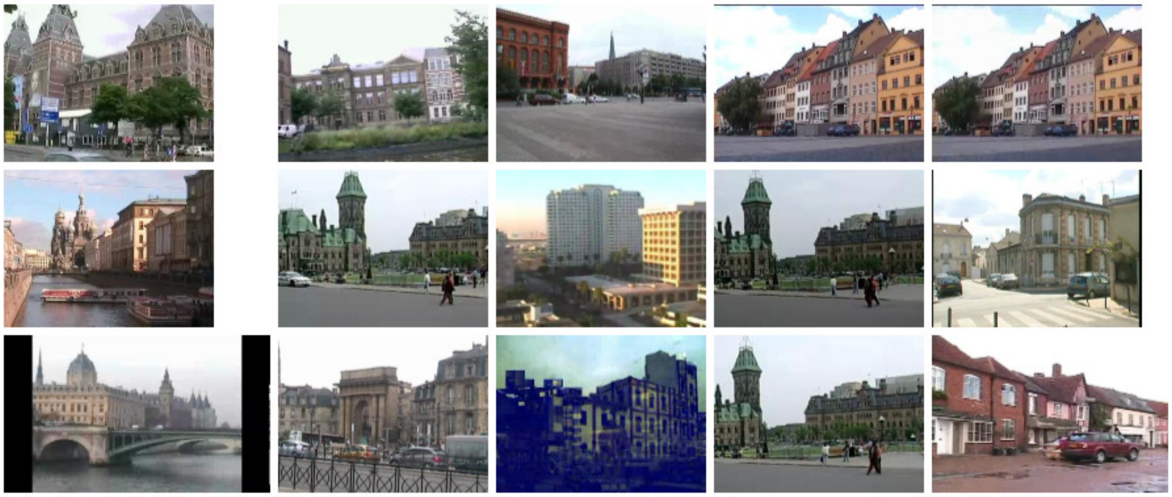

Figure 2: Top 15 shots from the TEST set (shown by keyframes) using the scene classifier for Cityscape category.

The model score at a certain image location is obtained as the score of the root and part filters minus the cost of the displacement of the parts (deformation):

$$\text{score}(p_0, ..., p_n) = \sum_{i=0}^{n} F_i \cdot \phi(H, p_i) - \sum_{i=0}^{n} d_i \cdot ({dx_i}^2, {dy_i}^2)$$

Here $p_0, \ldots, p_1$ denote the root and part locations, dx_i, dy_i denote the part displacements relative to their "neutral" location $p_0 + v_i$. $\Psi(H, p_i)$ denote the HOG descriptor extracted from a given part at a certain image location.

An overall score is computed for each root location p_0 by finding the best placement of parts:

$$\text{score}(p_0) = \max_{p_1, \ldots, p_n} \text{score}(p_0, ..., p_n)$$

3.2.2 Results

The model described above yields candidate bounding boxes for each image along with a confidence score (see for example Fig. 7). Since our objective is classification, we let the score of an image to be the maximum score of any window detected in it, and use this score for classification.

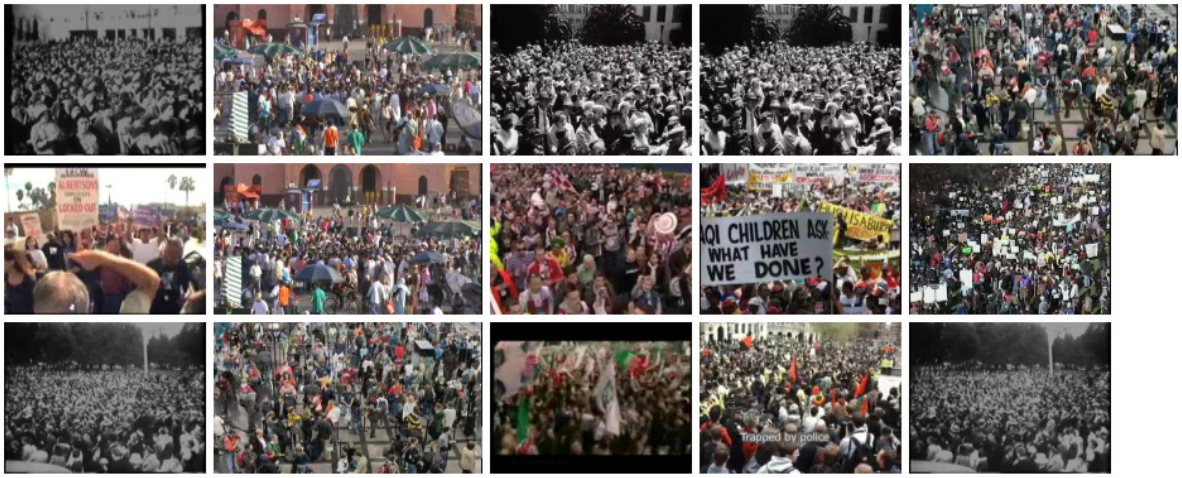

Figure 3: Top 15 shots from the TEST set (shown by keyframes) using the scene classifier for Demonstration_or_Protest category.

