Multiple Kernel Learning for Vehicle Detection in Wide Area Motion Imagery

Pengpeng Liang[1] **Gregory Teodoro**[1] **Haibin Ling**[1] **Erik Blasch**[2] **Genshe Chen**[3] **Li Bai**[4]

[1]Computer & Information Science Department, Temple University, Philadelphia, PA, USA
[2]Air Force Research Lab, USA
[3]I-Fusion Technologies, Inc, Germantown, MD, USA
[4]Electrical & Computer Engineering Department, Temple University, Philadelphia, PA, USA

{*pliang,gregory.teodoro,hbling,lbai*}*@temple.edu, erik.blasch@gmail.com, genshe.chen@ieee.org*

***Abstract*—Vehicle detection in wide area motion imagery (WAMI) is an important problem in computer science, which if solved, supports urban traffic management, emergency responder routing, and accident discovery. Due to large amount of camera motion, the small number of pixels on target objects, and the low frame rate of the WAMI data, vehicle detection is much more challenging than the task in traditional video imagery. Since the object in wide area imagery covers a few pixels, feature information of shape, texture, and appearance information are limited for vehicle detection and classification performance. Histogram of Gradients (HOG) and Haar descriptors have been used in human and face detection successfully, only using the intensity of an image, and HOG and Haar descriptors have different advantages. In this paper, we propose a classification scheme which combines HOG and Haar descriptors by using Generalized Multiple Kernel Learning (GMKL) that can learn the trade-off between HOG and Haar descriptors by constructing an optimal kernel with many base kernels. Due to the large number of Haar features, we first use a cascade of boosting classifier which is a variant of Gentle AdaBoost and has the ability to do feature selection to select a small number of features from a huge feature set. Then, we combine the HOG descriptors and the selected Haar features and use GMKL to train the final classifier. In our experiments, we evaluate the performance of HOG+Haar with GMKL, HOG with GMKL, Haar with GMKL, and also the cascaded boosting classifier on Columbus Large Image Format (CLIF) dataset. Experimental results show that the fusion of the HOG+Haar with GMKL outperforms the other three classification schemes.**

I. Introduction

Vehicle detection in wide area motion imagery is an important task, and can be used to monitor traffic flow, identify illegal behavior, and follow nominated objects. WAMI provides global coverage which aids surveillance, but induces unique challenges. Due to the large amount of camera motion, small number of pixels on the targets, and the low frame rate of the video [1], the detection task is much more challenging than vehicle detection in tradition static images. In [1], Reilly et. al. provided a framework to detect and track large number of cars in a wide area image. In [2], Ling et. al. evaluated the performance of several state-of-the-art visual trackers using the Columbus Large Image Format (CLIF) dataset [3]. The approach proposed in [1] is to perform object detection by using the background subtraction technique. Background subtraction considers the static part of an image as background, and the difference between an image and its corresponding background model is considered as foreground. One famous background subtraction method is the background mixture models based on the Gaussian mixture [4, 5]. However, since the camera also moves in wide area surveillance, we have to do image registration before background subtraction. The overlap among consecutive frames decreases with the increase of the number of the frames. Nevertheless, from the results in [5], at least 235 frames are required for the Gaussian mixture model to get good performance, so the Gaussian mixture model is inappropriate for our task. In our detection approach, we use the median image background model which only requires several consecutive frames [1]. Other related work can be found in [6, 7].

Due to the flaws of the image registration using median image background model, after background subtraction, some background areas are judged as foreground areas, i.e., objects which belong to background are detected as vehicles. So, we have to distinguish vehicles and background, and the classification result will directly affect the results of confirmed detections. Fig.1 is an illustration of the detection approach using image registration and background subtraction techniques. Because our experiments use a wide area motion image (see Fig.2), the target is too small to use shape, appearance, color, or texture models such as was used in [8] for objects. Thus, we focus on WAMI vehicle classification, which is more challenging than that of aerial imagery, to confirm detected targets for improved tracking and urban surveillance.

Good descriptors and high quality classifiers are two key issues for image classification. In [9], Dalal and Triggs proposed HOG descriptor. HOG descriptor, which is based on Scale-Invariant Feature Transform (SIFT) [10] and uses oriented gradient histogram to delineate objects, has been successfully used with support vector machines (SVM) in human detection. In [11], Viola and Jones proposed a robust real-time face detection approach by using Haar feature and a cascaded boosting classifier, which is able to select a small number

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **JUL 2012**	2. REPORT TYPE	3. DATES COVERED **00-00-2012 to 00-00-2012**

4. TITLE AND SUBTITLE **Multiple Kernel Learning for Vehicle Detection in Wide Area Motion Imagery**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Temple University,Computer & Information Science Department,Philadelphia,PA,19122**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

Presented at the 15th International Conference on Information Fusion held in Sinapore on 9-12 July 2012. Sponsored in part by Office of Naval Research and Office of Naval Research Global.

