Discriminatively-trained model mixture for object detection in aerial images

Hicham Randrianarivo*, Bertrand Le Saux*, Michel Crucianu† and Marin Ferecatu†
*Onera The French Aerospace Lab
F-91761 Palaiseau, France
† Conservatoire National des Arts et Métiers, Laboratoire CEDRIC
292 Rue St Martin FR-75141 Paris Cedex 03

***Abstract*—In this work we propose a new method for vehicle detection in very high resolution aerial images. Our model is based on a mixture of filters which capture the visual appearance of the object of interest. Each filter is discriminatively trained in order to model the implicit subcategories in the training dataset. We use an iterative hard-negative mining procedure to focus the detector on difficult samples. We assess our approach on several large datasets and show it tackles efficiently major problems in remote sensing such as orientation change and data size.**

I. Introduction and related works

Automatic analysis of aerial images is an important research topic with direct application to many high-level tasks such as scene interpretation or autonomous visual navigation. A necessary step to achieve this is to be able to localize and recognize the objects in an image. Object detection in aerial scene has some specificities compared to natural scenes as in the Pascal VOC challenge[1]. Images taken from an aerial aircraft map very large areas which means images are very large, usually greater than 3000×3000, and contain thousands of objects. Another particularity is the view of the objects: they are seen from the top, vary in orientation and can be occluded.

Up to now, the resolution of available aerial images was too low to use efficiently popular methods from computer vision widely used for object detection [2], [3]. However, recently more and more very high resolution datasets (greater than 10cm/pixels) of aerial images have been becoming available. With the availability of such very high resolution aerial images, object detection methods are becoming increasingly demanded. In [4] the authors use the Deformable Parts-based Model (DPM) of [2] to classify urban areas in the image. [5] use discriminative autoencoders to learn a representation of the small targets in aerial images (mainly vehicles) and detect them. [6] perform vehicles detection from an Unmanned Aerial Vehicle (UAV) view flying at low altitude.

In this paper we propose an efficient object detector for aerial images based on the modified Histogram of Oriented Gradient (HOG) feature of [2] and the modeling of subcategories [7], [8]. Our framework for object detection in aerial images is based on a mixture of discriminatively trained templates. It models automatically subcategories that can exist in an object category and a template is then learned for each of the subcategories. Figure 1 shows an example of a mixture of templates for car category. The final detector is the mixture of the templates of each subcategory, called Discriminatively trained Model Mixture (**DtMM**) in the following.

Fig. 1: On the left: a representative example from the subcategory. On the right: the HOG filter learned using our algorithm.

II. Discriminatively trained object detector

We propose an object detector framework based on two main steps. First we propose to disambiguate the subcategories that exists in an object category using a clustering method. Second we trained for each subcategory a template filter based on the popular HOG feature using a hard mining procedure.

A. Subcategory modeling

We employ a discriminative classifier to train a filter which will be used as a template to find object in an image. The learning process implies to gather positive and negative examples corresponding to a specific category from a dataset. Categories can contain thousands of objects and finding a global category representation is quite challenging. We propose a method to model the variation of appearance in a category.

In fact, trying to model a whole category of objects with a single rigid template is a very hard task. Objects in the dataset can vary in orientation, appearance or scale. We tackle this issue using an unsupervised framework to split the training samples into several clusters wich are visually alike as shown in fig. 2. Moreover training a discriminative template on each cluster instead of all the training samples results in an easier classification problem.

