

# CLASSIFIER OPTIMIZATION FOR MULTIMEDIA SEMANTIC CONCEPT DETECTION

Sheng Gao and Qibin Sun

Institute for Infocomm Research (I<sup>2</sup>R), A-STAR, Singapore, 119613

{gaosheng, qibin}@i2r.a-star.edu.sg

## ABSTRACT

*In this paper, we present an AUC (i.e., the Area Under the Curve of Receiver Operating Characteristics (ROC)) maximization based learning algorithm to design the classifier for maximizing the ranking performance. The proposed approach trains the classifier by directly maximizing an objective function approximating the empirical AUC metric. Then the gradient descent based method is applied to estimate the parameter set of the classifier. Two specific classifiers, i.e. LDF (linear discriminant function) and GMM (Gaussian mixture model), and their corresponding learning algorithms are detailed. We evaluate the proposed algorithms on the development set of TRECVID'05<sup>1</sup> for semantic concept detection task. We compare the ranking performances with other classifiers trained using the ML (maximum likelihood) or other error minimization methods such as SVM. The results of our proposed algorithm outperform ML and SVM on all concepts in terms of its significant improvements on the AUC or AP (average precision) values. We therefore argue that for semantic concept detection, where ranking performance is much interested than the classification error, the AUC maximization based classifiers are preferred.*

## 1. INTRODUCTION

The task of the high-level feature extraction or semantic visual concept detection in TREC video information retrieval (TRECVID) is to train a system to retrieve the top-N (e.g. 2,000) video shots relevant to the given semantic concept. Its efficiency of the system is measured by the ranking performance metric such as non-interpolated average precision (AP) or the area under the ROC curve (AUC). It is different from the classification problem where the classification error is a measurement of the system performance. Thus learning algorithms for error minimization will not be optimal for the detection task. In the paper, we study the learning algorithm for designing classifier optimized for ranking.

The traditional approaches for semantic concept detection are to train a binary classifier by minimizing the penalized classification error (e.g. SVM) or maximum likelihood (ML) (e.g. Gaussian mixture model (GMM)) [1, 2, 9]. The classifier here is served as scoring and sorting the video shot documents. Therefore the ranking function should rank the relevant shots as high as possible. However, the traditional algorithms are not designed for ranking, because the mismatch between the training and the evaluation affects the final ranking performance. It becomes worse when the dataset is heavily imbalanced. Cortes & Mohri studied the relation between the ratio of the class distribution and the classification error rate and the average AUC in [5]. Their results showed that the average AUC coincides with the accuracy only for the even class distribution. With improving accuracy, the average AUC monotonically increases and its standard deviation decreases. The analysis gives a reasonable explanation for the partial success of using the error minimized based classifier for ranking. For example,

SVM trained by penalized error minimization is widely and successfully exploited for semantic concept detection in TRECVID<sup>1</sup>. Nevertheless, they also proved that the uneven class distributions have the highest variance of AUC. It implies the classifier with a fixed classification error will show the noticeable difference of AUC for highly uneven class distribution, which is the situation frequently occurred in information retrieval such as semantic concept detection. A new learning criterion is therefore expected, to suit for ranking rather than minimizing classification error.

One of the available criterions is the AUC. It characterizes the correct rank statistics for a given ranking function and is measured by the normalized Wilcoxon-Mann-Whitney (WMW) statistic [6]. The AUC based objective function is a function of the scores output from the classifier. So the classifier parameters have already been naturally embedded in the AUC. Maximizing AUC will lead to an optimal configuration of those parameters. Note that like all other metrics such as classification error, recall, precision, or  $F_1$  measure, the AUC is a discrete metric of correct ranking of pair-wise positive-negative samples. It has to be smoothed and differentiated before optimisation.

The AP value, a measure of the area under the precision-recall curve, is closely related to the AUC. Both are two different views of the overall performance of the ranking system. Thus, the classifier optimized for AUC should be also optimal for AP. Later we will experimentally demonstrate their interesting relation.

In this paper we present a learning algorithm to estimate the classifier for AUC optimization. First we approximate the empirical AUC on the training set using the smoothing functions. Then the gradient descent based algorithm is applied to find its optimal configuration. The details of the learning algorithms are presented for two specific classifiers, i.e. linear discriminant function (LDF) and GMM. We evaluate it on the development set used for semantic concept detection in TRECVID 2005.