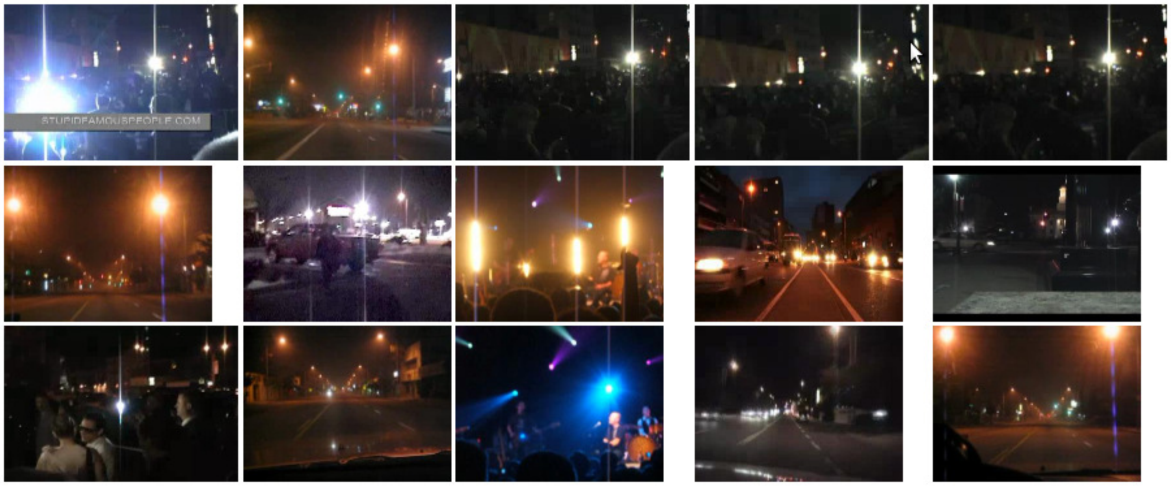

Figure 4: Top 15 shots from the TEST set (shown by keyframes) using the scene classifier for Nighttime category.

3.3 Combining classification and detection

As described above, given the detector output, we assign the classification score of an image to be the maximum score of any window detected in it. At this point, both our detection results and classification results are lists of $\langle \text{image}, \text{score} \rangle$ pairs. The two scores were combined by taking either (i) the maximum, (ii) a linear combination, or (iii) borda count.

Acknowledgment We are grateful to the UK-India Education and Research Initiative (UKIERI) for financial support, and to ERC grant VisRec.

References

[1] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *International Conference on Computer Vision*, 2007.

[2] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Discriminatively trained deformable part models, release 4. `http://people.cs.uchicago.edu/~pff/latent-release4/`.

[3] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Conference on Computer Vision and Pattern Recognition*, June, 2006.

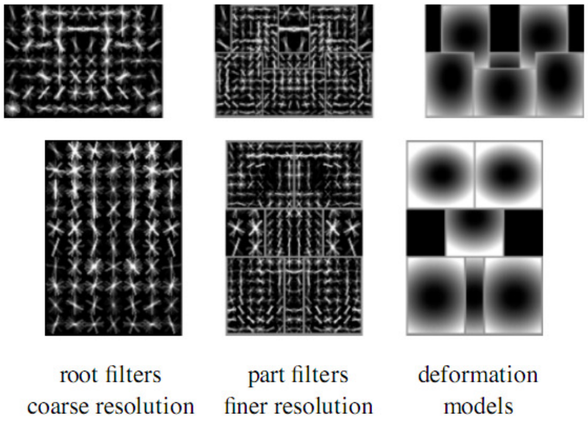

Figure 5: Bicycle Model

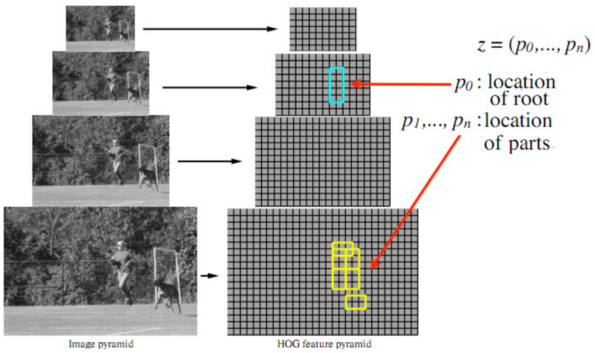

Figure 6: Object Hypothesis

[4] D. Lowe. Distinctive image features from scale invariant keypoints. *International Journal of Computer Vision, 60(2):91-110*, 2004.

[5] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *Proc. of the International Conference on Machine Learning*, 2007.

[6] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

[7] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010.

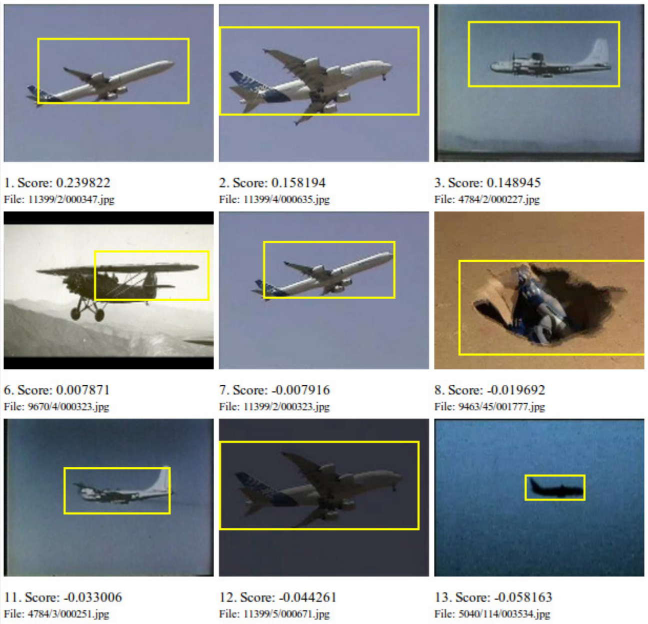

Figure 7: *Detections on Trecvid Test Data for category Airplane_flying*