14. ABSTRACT

Vehicle detection in wide area motion imagery (WAMI) is an important problem in computer science, which if solved, supports urban traffic management, emergency responder routing, and accident discovery. Due to large amount of camera motion, the small number of pixels on target objects, and the low frame rate of the WAMI data, vehicle detection is much more challenging than the task in traditional video imagery. Since the object in wide area imagery covers a few pixels, feature information of shape, texture, and appearance information are limited for vehicle detection and classification performance. Histogram of Gradients (HOG) and Haar descriptors have been used in human and face detection successfully, only using the intensity of an image, and HOG and Haar descriptors have different advantages. In this paper, we propose a classification scheme which combines HOG and Haar descriptors by using Generalized Multiple Kernel Learning (GMKL) that can learn the trade-off between HOG and Haar descriptors by constructing an optimal kernel with many base kernels. Due to the large number of Haar features, we first use a cascade of boosting classifier which is a variant of Gentle AdaBoost and has the ability to do feature selection to select a small number of features from a huge feature set. Then, we combine the HOG descriptors and the selected Haar features and use GMKL to train the final classifier. In our experiments, we evaluate the performance of HOG+Haar with GMKL, HOG with GMKL, Haar with GMKL and also the cascaded boosting classifier on Columbus Large Image Format (CLIF) dataset. Experimental results show that the fusion of the HOG+Haar with GMKL outperforms the other three classification schemes.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT **Same as Report (SAR)**	18. NUMBER OF PAGES **8**	19a. NAME OF RESPONSIBLE PERSON
a REPORT **unclassified**	b ABSTRACT **unclassified**	c THIS PAGE **unclassified**			

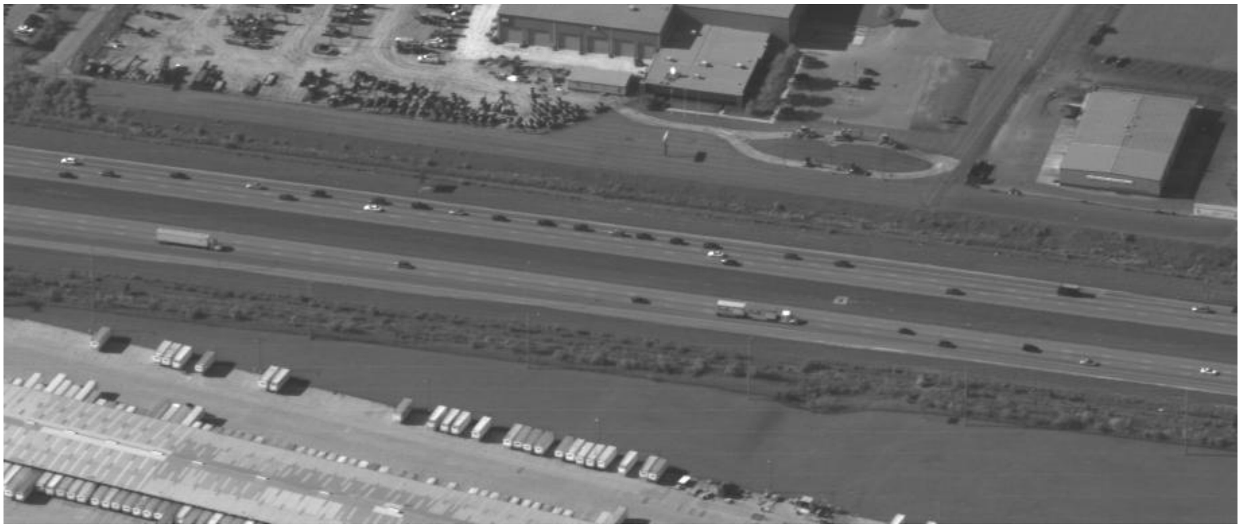

Fig. 2. Part of original aerial image

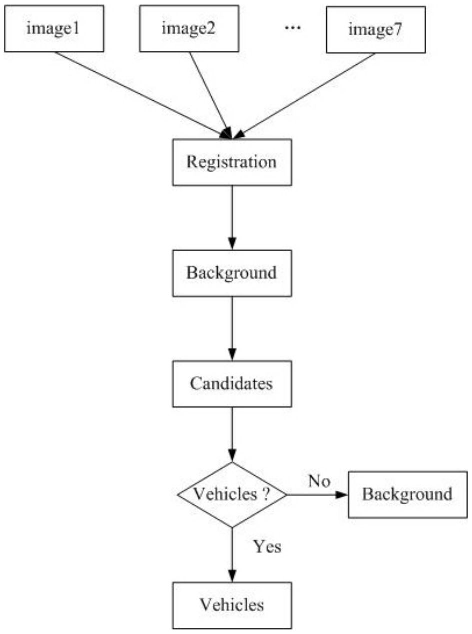

Fig. 1. Detection approach

of critical features from a huge number of potential features. Compared with HOG descriptor, the Haar feature is more appropriate to describe local characteristics of an image, but is more sensitive to color and illumination variations than HOG. Also, HOG is more suitable to represent the silhouette of objects. A key approach to confirm object detections is to use sliding window (i.e. 2D pixel patch of the image) to scan a whole image and classify each patch as either a target, or not. From the success of HOG and Haar in the detection task, we believe that Haar and HOG are powerful descriptors of image patches without using shape, color, and texture information. In our application, the data consists of grayscale images, where most of lighter colored vehicles have obvious contour, but the contour of darker colored vehicles is not as distinguishable as light ones. At the same time, the inner part of a patch can also provide useful information for classification. So, combining the HOG and HAAR descriptors complements each other for complex scene detections.

Though both HOG and Haar descriptors can benefit the classification task, it is hard for people to decide the importance of the two descriptors, i.e. calculating the trade-off between two descriptors, and the trade-off among descriptors depends on specific task. In [12, 13], Varma et. al., proposed an approach to learn the trade-off automatically based on multi-kernel learning (MKL) and Vedaldi et. al. [14] applied the multiple kernel approach to the object detection task. MKL solves this problem by learning a kernel based on the training data. The learned kernel is a combination of some base kernels, and the trade-off is represented by the coefficients of the combination. The larger the coefficient is, the more important the corresponding base kernel is. The usefulness of MKL has been demonstrated in computer vision applications, bioinformatics, ect.