In the next section, we present the definition of the objective function for rank optimization and the learning algorithm. In Section 3, we discuss its application to design two specific classifiers used in our experiments. The experimental results and analyses are given in Section 4. Finally, we summarize our contributions in Section 5.

## 2. LEARNING CLASSIFIER FOR AUC OPTIMISATION

Learning the classifier for ROC optimization has been studied in [3-6]. In [5], a theoretical analysis of the relations between AUC optimization and error minimization is demonstrated and then RankBoost is applied to learn a set of weaker classifiers for minimizing the pair-wise ranking error. In [3, 4, 6], the classifier is trained by directly maximizing the AUC based objective function using the gradient descent method. In our previous work [8], the metric-oriented objective function is defined for the preferred metric optimization such as  $F_1$  measure. This work is a further extension of [8] and [3, 4, 6] targeting at solving semantic concept detection.

First we give a formal definition for our task and notations used throughout the paper. Given a set of training samples,  $T$ ,

<sup>1</sup> <http://www-nlpir.nist.gov/projects/tv2005/tv2005.html>

with  $M$  positive samples and  $N$  negative ones, learn a binary classifier,  $f(X|\Lambda)$  with the parameter set  $\Lambda$ , so that the values of the function for positive samples are higher than those of the negatives. We denote  $X$  as a representation of the sample,  $X^+$  for the positive and  $X^-$  for the negative.  $S^+$  and  $S^-$  are the values of  $f(X^+|\Lambda)$  and  $f(X^-|\Lambda)$ , respectively, given the known parameter set.

## 2.1 AUC Metric

The AUC is a one-dimensional metric for the quality of the ROC curve, and is a measurement of capability of the classifier for ranking. It is formally defined as the probability of the positive sample ranking higher than the negative, i.e.,

$$U = P(S^+ > S^-) \quad (1),$$

where  $S^+$  is the positive score from the classifier or ranking function and  $S^-$  negative score. In real applications, it is empirically estimated from the training set as,

$$U = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N I(S_i^+, S_j^-) \quad (2).$$

$S_k^c = f(X_k^c|\Lambda)$ ,  $c = \{+, -\}$ , for the  $k$ -th sample,  $X_k^c$ , of class  $c$ .  $\Lambda$  is the parameter set of the classifier.  $I(S_i^+, S_j^-)$  is an indicator function. It is set to one when ranking is correct, i.e.  $S_i^+ > S_j^-$ . Otherwise, it is zero.

We would like to find a configuration of the classifier parameters so that the maximal value in Eq. (2) is obtained.

## 2.2 Approximating AUC

The score values in the indicator function are calculated from the classifier. So it is dependent on the classifier configuration. The indicator function bridges the classifier parameters with the AUC value. Clearly, Eq. (2) is a natural embedding of the classifier parameters. Maximizing Eq. (2) will result in a classifier with optimal ranking. However, Eq. (2) is discrete and non-differential. It must be smoothed before optimisation.

When we use a smoothing function to approximate the indicator function, the overall AUC will be smoothed. One popular way is to use the sigmoid function. It has been shown successfully in the past work [6, 8]. Here we discuss two smoothing functions, sigmoid function and Gaussian-like function, used in our experiments.

### Case 1: Sigmoid function

The sigmoid function for approximating the indicator function is defined as,

$$I(S_i^+, S_j^-) = 1 / (1 + \exp(-\alpha \cdot z_{ij})) \quad (3),$$

with  $\alpha$  being a positive constant to control the window size [8] and  $z_{ij} = S_i^+ - S_j^-$ . For any pair-wise positive and negative sample, Eq. (3) is close to 1 if the positive is ranked much higher than the negative. Otherwise, it closes to zero. When the rank of the positive sample is close to the negative, it closes to 0.5. Soon after we will see (See sub-section 2.3) that the pair-wise samples, whose value is close to 0.5, play an important role in the learning.

### Case 2: Gaussian-like function

The second case has a form as,

$$I(S_i^+, S_j^-) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\infty} \exp(-x^2/2\sigma^2) \cdot \int_{-\infty}^{x+z_{ij}} \frac{1}{\sqrt{2\pi}\sigma} \exp(-y^2/2\sigma^2) dy dx \quad (4),$$

with a positive constant value,  $\sigma$ , to control the smoothing window size. Its gradient over  $z_{ij}$  has an analytic form (See sub-section 2.3). Although Eq. (4) has no analytic form, it does not affect the learning algorithm used in the paper.

