ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.2, Pages : 07-13 (2013)
Special Issue of NCRTECE 2013 - Held during 8-9 February, 2013 in SMK Fomra Institute of Technology, OMR, Thaiyur, Kelambakkam, Chennai

Vehicle Detection From Aerial Imagery

C.N.Savithri*
C.L.Vijai Kumar**

* Assistant Professor, Sri Sairam Engineering College, Chennai-44
Savithri.ece@sairam.edu.in

**Final year Student-M.E-Communication systems, Sri Sairam Engineering College, Chennai-44
vijai10.7.1988@gmail.com

Abstract*—This paper presents a new technique for vehicle detection from aerial imagery. The proposed technique is simple but more effective than some of the existing techniques reported earlier. This research involves an overall cascade system that concentrates on vehicle detection mainly in rural and dense environments. This research uses and compares between certain feature extraction techniques and similar usage and comparisons are performed between four classification techniques. The overall efficiency of the overall cascade system is quite good but has certain false alarms and the efficiency is improved in the future work thereby reducing the amount of false alarms at the output.

Keywords—Feature Extraction, Histograms, Image Edge Detection, Training, Vehicle Detection

I.INTRODUCTION

Vehicle detection is very important for civilian and military applications, such as highway monitoring, and the urban traffic planning. For the traffic management, vehicles detection is the critical step. Vehicles detection must be implemented at different environment where the traffic status changing. The vehicle detection system described in this research uses nadir aerial images and compares the experimental results for several feature extraction techniques with strong discriminant power over vehicles and background, and a set of statistical classifiers including nearest neighbor, random forests and support vector machines.

The method described in this paper analyzes each location in an image to determine the target presence. The method presented here starts with a fast detection stage that looks for man-made objects and rejects most of the background. The second stage of the algorithm refines the detection results using a binary classifier for vehicle and background.

The research is organized as follows. Section I describes the fast detection stage, Section II describes the feature extraction and classification techniques, Section III makes a quantitative comparison of the techniques, and finally Section IV presents the conclusion of this work and gives directions for future research.

II.THEORY

A.FAST DETECTION

The first stage of the algorithm inspects every image location at several scales and efficiently eliminates the large majority of the background areas. The algorithm begins by quickly detecting features using the Harris corner detector.Next, areas containing a high density of features are detected. The third step clusters heavily overlapping responses. In the final step, color-based properties are used to further refine the results.

I.FEATURE DETECTION

In computer vision and image processing the concept of feature detection refers to methods that aim at computing abstractions of image information and making local decisions at every image point whether there is an image feature of a given type at that point or not. The resulting features will be subsets of the image domain, often in the form of isolated points, continuous curves or connected regions. Feature detection is a low-level image processing operation.

That is, it is usually performed as the first operation on an image, and examines every pixel to see if there is a feature present at that pixel. If this is part of a larger algorithm, then the algorithm will typically only examine the image in the region of the features. As a built-in pre-requisite to feature detection, the input image is usually smoothed by a Gaussian kernel in a scale-space representation and one or several feature images are computed, often expressed in terms of local derivative operations.

Image features based on edges detected using a Sobel operator represent a viable solution to detect a large number of man made objects and discriminate from background.

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.2, Pages : 07-13 (2013)
Special Issue of NCRTECE 2013 - Held during 8-9 February, 2013 in SMK Fomra Institute of Technology, OMR, Thaiyur, Kelambakkam, Chennai

Fig-2.1 Input Image

An improved alternative to Sobel edge detection is the use of Harris corner detection. Corners represent a better descriptor for vehicles and are able to reject background areas with large areas of random edge distribution.

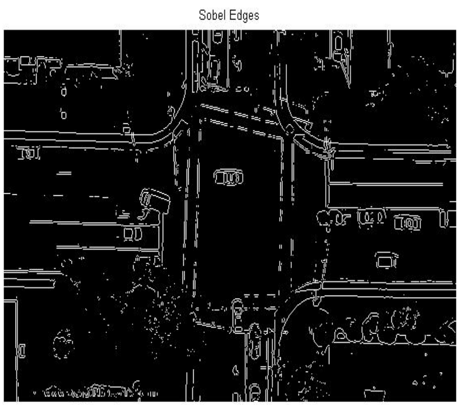

Fig 2.2 Sobel edge detection

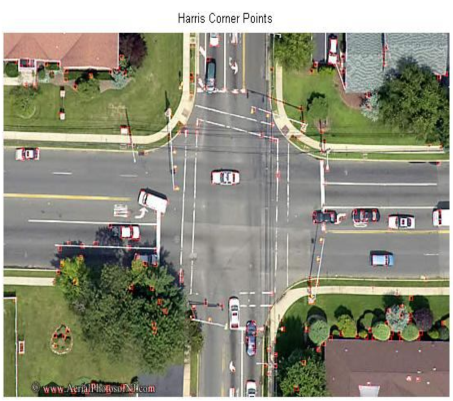

Fig 2.3 Harris corner detection

II.FEATURE DENSITY ESTIMATION

The next stage of our system involves the efficient detection of areas with high concentration of features. The algorithm searches through all rectangular windows of all aspect ratios and scales to determine those rectangles with feature density higher than a fixed threshold. The feature density score *Scorefeat*(*x*, *y*,*w*, *h*) for a particular rectangle with top left corner at position *x*, *y* in the image and of width *w* and height *h* is defined as

$$Scorefeat\ (x, y, w, h) = S\ x,y,w,h\ /\ w \times h \qquad (1)$$

rectangle. An important aspect in the computation of the *Scorefeat* is maintaining a low computational complexity. This is accomplished by discarding all redundant computations in summing over features extracted in overlapping windows.
In this approach the efficient computation of the number of features is obtained using integral images.