Several criteria can be used to find visually homogeneous clusters. The vehicles in aerial images are very similar but their color, orientation and size can vary a lot. The criterion we use to model the subcategories of the cars is the aspect-ratio. The main benefits of this criterion is to model the *orientation* of the training samples in the dataset. Training several clusters instead of rotating the template to match every orientation of the object allows to find several sizes for the sliding window wich fit the object of interest. The size of

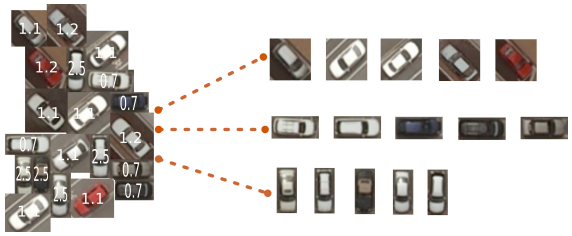

Fig. 2: For each sample from the dataset, we compute the aspect-ratio then we use GMM in order to cluster the samples.

the sliding window and more importantly the aspect-ratio of the window is critical to match the template with an object in the image. We use the logarithm on the aspect-ratio as a feature to roughly estimate the orientation of the vehicle in the image (horizontal, vertical or diagonal). Then we learn a Gaussian Mixture Model (GMM) using this feature and cluster the training samples using the GMM. For each subcategory we compute the size of the template based on the median size of the samples in the cluster.

B. Filter training

We use template matching to perform object detection in the images. For this task the template must be robust to changes of appearance and orientation of the objects. The templates are trained using annotated samples from a dataset. The objects are described using the popular HOG feature of [2] which is a modified version of [3] that includes texture. This feature vector offers a great combination of discriminative power and fast computing [9]. In order to ensure the robustness of the detector, the template are trained using hard negative mining. We first train a template from the annotation data and randomly sampled negative examples. We perform detection on the training set to extract the areas classified as the targeted object and train again the template using the miss-classified negative samples (*hard negatives*) in addition to the previous training set. We repeat this operation until the detection score on the validation test converges. The details of the training algorithm are shown in Algorithm 1.

C. Detection

We handle the object detection task as a template matching problem. We perform the computation in the HOG feature space to fasten the detection of the objects. A property of the HOG is that it can be seen as a $2D$-spatial filter so instead of extracting patches one by one from the image and computing the feature individually for each patch, we transform the whole image in its HOG representation. Performing the sliding window in the HOG space instead of the RGB space reduces the spatial extent of the image but each location is now described by a powerful descriptor instead of the RGB value of the pixels. We use the precise template trained on each subcategory to produce a heatmap. The hot points of the heatmap are the areas of the image where the detector estimates that there are objects. We produce the heatmap using

Algorithm 1 Training of the mixture of filters of orientated gradients

1: Selection of positifs examples in annotations $\longrightarrow$ $\{(O_i, y_i = +1)\}$
2: Cluster $\{(O_i, y_i = +1)\}$ with mixture of Gaussian models $\longrightarrow$ sub-categories $S^{(k)} = \{(O_i^{(k)}, y_i = +1)\}$
3: **for all** cateroy $S^{(k)}$: **do**
4: Compute the size of the filter (median of the heights and widths in $S^{(k)}$)
5: Select random negatifs samples from the dataset with overlap smaller than 50% with positifs samples $\{(O_i, y_i = +1)\})$ $\longrightarrow \{(O_i, y_i = -1)\}$
6: Resize $\{(O_i, y_i = +1)\}$ to the size of the filter
7: Transform $\{(O_i^{(k)}, y_i)\}$ with HOG feature $\longrightarrow \{(x_i^{(k)}, y_i)\}$
8: Train the SVM classifier on $H^0 = \{(x_i^{(k)}, y_i = +1)\} \cup \{(x_i, y_i = -1)\}$ $\longrightarrow \beta^{(k)}$: the HOG filters model the subcategory $S^{(k)}$
9: **for hard-mining** $\forall m \leq M$: **do**
10: Classifier les images par $f^{(k)}(x) = \beta^{(k)} \cdot x$ pour extraire de nouveaux exemples ngatifs "difficiles" $\longrightarrow H^{m+1} = \{(x_i', y_i' = -1)\} \cup H^m$
11: Classify the images with $f^{(k)}(x) = \beta^{(k)} \cdot x$ in order to find the new "hard samples" $\longrightarrow H^{m+1} = \{(x_i', y_i' = -1)\} \cup H^m$
12: Re-train on H^{m+1}
13: **end for**
14: **end for**
15: Out: $\longrightarrow$ mixture of models $\{\beta^{(k)}\}$

the fast Normalized Cross-Correlation (NCC) score from [10] defined as:

$$S_d(o) = \frac{\sum_{x',y'} t(x',y') I_h(x+x', y+y')}{\sqrt{\sum_{x',y'} t(x',y')^2 I_h(x+x', y+y')^2}} \quad (1)$$