With the above approximations for the indicator function, we get the following objective function by substituting them into Eq. (2).

$$U = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N I(S_i^+, S_j^-) \quad (5)$$

## 2.3 Parameter Estimation

The objective function in Eq. (5) is only the function of the classifier parameter. In general, the function is highly non-linear. We solve the function using the gradient descent algorithm. Thus the solution is locally optimal and the starting point will affect it. In our experiments we first estimate the parameter set using the traditional algorithms (e.g. *ML*) and then select the parameters as the initial point for the iterative algorithm.

The gradient of Eq. (5) with respect to  $\Lambda$  is as,

$$\nabla U|_{\Lambda} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N \nabla I(S_i^+, S_j^-)|_{z_{ij}} \cdot (\nabla f(X_i^+|\Lambda)|_{\Lambda} - \nabla f(X_j^-|\Lambda)|_{\Lambda}) \quad (6)$$

The first term in the summary is the gradient with respect to  $z_{ij}$ , a variable of difference scores between the positive sample and the negative. Its form depends on the chosen smoothing function. And the second term, whose form relates with the classifier, is the gradient difference with respect to the parameter set between the positive sample and the negative.

When the sigmoid is used, Eq. (6) will be,

$$\nabla I(S_i^+, S_j^-)|_{z_{ij}} = \alpha \cdot I(S_i^+, S_j^-) (1 - I(S_i^+, S_j^-)) \quad (7).$$

And for the Gaussian-like function, it will be as,

$$\nabla I(S_i^+, S_j^-)|_{z_{ij}} = \frac{1}{\sqrt{2\pi}\sqrt{2}\sigma} \exp\left(-z_{ij}^2 / \left(2(\sqrt{2}\sigma)^2\right)\right) \quad (8)$$

The second term in Eq. (6) is determined as the classifier is chosen. In Section 3, we will illustrate it using the LDF and GMM classifier. When the two types of gradients in Eq. (6) are gotten, the classifier parameters is obtained using the iterative algorithm as,

$$\Lambda_t = \Lambda_{t-1} + \kappa_t \cdot \nabla U|_{\Lambda_{t-1}} \quad (9).$$

$\Lambda_t$  is the parameter set at the  $t$ -th iteration, and  $\kappa_t$  is a positive constant learning rate.

## 3. DESIGN SPECIFIC CLASSIFIERS

Now we detail the learning algorithms for two specific classifiers which will be used in our experiments.

### 3.1 LDF Classifier

The LDF defined in Eq. (10) is trained for text modality.

$$f(X; \Lambda) = W^T \cdot X, \quad (10)$$

with  $X$  being a  $D$ -dimensional feature vector for sample representation, and  $W$  being a  $D$ -dimensional parameter vector of the classifier, i.e.  $\Lambda$ . Thus its gradient over  $W$  for  $X$  is,

$$\nabla f(X|\Lambda)|_{\Lambda} = X \quad (11)$$

### 3.2 GMM Classifier

When the representation  $X$  is not a single vector, but a set of  $D$ -dimensional feature vectors, GMM is applied to model the distributions of the positive and the negative samples respectively. Denote  $X = (x_1, x_2, \dots, x_L)$  with  $x_i \in R^D$  and  $L$  being its size. Then the ranking function is,

$$f(X|\Lambda) = \frac{1}{L} \left( \sum_{i=1}^L \log g(x_i | w^+, \mu^+, \Sigma^+) - \sum_{i=1}^L \log g(x_i | w^-, \mu^-, \Sigma^-) \right) \quad (12),$$

with

$$g(x_i | w^c, \mu^c, \Sigma^c) = \sum_{k=1}^K w_k^c \cdot N(x_i | \mu_k^c, \Sigma_k^c), c = \{+, -\} \quad (13),$$

where  $K$  is the mixture number,  $w_k^c$  is the weight coefficient, and  $N(\cdot)$  is the Gaussian distribution with the mean  $\mu_k^c$  and covariance matrix  $\Sigma_k^c$  (Here diagonal matrix is used). Thus the parameter set being estimated is  $\Lambda = \{w_k^c, \mu_k^c, \Sigma_k^c\}, k \in [1, K], c \in \{+, -\}$ . The gradients with respect to these parameters are as,

$$\nabla f(X|\Lambda) \Big|_{w_k^c} = I^c \cdot \frac{1}{L} \sum_{i=1}^L w_k^c \quad (14)$$