In this paper, we present an effective classification scheme for vehicle classification in wide area motion image. We first use cascaded boosting algorithm [11] to do feature selection due to the large number of Haar features, we also evaluate the performance of the cascaded classifiers. After the selection of Haar feature, we combine the selected Haar features and HOG descriptors, then we use GMKL [13] to learn the trade-off and train the final classifier. We evaluate the proposed scheme on CILF dataset, and there is a significant improvement in positive vehicle detection at low false positive (alarm) rates using learning and fusion of feature descriptors. The combination of Haar and HOG descriptors outperforms both Haar and HOG when they are used separately.

The rest of the paper is organized as follows: In Section 2 and Section 3, we give a brief introduction to HOG descriptor, Haar feature and the cascaded boosting classifier. In Section 4, we describe the proposed classification scheme with GMKL using both HOG descriptor and Haar feature. In Section 5, we describe the dataset used in our experiments and give the experimental results. Finally, we conclude this paper in Section 6.

II. Histograms of Oriented Gradients (HOG)

The main idea of HOG is that a local part of an image can be described by the distribution of the oriented gradients. This distribution is estimated by histograms. The gradient of a pixel in an image is calculated by masks. Several masks were tested in [9], including a 1-D point derivative (un-centered [-1,1], centered [-1,0,1] and cubic-corrected [1,-8,0,8,-1]), 2-D derivatives $\left(\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}\right)$ and a 3×3 Sobel mask. Test results show that the simplest mask [-1,0,1] works best. For color images, gradients for each color channel are calculated and the one with the largest normal will be chosen as the pixel's gradient vector. In our application, the image is gray, so we do not need to consider the color.

For each pixel, we use [-1,0,1] to calculate a gradient horizontally and use $[-1,0,1]^T$ to calculate a gradient vertically. Based on the horizontal and vertical gradients, we can get an orientation and a weight which vote for an edge orientation histogram channel. The votes are accumulated into orientation bins over local spatial regions called cells composed by pixels. The weight of a vote is a function of the gradient magnitude and the function can be the magnitude itself, it's square, it's square root, etc. In practice the magnitude itself gives the best result.

Gradient strengths vary a lot due to the local variations in illumination and foreground-background contrast; however, effective local normalization can alleviate the variation of gradient strengths. Normalization is done by grouping cells into larger spatial blocks and normalizing each block separately. Suppose that each cell contains m bins, each block contains n cells and a detection window is composed by s blocks, the final descriptor will be vector of $m \times n \times s$ components, each of which corresponds to a bin. In practice, the blocks are overlapped so that each cell response contributes to several components of the final descriptor. In [9], four different normalization scheme are evaluated. They are (a) *L1-norm*, $\boldsymbol{v} \rightarrow \boldsymbol{v}/(\|\boldsymbol{v}\|_1 + \epsilon)$; (b) *L1-sqrt*, $\boldsymbol{v} \rightarrow \sqrt{\boldsymbol{v}/(\|\boldsymbol{v}\|_1 + \epsilon)}$; (c) *L2-norm*, $\boldsymbol{v} \rightarrow \boldsymbol{v}/\sqrt{\|\boldsymbol{v}\|_2^2 + \epsilon^2}$; (d) *L2-Hys*, L2-norm followed by limiting the maximum value of $\boldsymbol{v}$ to 0.2. It turns out that the performance of L2-Hys, L2-norm, L1-sqrt is almost the same, the and L1-norm cannot compete with the other three schemes.

III. Haar Feature and Cascade of Boosting Classifier

A. Haar Feature

The Haar feature is calculated by the difference in average intensities between different regions and was used for face detection and people detection in [15]. In [11], the Haar feature was also used and three kinds of feature were chosen. They are (a) *two-rectangle feature*, the value of this kind of feature is the difference between the sum of all pixels within two horizontally or vertically adjacent rectangular regions which have the same size and shape; (b) *three-rectangle feature*, the value is calculated by subtracting the sum within two outside rectangle from the sum in a center rectangle; and (c) *four-rectangle feature*, the value is the difference between diagonal pairs of rectangle.

In [16], Lienhart and Maydt extended the Haar featrues used in [11] and both upright and 45° rotated rectangles are considered. Also, center-surround features are added to the feature pool. Experiments in [16] shows that the additional features can enhance the expressional power of the learning system and consequently improve the performance of the detection system. In our application, we use the extended feature pool. The Haar feature can be calculated efficiently by integral image which was proposed by Viola and Jones in [11] and [16] also extended the integral image method to compute rotated Haar feature. We will not introduce the integral image method in this paper and we refer readers to [11, 16].

B. Boosted Classifier with Feature Selection

When we have a feature set, training data, and test data; any classification scheme can theoretically be used. However, the number of Haar features is huge. For example, the total number of features used in [16] within a 24×32 window is 244,162. If we use all these features, the training and test process will be very slow. Also, some features are not useful for classification and may be noise. So, feature selection is a good choice for us when we face a huge number of features.

Boosting algorithms have been used for object detection in [11, 17] and for object recognition and segmentation in [18]. The boosting classifier used for both feature selection and classification in [11] is a variant of AdaBoost. As we know, AdaBoost boosts the performance of classification by combining a set of weak classifiers to form a strong classifier. In AdaBoost, each training example is associated with a weight and we sample examples with replacement based on the weight to train a weak classifier. After the training of one weak classifier finishes, we re-weight the samples and the weight of examples classified correctly will decrease so that the next weak classifier will pay more attention to the examples misclassified by the previous weak classifier.