The NCC return a correlation score between the template and a position of the sliding window in the image. The heatmaps of correlations can be compared between the templates in the mixture. The final result of detection is the fusion of the heatmaps of each template. We fuse the heatmap by using the Non-Maximum Suppression (NMS) algorithm describe in [11]. For a set of overlapping windows extracted from all the heatmaps, the algorithm keeps only the window with the maximal score, the one with the greatest confidence.

This procedure drastically reduces the number of candidate windows and increases the precision of the detector. Figure 3 shows the heatmaps produced using the NCC alongside with the original image for each of the templates in the mixture. The *scale* issue is handled by resizing the test image before applying the HOG transformation.

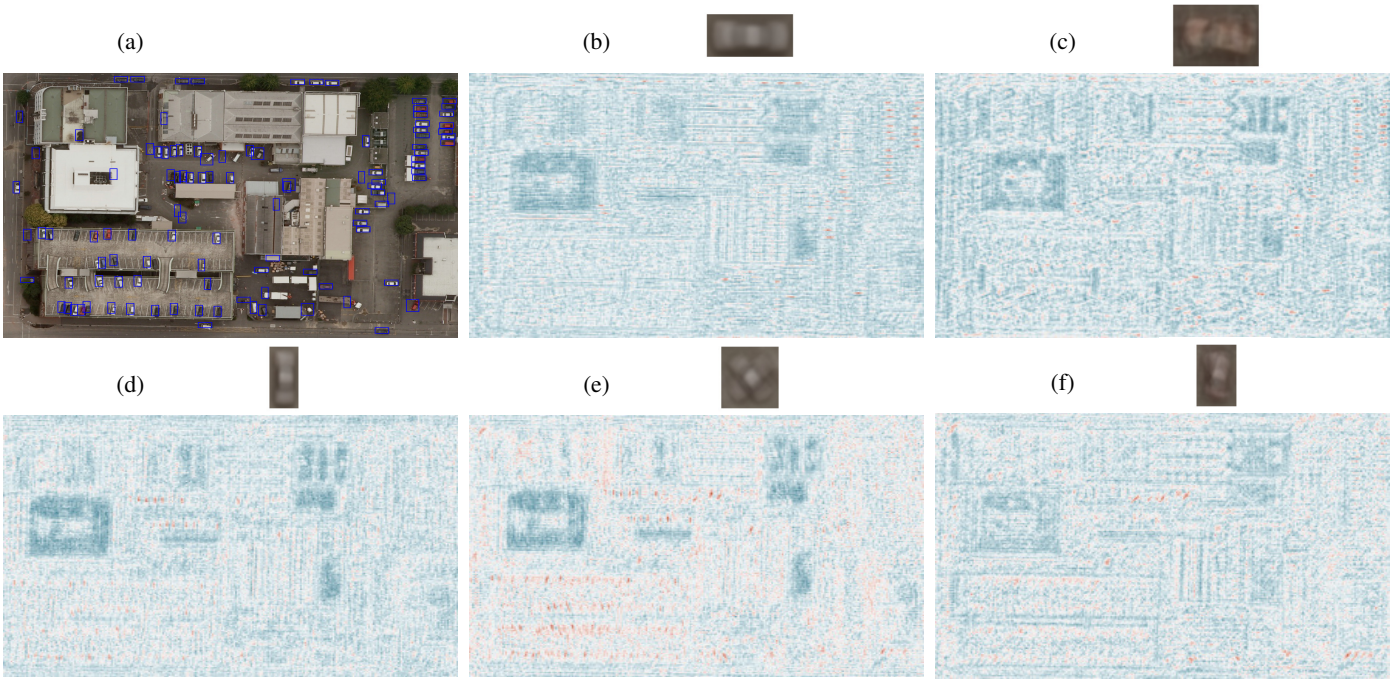

Fig. 3: (a) shows final detection on the Christchurch dataset area, while (b) to (f) are heatmaps produced by correlation of each subcategory template with the image. 5 clusters were found to optimally model the "car" class: (b) and (c) are templates for cars with horizontal orientation, (d) is the vertical car template and (e) and (f) are diagonal templates with different angles. Heatmaps are then combined by NMS. This avoids a bad impact from templates which produce a lots of false alarms such as (e). On the top of each heatmap we show the mean representative of the subcategory

III. Experiments

We test the object detector on two different datasets: one provided by the New Zealand Aerial Mapping[1] and the other provided by the IEEE Geoscience and Remote Sensing Society Data Fusion Contest 2015, called Zeebrugge in the following. The images have different resolutions (10 cm/pixel for the Christchurch dataset and 5 cm/pixel for the Zeebrugge dataset).

A. Christchurch dataset

The Christchurch Dataset[2] consists of 4 images whose resolution is 10cm/pixels and size is around 5000×4000 pixels. 2 images are used for the training set, 1 for the validation set (1703 cars) and 1 for the test set (654 cars).