$$\nabla f(X|\Lambda) \Big|_{\mu_k^c} = I^c \cdot \frac{1}{L} \sum_{i=1}^L w_k^c \cdot w_k^c \cdot \Sigma_k^{c-1} \cdot (x_i - \mu_k^c) \quad (15)$$

$$\nabla f(X|\Lambda) \Big|_{\Sigma_k^c} = -I^c \cdot \frac{1}{2L} \sum_{i=1}^L w_k^c \cdot w_k^c \cdot \left( \Sigma_k^{c-1} - \Sigma_k^{c-1} (x_i - \mu_k^c) (x_i - \mu_k^c)^T \Sigma_k^{c-1} \right) \quad (16)$$

with  $w_k^c = N(x_i | \mu_k^c, \Sigma_k^c) / g(x_i | \mu^c, \Sigma^c)$ , and,  $I^c$  is 1 if  $c$  is positive and  $-1$  otherwise. Due to the constraints that the variances and weight coefficients must be positive (summarized equal to 1 for the latter), they are updated in the log-domain in practice.

## 4. EXPERIMENTAL RESULTS AND ANALYSIS

We analyze the proposed learning algorithm on the development set of TRECVID 2005 for evaluating semantic concept detection task. We train the LDF and GMM classifiers using our learning algorithms introduced in Section 3 and Section 2. They are compared with: 1) trained using the *ML* algorithm and 2) SVM which is widely used for semantic concept detection.

### 4.1 Evaluation Metrics

We compare the different systems using the AUC metric (See Eq.2) and non-interpolated average precision (*AP*) defined as,

$$AP = \frac{1}{R} \sum_{i=1}^Q \frac{R_i}{i} \cdot I_i \quad (17)$$

$R$  is the number of true relevant image documents in the evaluation set.  $Q$  is the number of retrieved documents by the system (Here  $Q=2000$  same as used in TRECVID official evaluation).  $I_i$  is the  $i$ -th indicator in the rank list with  $Q$  images. It is 1 if the  $i$ -th image is relevant and zero otherwise.  $R_i$  is the number of relevant in the top- $i$  images.

### 4.2 Experimental Setup

The development set of TRECVID 2005 has 74,509 keyframes extracted from 137 news videos (~80 hours). 10 concepts for official evaluation are used. Two modalities, i.e., text and visual, are used. The news videos are divided into 3 sets for training, validation, and evaluation, respectively. The shots without any ASR (automatic speech recognition) /MT (machine translation) outputs are removed. Then 3,464-dimensional *tf-idf* feature is extracted for representing the shot-level text document within 3-window shots. Then LDF is trained on the text documents. The visual feature is the 12-dimensional texture (energy of log Gabor filter) extracted from a 32x32 grid. Other expressive visual features will be evaluated in the future. But here we concern the efficiency of the proposed learning algorithm rather than feature extraction. Each keyframe is uniformly segmented

into 77 grids. GMM is used for modeling texture. The dataset details are listed in Table 1.

### 4.3 Experimental Results

The results are presented on the individual modality and the fusion is not the concern here. *SVM-light* [10] with the default setting is used for text (We select the configuration based on a selected concept, *Building*, and find the default works best).

First we study the comparison on text modality. The AUC values on validation (Column *V*) and evaluation (Column *E*) sets are in Table 2. First we check the effect of the smoothing functions on the ranking performance. Comparing the Gaussian-like smoothing (Column *GAUS*) with sigmoid smoothing (Column *SG*), the former, having the average AUC values over 10 concepts as 71.7% for the validate set and 63.7% for the evaluation set, performs better than the latter whose average AUC values are 65.7% and 62.9% correspondingly. Both are significantly better than the systems trained using the ML or SVM. For all concepts except for *Prisoner* on the evaluation set, the improvements obtained from the proposed algorithms are observable. The similar conclusions as the above are obtained when comparing the AP values (See Table 3). The average AP values for *GAUS* are 6.8% on the validate set and 6.2% on the evaluation set. As the comparison they are 3.7% and 2.5% for SVM, and 2.2% and 2.2% for the ML, correspondingly.

These results demonstrate that the AUC maximization based classifier outperforms the traditional ML or error minimization based classifiers.