AdaBoost can be used for feature selection by adding a constraint to each weak classifier. The constraint is that each weak classifier has to belong to a set of classification functions each of which only depends on one feature. The weak classifier used in [11] is a simple perceptron. The weak learning algorithm is designed to select the feature that best separates the positive and negative examples and determine the optimal threshold. In [16], Lienhart and Maydy compared three different boosting algorithms: Discrete AdaBoost, Real AdaBoost and Gentle AdaBoost. Experiment results in [16] shows that Gentle AdaBoost outperforms the other two boosting algorithms with fewer features on average. So in this paper, we choose Gentle AdaBoost. In order to let the Gentle

AdaBoost have the feature selection ability, the weak classifier $h_j(x)$ is defined as follows:

$$h_j(x) = \begin{cases} \frac{W^L_{+1} - W^L_{-1}}{W^L_{+1} + W^L_{-1}} & \text{if} \quad f_j(x) < \theta \\ \frac{W^R_{+1} - W^R_{-1}}{W^R_{+1} + W^R_{-1}} & \text{otherwise} \end{cases}$$

where $f_j(x)$ is the value of the jth feature, W^L_k is the total weight of class k of the examples where $f_j(x)$ is less than θ; W^R_k is the total weight of class k of where $f_j(x)$ is greater than or equal to θ. Table I gives the description of a variant of Gentle AdaBoost according to [19] that has the ability to do feature selection.

TABLE I
A VARIANT OF GENTLE ADABOOST

```
• Given example images $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$ where
  $y_i = +1, -1$ for positive and negative examples respectively.
• For positive examples, initialize weights $w_i = \frac{1}{2m}$; for negative
  examples, initialize weights $w_i = \frac{1}{2n}$, where m and n are the
  number of positive and negative examples respectively; $H(x) = 0$
• Repeat for $m = 1, 2, \cdots, M$ :
    – For each feature $j$, train a weak classifier $h_j(x)$ by
      weighted least-squares.
    – Choose the weak classifier $h_j(x)$ that has the smallest
      weighted least-squares.
    – $H(x) \leftarrow H(x) + h_j(x)$.
    – Update $w_i \leftarrow w_i exp(-y_i h_j(x_i))$ and renormalize.
• Output the classifier $sign[H(x)] = sign[\sum_{m=1}^{M} H_m(x)]$.
```

C. Cascade of Boosting Classifier

In order to improve the detection performance and radically reduce the computation time, Viola and Jones [11] proposed an algorithm for constructing a cascade of boosted classifiers described above. The main idea is that in the detection task, the number of positive examples is much smaller than negative examples. Moreover, it is much easier to distinguish positive examples and a few negative examples than a much larger number of negative examples. Training a classifier to classify positive examples and all possible negative examples is time consuming. In the cascade scheme, we can let a classifier detect almost all of the positive examples by allowing a high false positive rate, e.g., 0.5. Though the false positive rate is high, we can still filter out a lot of negative examples, due to their number. So the subsequent classifier can now just pay attention to the hard negative examples.

The false positive rate of a trained cascade of classifiers is $F = \prod_{i=1}^{n} f_i$ and the detection rate is $D = \prod_{i=1}^{n} d_i$, where n is the number of classifiers, the f_i is the false positive rate of the ith classifier, and d_i is the detection rate of the ith classifier.

When we use the boosted classifier described in the previous section to construct the cascaded classifier, we have to pay special attention to the design. The classifier is designed to minimize errors, and thus is not specifically designed to achieve a high detection rate at the expense of a large false positive rate. We can get a trade-off between the detection rate and false positive rate by adjusting the threshold of the classifier. High threshold will produce a lower false positive rate and a lower detection rate, while a lower threshold will produce a higher false positive rate and a higher detection rate.

In practice, we first fix N, the total number of stages, f, the maximum false positive rate, and d, the minimum detection rate that each layer has to satisfy. Then each layer of the cascaded classifier is trained by the boosted classifier with the number of features used being increased until the false positive rate at that level is met. If the overall false positive rate is not yet met, another layer is added, and the positive training samples which are classified correctly by the current layer and the negative training samples which are classified incorrectly will go to the next layer. Table II gives the detail of the training algorithm for building a cascaded classifier.

TABLE II
THE TRAINING ALGORITHM FOR BUILDING A CASCADED CLASSIFIER

```
• User selects the number of layers $N$, the maximum false positive
  rate $f$ and the minimum detection rate $d$ that each layer has to
  satisfy.
• $Pos$ =set of positive training examples
• $Neg$ =set of negative training examples
• $i = 1$
• while($i \leq N$&&both $Pos$ and $Neg$ are not empty)
    – $f_i$=1
    – $n_i$=0
    – while($f_i > f$)
        ∗ $n_i \leftarrow n_i + 1$
        ∗ Use $Pos$ and $Neg$ to train a classifier with $n_i$ features
          using the variant of Gentle AdaBoost
        ∗ Choose a threshold so that $d_i$ of the current classifier
          is at least $d$
        ∗ Evaluate $f_i$ of the current classifier
    – $i \leftarrow i + 1$
    – If $i \leq n$ then evaluate the current classifier on $Pos$, and
      take the wrong classified examples out of $Pos$; evaluate the
      current classifier on $Neg$, and take the correct classified
      examples out of $Neg$.
```

IV. COMBINING HOG AND HAAR DESCRIPTORS USING GENERALIZED MKL

A. Generalized MKL

The objective of multiple kernel learning described in [12] is to create a kernel which is a combination of given base kernels and thus is the optimal descriptor. The optimal kernel is approximated as $\mathbf{K}_{opt} = \sum_k d_k \mathbf{K}_k$ where $\mathbf{d}$ corresponds to the trade-off among the base kernels. In [12], the optimization is carried out in a SVM framework subject to regularization.