We test our method in the context of object detection, especially we are interested in car detection in large image. For this task we must define a measure of good detection. We consider an hypothesis bounding box for a car as a good detection if its overlap score is over 0.5. The overlap score is defined by:

$$\frac{area(A \cup B)}{area(A \cap B)} \tag{2}$$

If several hypothesis have an overlap with an object greater than 0.5 we only keep the hypothesis with the greatest score and the other are all marked as false positive. Figure 4 shows the precision-recall curves obtained with the **DtMM** method. At the time being we do not provide a method to automatically find the good number of models to maximize the average-precision score. In order to find that number of models we vary manually the number of cluster in our method.

We show the results on an area of Christchurch in fig. 3(a). We vary the number of clusters in the model then we perform detection on the test dataset. We compute the precision-recall curves for each detectors to find the optimal number of clusters for this dataset. Our framework is compared with the object detector proposed by [3].

B. Zeebrugge Dataset

The Zeebrugge dataset consists of 7 images the resolution of which is 5cm/pixels and size is 10000×10000 pixels. We use a subset of 3 images to train the methods and 2 images to test. We compare the **DtMM** method with several methods in the context of pixel-wise classification. Two metrics are used to evaluate the classifiers, the $f1$-score and the area under the precision-recall curve (AUC). We compare our method with the methods tested in [12]:

[1] http://nzam.co.nz

[2] Images are freely available on http://www.linz.govt.nz/land/maps/ Source: Land Information New Zealand (LINZ) and licensed by LINZ for re-use under the Creative Commons Attribution 3.0 New Zealand licence. Annotations are available on demand

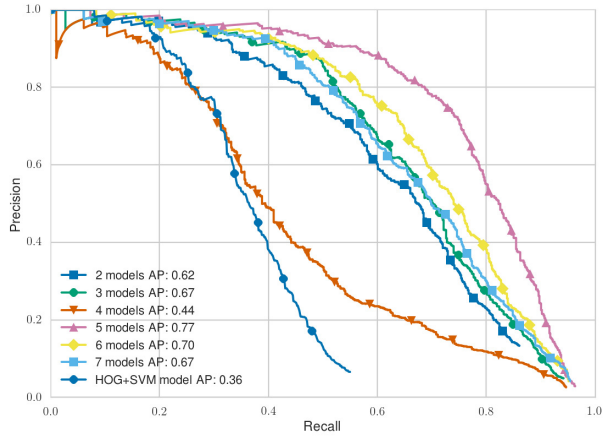

Fig. 4: Precision-recall curves for **DtMM** method. Contrary to intuition, the average-precision score do not increase with the number of model, for example 3 models is better than 4 models and the best results are achieved using 5 models.