Table 1 Description of the TRECVID 2005 dataset

	Text	Texture
Building	T: 19,943 (2,008) V: 8,538 (1,254) E: 6,447 (958)	T: 41,978 (3,604) V: 8,295 (1,064) E: 11,173 (1,416)
Car	T: 19,943 (1,204) V: 8,538 (624) E: 6,447 (272)	T: 41,919 (2,253) V: 11,325 (767) E: 8,487 (370)
Explosion	T: 19,943 (492) V: 8,538(71) E: 6,447 (23)	T: 42,038 (641) V: 11,301 (81) E: 8,497 (26)
US_Flag	T: 19,943 (285) V: 8538(48) E: 6,447 (90)	T: 42052 (337) V: 10,970 (51) E: 8,497 (92)
Maps	T: 19,943 (423) V: 8,538(161) E: 6,447 (142)	T: 41,988 (594) V: 11,290 (171) E: 8,473 (145)
Mountain	T: 19,943 (139) V: 8,538(154) E: 6,447 (65)	T: 42,073 (385) V: 11,331 (168) E: 8,496 (73)
People	T: 19,943 (715) V: 8,538(209) E: 6,447 (86)	T: 42,021 (996) V: 11,321 (221) E: 8,473 (91)
Prisoner	T: 19,943 (43) V: 8,538(41) E: 6,447 (2)	T: 42,003 (61) V: 11,332 (43) E: 8,112 (2)
Sports	T: 19,943 (332) V: 8,538(240) E: 6,447 (98)	T: 41,753 (1,140) V: 11,310 (295) E: 8,498 (135)
Waterscape	T: 19,943 (372) V: 8,538(122) E: 6,447 (92)	T: 42,043 (819) V: 11,312 (152) E: 8,484 (110)

For LDF an image is represented by a feature vector. Another popular way of representation is to extract a set of feature vectors and the generative models such as GMM are chosen. To see the

efficiency of the presented algorithm on this representation, we extract the grid-based texture features and train the GMM models for the positive class and the negative, and use the likelihood ratio for ranking. The benchmark for comparison is the *ML* trained GMM. The values of the AUC are illustrated in Table 4 and those of AP values are in Tables 5. We see that 1) the two smoothing functions have a comparative result, and 2) the presented AUC based learning algorithms are obviously better than the benchmark.

The experiments presented above demonstrate the capability of the AUC based learning algorithm for semantic concept detection. It shows the importance of designing special classifiers for detection. The different behaviors of two smoothing functions on text and texture features exemplifies the significant role of the smoothing function on the presented learning algorithm. We will exploit other smoothing functions in the future and study how to find an optimal one for a given task.

## 5. CONCLUSION

In the paper we presented an AUC maximization based learning algorithm to design the classifier for maximizing the ranking performance. The proposed approach trains the classifier by directly maximizing an objective function approximating the empirical AUC metric. The gradient descent based method is applied to estimate the parameter set of the classifier. Two specific classifiers, i.e. LDF and GMM, and their corresponding learning algorithms are discussed. We evaluate the proposed algorithms on the development set of TRECVID'05 for evaluating semantic concept detection task. We compare the ranking performances with the classifiers trained using the ML and the error minimization method such as SVM. The systems trained using the proposed algorithm perform best on all concepts, and significant improvements on the AUC or AP values are observed. It demonstrates that for semantic concept detection, where ranking performance is much interested than the classification error, the AUC maximization based classifiers are preferred.

## 6. REFERENCES

- [1] A. Amir, et al., "IBM research TRECVID-2005 video retrieval system", *Proc. of TRECVID '05*.
- [2] A. G. Hauptmann, et al., "CMU Informedia's TRECVID 2005 Skirmishes", *Proc. of TRECVID '05*.
- [3] A. Herschtal & B. Raskutti, "Optimising area under the ROC curve using gradient descent", *ICML '04*.
- [4] C. Burges, et al., "Learning to rank using gradient descent", *ICML '05*.
- [5] C. Cortes & M. Mohri, "AUC optimization vs. error rate minimization", *Neural Information Processing Systems*, 2003.
- [6] L. Yan, et al., "Optimizing classifier performance via an approximation to the Wilcoxon-Mann-Whitney statistic", *ICML '03*.
- [7] R.O. Duda, P.E. Hart & D.G. Stock, *Pattern classification*, Wiley Interscience, 2<sup>nd</sup> edition, 2001.
- [8] S. Gao, et al., "A MFoM learning approach to robust multiclass multi-label text categorization", *ICML '04*.
- [9] S.-F. Chang, et al., "Columbia university TRECVID-2005 video search and high-level feature extraction", *Proc. of TRECVID '05*.
- [10] T. Joachims, *Learning to classify text using support vector machines*, Kluwer Academic Publishers, 2002.
- [11] T.-S. Chua, et al., "TRECVID 2005 by NUS PRIS", *Proc. of TRECVID '05*.