$$\begin{array}{rl} \underset{\boldsymbol{w}, \boldsymbol{d}, \boldsymbol{\xi}}{\text{Min}} & \frac{1}{2}\boldsymbol{w}^t w + C\mathbf{1}^t\boldsymbol{\xi} + \boldsymbol{\sigma d} \\ \text{subject to} & y_i(\boldsymbol{w}^t \phi(\boldsymbol{x}_i) + b) \geq 1 - \xi_i \\ & \boldsymbol{\xi} \geq 0, \boldsymbol{d} \geq 0, \boldsymbol{Ad} \geq \boldsymbol{p} \\ \text{where} & \phi^t(\boldsymbol{x}_i)\phi(\boldsymbol{x}_j) = \sum_k d_k \phi^t_k(\boldsymbol{x}_i)\phi_k(\boldsymbol{x}_j) \end{array}$$

The above optimization problem can be transformed into a dual problem directly:

$$\begin{aligned} \underset{\boldsymbol{\alpha},\boldsymbol{\delta}}{\text{Max}} \quad & \mathbf{1}^t\boldsymbol{\alpha} + \boldsymbol{p}^t\boldsymbol{\delta} \\ \text{Subject to} \quad & \boldsymbol{\delta} \geq 0, 0 \leq \boldsymbol{\alpha} \leq C, \mathbf{1}^t\boldsymbol{Y}\boldsymbol{\alpha} = 0 \\ & \frac{1}{2}\boldsymbol{\alpha}^t\boldsymbol{Y}\boldsymbol{K}_k\boldsymbol{Y}\boldsymbol{\alpha} < \sigma_k - \boldsymbol{\delta}^t\boldsymbol{A}_k \end{aligned}$$

where $\boldsymbol{\alpha}$ corresponds to the support vectors, $\boldsymbol{Y}$ is a diagonal matrix with the labels of each training data $(\boldsymbol{x}_i, y_i)$ on the diagonal and $\boldsymbol{A}_k$ is the k^{th} column of $\boldsymbol{A}$. With the constraint $\boldsymbol{Ad} \geq \boldsymbol{p}$, we can encode our prior knowledge about the problem.

This dual problem is convex and has a global optimum. However, the combination of base kernels is too constraining, only the sum of the base kernels is allowed. The sum of base kernels is just the concatenation of an individual kernel's feature space. Combing base kernels in more complicated fashions can generate more expressive descriptors. So in [13], Varma and Babu extended the MKL framework in [12] to generalized MKL (GMKL) so that the combination form of base kernels can be more flexible. The corresponding optimization problem of GMKL is:

$$\begin{aligned} \min_{\boldsymbol{w},\boldsymbol{d}} \quad & \frac{1}{2}\boldsymbol{w}^t\boldsymbol{w} + \sum_i l(y_i, f(\boldsymbol{x}_i)) + r(\boldsymbol{d}) \\ \text{subject to} \quad & \boldsymbol{d} \geq 0 \end{aligned}$$

where l is the loss function, and both the regularizer r and the kernel can be any general differentiable functions of $\boldsymbol{d}$ with a continuous derivative.

The optimization problem of GMKL is non-convex and the global optimum cannot be obtained. In [13], Varma and Babu use descent gradient to calculate the local optimum with a two layer loop. In the outer loop, the optimal kernel is learnt by optimizing $\boldsymbol{d}$, while the SVM parameters of each individual kernel is learned in the inner loop with $\boldsymbol{d}$ fixed. The optimization problem can be rewritten as follows:

$$\begin{aligned} \min_{\boldsymbol{d}} \quad & T(\boldsymbol{d}) \quad \text{subject to} \quad \boldsymbol{d} \geq 0 \\ \text{where} \quad & T(\boldsymbol{d}) = \min_{\boldsymbol{w},\boldsymbol{d}} \frac{1}{2}\boldsymbol{w}^t\boldsymbol{w} + \sum_i l(y_i, f(\boldsymbol{x}_i)) + r(\boldsymbol{d}) \end{aligned}$$

In [13], the proof of the existance of $\nabla_{\boldsymbol{d}}T$ is achieved by using the dual formulation of T, which is as follows for the classification problem:

$$\begin{aligned} W_c(\boldsymbol{d}) = \underset{\boldsymbol{\alpha}}{max} \quad & \mathbf{1}^t\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha}^t\boldsymbol{Y}\boldsymbol{K}_{\boldsymbol{d}}\boldsymbol{Y}\boldsymbol{\alpha} + r(\boldsymbol{d}) \\ \text{subject to} \quad & \mathbf{1}^{\boldsymbol{t}}\boldsymbol{Y}\boldsymbol{\alpha} = 0, 0 \leq \boldsymbol{\alpha} \leq C \end{aligned}$$

where $\boldsymbol{K}_{\boldsymbol{d}}$ is the nernel matrix for a given $\boldsymbol{d}$. $T(\boldsymbol{d}) = W(\boldsymbol{d})$ for and given value of $\boldsymbol{d}$ can be obtained by writing $T = r + P$ and $W = r + D$ with strong duality between P and D.