- RGB/SVM: A RBF Support Vector Machines (SVM) is trained on the superpixels [13] extracted from the image
- HOG32/SVM: Patches of size 32×32 are indexed by HOGs then a RBF SVM is trained on the patches
- RGB VGG/SVM: Features are learned on patches of size 231×231 using the fast network [14] of 8 layers (including 5 convolutional layers) cut at layer #7. A SVM is trained on the output feature.
- Overfeat/SVM: Same as before, except we use the overfeat implementation of 6 convolutional layers network and no drop-out.

The results are presented in table I. The results show that for objects with well defined boundaries, approaches able to model the entire object (and not only the pixels or a small patch information) outperform state of the art methods. Even in the context of pixel-wise classification, taking into account the entire object leads to a performance gain.

TABLE I: Pixel-wise classification on the Zeebrugge dataset.

Method	$f1$-score	AUC
RGB+SVM	24.02	13.44
HOG32+SVM	30.24	19.73
VGG+SVM	31.46	25.71
Overfeat+SVM	36.03	30.6
DtMM	**48.46**	**35.29**

We show an example of the bounding boxes generated with our method for vehicle detection in fig. 5.

IV. Conclusion

We presented a framework for vehicle detection in very high resolution images using a mixture of discriminatively trained models. Each model in the mixture corresponds to a visual subset of the cars in the training dataset. Each model is trained using the HOG feature with hard mining of negative examples. This approach shows great results in difficult urban areas and can be used at multiple scales.

Fig. 5: Detection of vehicles on a neighboorhood of Zeebrugge. The detector is able to find cars all around the roundabout at various orientations.

Acknowledgements

This work was partially supported by funding from ANR CONTINT project POEME, France.

References

[1] M. Everingham, L. Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, sep 2010.

[2] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester, "Discriminatively Trained Deformable Part Models, Release 5."

[3] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Computer Vision and Pattern Recognition*, 2005.

[4] H. Randrianarivo, B. Le Saux, and M. Ferecatu, "Urban structure detection with deformable part-based models," in *International Geoscience and Remote Sensing Symposium*, 2013.

[5] S. Razakarivony and F. Jurie, "Discriminative Autoencoders for Small Targets Detection," in *IAPR International Conference on Pattern Recognition*, pp. 3528–3533, Ieee, aug 2014.

[6] J. Gleason, A. V. Nefian, X. Bouyssounousse, T. Fong, and G. Bebis, "Vehicle detection from aerial imagery," *International Conference on Robotics and Automation*, pp. 2065–2070, may 2011.

[7] S. Divvala, A. Efros, and M. Hebert, "How important are "Deformable Parts" in the Deformable Parts Model?," in *European Conference on Computer Vision*, 2012.

[8] C. Gu, P. Arbeláez, Y. Lin, K. Yu, and J. Malik, "Multi-component models for object detection," in *European Conference on Computer Vision*, 2012.

[9] H. Bristow and S. Lucey, "Why do linear SVMs trained on HOG features perform so well?," tech. rep., 2014.

[10] J. P. Lewis, "Fast Normalized Cross-Correlation," *Vision Interface*, vol. 1995, no. 1, pp. 1–7, 1995.

[11] A. Neubeck and L. Van Gool, "Efficient Non-Maximum Suppression," *International Conference on Pattern Recognition*, vol. 3, no. 1, pp. 850–855, 2006.

[12] A. Lagrange, B. Le Saux, A. Beaupere, A. Boulch, A. Chan-Hon-Tong, S. Herbin, H. Randrianarivo, and M. Ferecatu, "Benchmarking classification of earth-observation data: from learning explicit features to convolutional networks," in *IEEE International Geoscience and Remote Sensing Symposium*, 2015.

[13] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient Graph-Based Image Segmentation," *International Journal of Computer Vision*, vol. 59, pp. 167–181, sep 2004.

[14] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the Devil in the Details: Delving Deep into Convolutional Nets," *British Machine Vision Conference*, pp. 1–11, 2014.