Table 2 AUC values (%) for Gaussian-like smoothing, sigmoid smoothing, ML, and SVM (Text)

Class	V				E			
	GAUS	SIG	ML	SVM	GAUS	SIG	ML	SVM
Building	57.5	53.1	48.5	55.2	55.0	52.1	52.1	51.2
Car	63.9	64.0	47.2	57.6	66.5	65.0	46.8	60.5
Explosion	76.9	69.4	66.8	66.3	74.1	66.0	58.8	53.7
US Flag	84.3	70.9	76.6	67.0	69.3	63.0	64.0	55.2
Maps	75.0	72.9	70.0	68.1	73.8	64.2	70.8	62.4
Mountain	62.7	61.3	42.6	56.7	67.1	60.7	39.3	45.3
People	74.1	58.4	51.4	57.4	60.8	51.5	48.6	48.9
Prisoner	69.6	68.4	73.2	55.4	27.4	70.1	87.9	32.0
Sports	89.4	79.4	71.7	81.8	79.4	73.7	62.4	69.6
Waterscape	63.1	58.9	44.6	58.0	63.1	62.2	50.2	49.0
Avg.	71.7	65.7	59.3	62.4	63.7	62.9	58.1	52.8

Table 3 AP values (%) for Gaussian-like smoothing, sigmoid smoothing, ML, and SVM (Text)

Class	V				E			
	GAUS	SIG	ML	SVM	GAUS	SIG	ML	SVM
Building	6.3	4.5	3.1	6.13	7.8	5.5	5.3	4.51
Car	10.2	7.2	1.3	6.6	8.5	5.7	0.8	6.8
Explosion	3.6	2.0	0.8	0.9	2.1	2.2	0.3	0.3
US Flag	4.4	0.9	3.1	0.4	4.7	1.9	5.5	0.8
Maps	10.4	4.8	6.4	2.2	10.1	3.9	4.0	2.2
Mountain	1.5	2.8	0.3	2.8	1.6	3.0	0.2	0.9
People	2.0	2.2	0.6	1.7	1.0	0.5	0.2	0.5
Prisoner	4.0	1.3	1.6	0.3	0.0	0.0	0.2	0.0
Sports	23.8	12.6	4.3	15.0	18.7	12.8	1.8	8.1
Waterscape	1.7	1.1	0.3	1.2	7.0	3.5	3.2	0.8
Avg.	6.8	3.9	2.2	3.7	6.2	3.9	2.2	2.5

Table 4 AUC values (%) for Gaussian-like smoothing, sigmoid smoothing, and ML (Visual)

Class	V			E		
	GAUS	SIG	ML	GAUS	SIG	ML
Building	70.1	70.2	60.4	69.7	69.6	62.3
Car	72.1	71.2	65.6	77.1	76.9	67.1
Explosion	75.5	73.9	64.3	79.4	78.4	73.7
US Flag	83.6	83.1	74.2	76.6	76.9	75.8
Maps	77.4	75.8	65.5	84.0	83.1	72.1
Mountain	86.4	86.2	71.4	91.6	91.7	74.9
People	76.3	73.7	70.0	84.3	81.9	77.1
Prisoner	64.1	64.1	58.2	82.4	82.2	42.7
Sports	82.9	83.4	72.5	78.8	78.7	69.8
Waterscape	82.4	82.4	76.0	79.6	79.2	69.4
Avg.	77.1	76.4	67.8	80.4	79.9	68.5

Table 5 AP (%) values for Gaussian-like smoothing, sigmoid smoothing, and ML (Visual)

Class	V			E		
	GAUS	SIG	ML	GAUS	SIG	ML
Building	10.8	10.3	3.9	12.4	12.2	7.7
Car	7.0	6.4	3.5	10.3	10.0	3.8
Explosion	1.8	1.5	0.8	1.8	1.6	0.8
US Flag	5.1	5.5	1.0	3.6	3.8	1.5
Maps	5.4	7.7	1.8	9.5	7.6	2.1
Mountain	8.5	8.3	2.5	8.4	8.2	2.7
People	3.2	2.6	1.8	4.4	3.8	2.6
Prisoner	0.5	0.4	0.1	0.1	0.1	0.0
Sports	15.6	16.2	3.1	7.4	7.4	1.9
Waterscape	10.2	10.3	6.0	6.1	5.9	2.5
Avg.	6.8	6.9	2.5	6.4	6.1	2.6