The derivative of T with respect to $\boldsymbol{d}$ can also be calculated through W,

$$\frac{\partial T}{\partial d_k} = \frac{\partial W}{\partial d_k} = \frac{\partial r}{\partial d_k} - \frac{1}{2}\boldsymbol{\alpha}^{*t}\frac{\partial \boldsymbol{H}}{\partial d_k}\boldsymbol{\alpha}^*$$

where $\boldsymbol{H} = \boldsymbol{YKY}$ for classification. Since $W_c(\boldsymbol{d})$ is identical to the single kernel SVM duals with kernel matrix $\boldsymbol{K}_{\boldsymbol{d}}$, $\boldsymbol{\alpha}^*$ can be obtained by any SVM optimization package, e.g. LIBSVM [20]. Table III gives the description of generalized MKL.

TABLE III
GENERALIZED MKL

```
• $n \leftarrow 0$
• Initialize $\boldsymbol{d}^0$ randomly
• repeat
    – $K \leftarrow k(\boldsymbol{d}^n)$
    – Use an SVM solver to solve the single kernel problem with kernel $\boldsymbol{K}$ and obtain $\alpha^*$
    – $d_k^{n+1} \leftarrow d_k^n - s^n(\frac{\partial r}{\partial d_k} - \frac{1}{2}\boldsymbol{\alpha}^{*t}\frac{\partial \boldsymbol{H}}{\partial d_k}\boldsymbol{\alpha}^*)$
    – Projected $\boldsymbol{d}^n + 1$ onto the feasible set if any constraints are vilated.
    – $n \leftarrow n + 1$
• until converged
```

B. Classification Scheme by Combining HOG and Haar Descriptors

From the descriptions of HOG and Haar descriptors in previous sections, we can find that HOG is good at representing the gradient information of images and is more robust to illumination changes due to normalization, and the Haar feature can represent the inner part of the image patch better. The main idea of the classification scheme is to combine HOG and Haar descriptors using GMKL, so that both HOG and Haar can contribute their benefits in the final classifier as shown in Fig. 3.

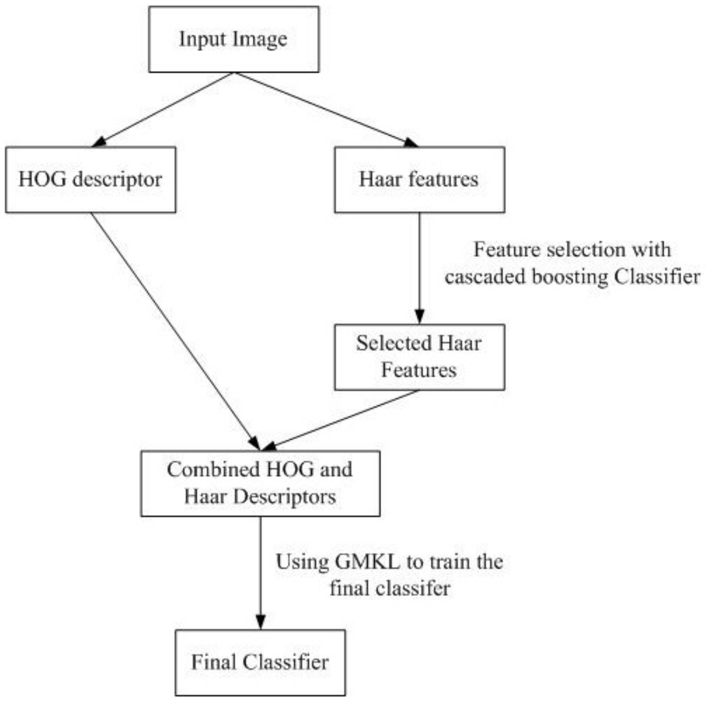
Input Image
HOG descriptor
Haar features
Feature selection with cascaded boosting Classifier
Selected Haar Features
Combined HOG and Haar Descriptors
Using GMKL to train the final classifer
Final Classifier

Fig. 3. Classification scheme

In order to create an optimal kernel, we treat each feature given to GMKL as a base kernel and radial basis function (RBF) is chosen as the kernel function. In [13], Varma and Babu demonstrated that taking product of the base kernels is superior to combing base kernels using sum which corresponds to MKL, since the product of base kernels can give a far richer representation than taking the sum. So, taking the product of base kernels is chosen to form the final optimal kernel. The optimal kernel is represented as $K_{\boldsymbol{d}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \prod_{k=1}^{M} e^{-d_k(x_{ik}-x_{jk})^2}$ where M is the total number of base kernels, i.e. the number of features, x_{ik} and x_{jk} are the value of the kth feature of examples i and j respectively.

Since each feature is considered as a base kernel, the GMKL itself has the ability to perform feature selection, the more important the feature is, the larger the coefficient of the corresponding base kernel is. For some useless features, the coefficients of the corresponding kernels are 0.

For HOG descriptors, the number of features of a 24 × 32 image is 576 when we set the block size to 12 × 12, the block stride to 4 × 4, the cell size to 6 × 6 and the number of bins to 6, so we can put the HOG descriptors into GMKL directly and GMKL will determine which features are more useful. But for Haar features, the number of features of a 24 × 32 image is 244,162. Since GMKL cannot sustain such a huge number of features, feature selection has to be done before giving Haar features to GMKL. In our approach, we use the variant of Gentle AdaBoost to do feature selection, and all the features selected by the cascaded boosting classifier are combined with the HOG descriptor. Fig. 3 illustrates our classification scheme.

V. Experiments

A. Details of the Dataset

The patches used in my experiment are cropped from images taken from an aerial platform and the patches are selected from those that have been judged as vehicles by the median image background model as used in [1]. Since the vehicles in the original images are too small, we normalize the vehicles to the size of 24 × 32. Fig.4(a) and Fig.4(b) are examples of the patches cropped from the original image. In our experiment, the training data contains 2200 positive examples and 6000 negative examples, the test data contains 633 positive examples and 1734 negative examples.

(a) (b)

Fig. 4. (a) Two positive samples. (b) Two negative samples.

B. Results

In order to select the most useful Haar features from a huge feature set, we first use the whole training set to train a cascaded boosting classifier. We set the number of stages N, the minimum detection rate of each stage, the maximum false positive rate of each stage to 14, 0.995, 0.5 respectively.

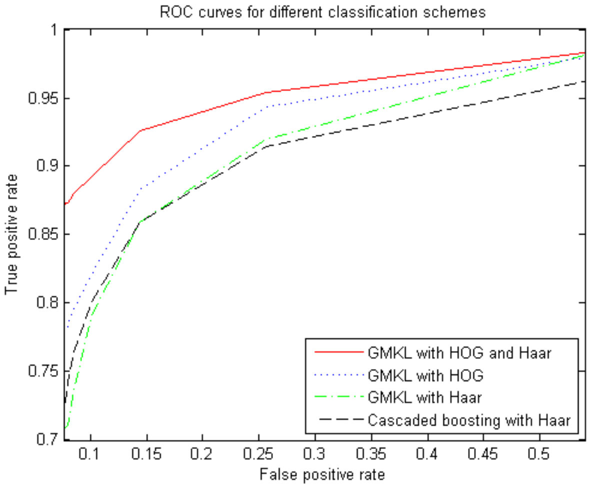

Fig. 5. ROC Curves for different classification schemes

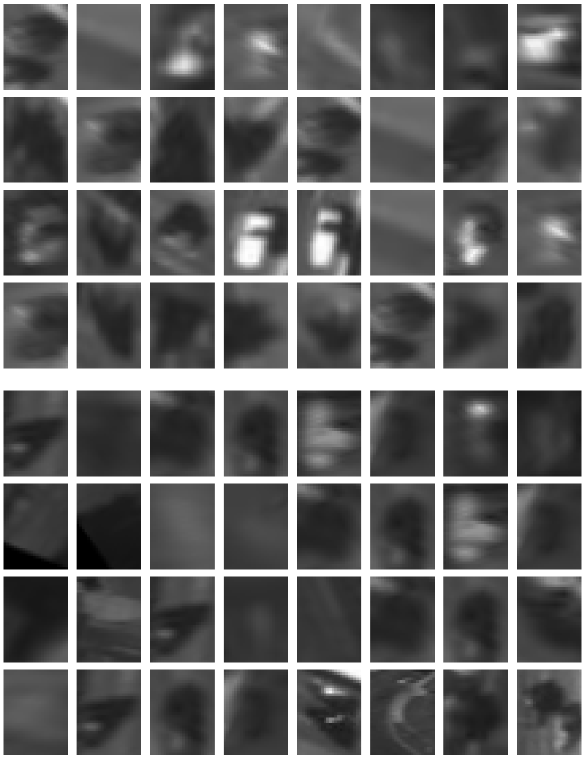

Fig. 6. Compare the wrong classified examples of each classification scheme. The first four rows are the wrong classified positive examples of GMKL(HOG+Haar), GMKL(HOG), GMKL(Haar) and Cascaded boosting respectively and the last four rows are wrong classified negative examples of each scheme respectively.

When the training process finishes, we get a 10 stage classifier which is less than 14. Note that the 10 (<14) stage classifier has already filtered out all the negative samples before the 11th stage. The total number of selected features is 388 features. From Table IV, we can see that the final classifier works well on negative samples, but poorly on positive samples. The classification strategy is that if an test example can pass

TABLE IV
THE ORIGINAL ACCURACY OF DIFFERENT CLASSIFIERS

	GMKL(Hog)	GMKL(Haar)	Cascaded boosting	GMKL(Hog+Haar)
True positive rate	0.8325	0.7930	0.7172	**0.8815**
True negative rate	0.8973	0.8929	**0.9239**	0.9141
Accuracy	0.88	0.8699	0.8687	**0.9054**

TABLE V
ACCURACY FOR DIFFERENT CLASSIFIERS(FALSE POSITIVE RATE=0.1442)

	GMKL(Hog)	GMKL(Haar)	Cascaded boosting	GMKL(Hog+Haar)
True positive rate	0.8831	0.8594	0.8594	**0.9258**
Accuracy	0.8631	0.8568	0.8568	**0.8745**

TABLE VI
BEST OVERALL ACCURACY FOR DIFFERENT CLASSIFIERS (ACCORDING TO ROC CURVES)

	GMKL(Hog)	GMKL(Haar)	Cascaded boosting	GMKL(Hog+Haar)
True positive rate	0.7804	0.7362	0.7646	**0.8736**
True negative rate	**0.9227**	0.9146	0.9146	**0.9227**
Overall accuracy	0.8847	0.8669	0.8745	**0.9096**

every stage of the cascaded classifier, i.e., every stage classifies the example as positive, the final classification result will be positive, or negative. So, a balance between positive and negative examples can be achieved by adjusting the number of stages. In our experiment, the number of total stages varies from 10 to 1, according to which, the receiver operating characteristic (ROC) curve is plotted that is shown is Fig. 5. A balance can be obtained when the true negative rate is 0.8558, i.e., the false positive rate is 0.1442, the true positive rate and accuracy at this point are 0.8594 and 0.8568 respectively.

Next, we evaluate the performance of combining HOG and Haar descriptors using GMKL. We also test the performance of HOG with GMKL and Haar with GMKL. As we know, SVM favours the class that has more examples. In order to let SVM classifier treat positive and negative samples equally, the number of positive samples and negative samples used in this part are same, 2200. For HOG descriptors, the block size is 12×12, the block stride is 4×4, the cell size is 6×6 and the number of bins is 6. For GMKL, we use the package provide by Varma which is available online[1]. We use LIBSVM to calculate $\boldsymbol{\alpha}^*$, and for the regularizer, we choose L1 regularization. Other parameters of GMKL are just the same as the value of parameters provided in the code. The original result of the trained classifier is in Table IV. Combining HOG and Haar outperforms the other three classification schemes obviously, where the accuracy of cascaded boosting classifier is almost the same to Haar with GMKL, and the HOG with GMKL is a little better.

Given a test example $\boldsymbol{x}_j$, the classification result is determined by the sign of $(\sum_i \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}_j) + b)$, where $\boldsymbol{x}_i$ is training data, y_i is the label of $\boldsymbol{x}_i$, K is the optimal kernel. So, the ROC curve of GMKL can be plotted by tuning b. In order to compare with cascaded boosting classifier, the ROC curves of Haar+Hog with GMKL, Haar with GMKL and HOG with GMKL are devleoped by varying the false positive rate in the same way as the cascaded boosting classifier which is achieved by changing the number of stages. The results are depicted in Fig. 5. From the ROC curves, we also can conclude that Haar+HOG with GMKL performs best and HOG with GMKL is better than the cascaded boosting classifier and the Haar with GMKL. Table V gives the true positive rate and accuracy of all the classification schemes when the false positive rate is fixed at 0.1442. Table VI gives the best performance of each classification scheme according to the ROC curves. Fig.6 gives the first 8 wrong classified examples of each classification scheme when the false positive rate is fixed at 0.1442. After a careful examination of the image patches, the wrong classified positive examples of GMKL with HOG and Haar are also very hard for people to distinguish. For the other three classification schemes, the 2nd image patch of GMKL with HOG, the 5th and 6th images patch of GMKL with Haar and the first image patch of cascaded boosting classifiers should be vehicles.

VI. CONCLUSION

In this paper, we have proposed a classification scheme via fusing the HOG and Haar descriptors using GMKL for classifying vehicles in wide area motion imagery. The proposed classification scheme can make use the advantages of both HOG and Haar descriptors, and can learn the trade-off between HOG and Haar automatically that is very hard for a human to determine even after careful examination of the dataset. The feature selection procedure we adopted which is a variant of Gentle AdaBoost is very effective and can achieve a satisfying degree of accuracy by selecting just 388 features from 244,162 features. Experiments conducted on CLIF dataset shows that the fusion of HOG and Haar descriptors through GMKL achieve 0.9054 accuracy without tuning any parameters which is better than HOG and Haar when they are used separately, as well as showing better results than the cascaded boosted classifier.

[1] http://research.microsoft.com/en-us/um/people/manik/code/GMKL/download.html

Acknowledgment. The work is supported in part by NSF Grant IIS-1049032.

References

[1] V. Reilly, H. Idrees, and M. Shah, "Detection and tracking of large number of targets in wide area surveillance." in *European Conference on Computer Vision*, 2010, pp. 186–199.

[2] H. Ling, Y. Wu, E. Blasch, G. Chen, H. Lang, and L. Bai, "Evaluation of visual tracking in extremely low frame rate wide area motion imagery." in *International Conference on Information Fusion*, 2011.

[3] O. Mendoza-Schrock, J. A. Patrick, and E. P. Blasch, "Video image registration evaluation for a layered sensing environment." in *IEEE Nat. Aerospace Electronics Conf. (NAECON)*, 2009.

[4] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking." in *Computer Vision and Pattern Recognition*, 1999, pp. 2246–2252.

[5] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection." in *European Workshop on Advanced Video Based Surveillance Systems*, 2001.

[6] M. Chen, S. K. Pang, T. J. Cham, and A. Goh, "Visual tracking with generative template model based on riemannian manifold of covariances," in *Int'l Conf. on Computer Vision*, 2011.

[7] H. Ling, L. Bai, E. Blasch, and X. Mei, "Robust infrared vehicle tracking across target pose change using l1 regularization," in *Proc. of the International Conference on Information Fusion (FUSION)*, 2010.

[8] M.-E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *Computer Vision and Pattern Recognition (2)*, 2006, pp. 1447–1454.

[9] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection." in *Computer Vision and Pattern Recognition*, 2005, pp. 886–893.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[11] P. A. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[12] M. Varma and D. Ray, "Learning the discriminative power-invariance trade-off," in *Int'l Conf. on Computer Vision*, 2007, pp. 1–8.

[13] M. Varma and B. R. Babu, "More generality in efficient multiple kernel learning," in *Int'l Conf. on Machine Learning*, 2009, pp. 134–141.

[14] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, "Multiple kernels for object detection," in *Int'l Conf. on Computer Vision*, 2009, pp. 606–613.

[15] C. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection." in *Int'l Conf. on Computer Vision*, 1998, pp. 555–562.

[16] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection." in *Int'l Conf. on Image Processing*, 2002, pp. 900–903.

[17] A. Torralba, K. P. Murphy, and W. T. Freeman, "Contextual models for object detection using boosted random fields," in *Neural Information Processing Systems (NIPS)*, 2004.

[18] J. Shotton, J. M. Winn, C. Rother, and A. Criminisi, "Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *International Journal of Computer Vision*, vol. 81, no. 1, pp. 2–23, 2009.

[19] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *The Annals of Statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[20] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, 2011.