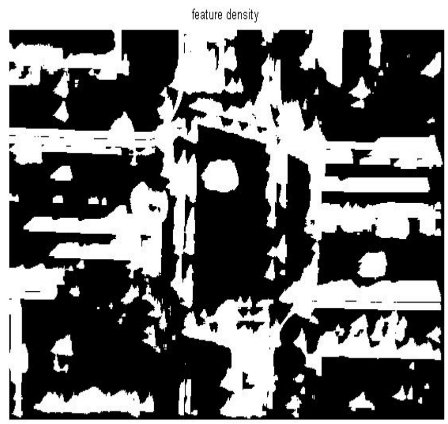

Fig 2.4 Feature Density

III.TARGET CLUSTERING

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters). It is a main task of explorative data mining, and a common technique for statistical data analysis used in many fields, including machine learning, pattern recognition, image analysis, information retrieval.

Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with low distances among the cluster members, dense areas of the data space, intervals or

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.2, Pages : 07-13 (2013)
Special Issue of NCRTECE 2013 - Held during 8-9 February, 2013 in SMK Fomra Institute of Technology, OMR, Thaiyur, Kelambakkam, Chennai

particular statistical distributions. Clustering can therefore be formulated as a multi-objective optimization problem.

The appropriate clustering algorithm and parameter settings (including values such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and intended use of the results. Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial and failure. It will often be necessary to modify preprocessing and parameters until the result achieves the desired properties.

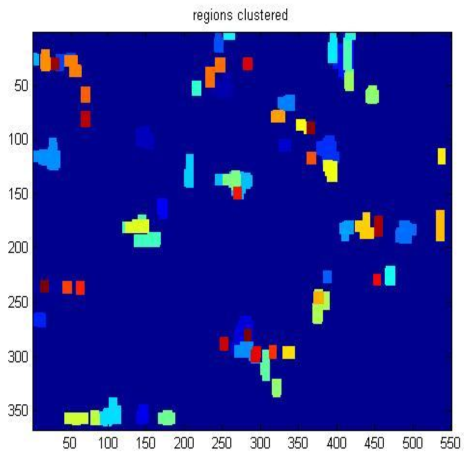

Fig 2.5 Target Clustering

All windows for which the feature density score is above a fixed threshold are assigned to potential targets. As expected the system returns a large number of responses around actual targets. At this stage, the overlapping responses are grouped together and the overlapping detection are rejected using the following iterative method.

- Step 1 Determine a set of overlapping windows.
- Step 2 Determine the centroid rectangle using the average width, height and center position of all overlapping rectangles.
- Step 3 Assign all rectangles that have an area of overlap with the centroid rectangle to the same class. The remaining rectangles are processed in step 1.
- Step 4 If the norm of the center, width and height of the centroid rectangle at consecutive iterations falls below a fixed threshold the algorithm converges, otherwise go to step 1.

IV.COLOR BASED DETECTION REFINEMENT

The target locations determined in the previous stages are refined to further reduce the false alarms using color information. A rectangular window is not a perfect fit for a vehicle and often a "correct" detection window contains background areas. On the other hand an "incorrect" detection window contains only background which often has a locally monochromatic distribution. The detection score used in this stage of the algorithm eliminates the background areas characterized by a monochromatic color distribution. The color score is given by

$$Scolor = \max F\,((\mu Fr - \mu Br)2, (\mu Fg - \mu Bg)2, (\mu Fb - \mu Bb)2) \qquad (2)$$

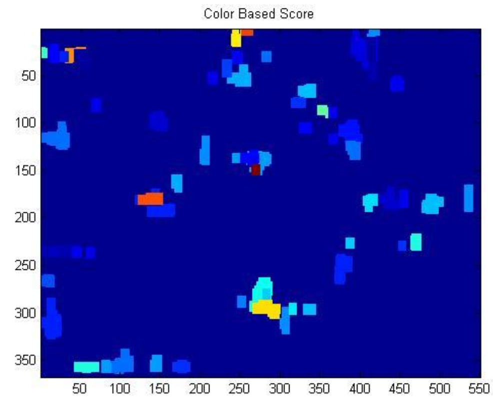

Fig 2.6 Color based detection refinement

where F is a detection window, B is a background window that includes F and μFr, μFg, μFb and μBr, μBg, μBb are the mean of the R,G,B colors inside windows F and B respectively. In our experiments the background window was chosen to have twice the number of pixels of the detection window.

B. TARGET CLASSIFICATION

classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known. The individual observations are analyzed into a set of quantifiable properties, known as various explanatory variables, features, etc. These properties may variously be categorical (e.g. "A", "B", "AB" or "O", for blood type), ordinal(e.g. "large", "medium" or "small"),integer-valued(e.g. the number of occurrences of a part word in an email) or real-valued (e.g. a measurement of blood pressure). Some algorithms work only in terms of discrete data and require that real-valued or integer-valued data be discretized into groups (e.g. less than 5, between 5 and 10, or greater than 10). An example would be assigning a given email into "spam" or "non-spam" classes or assigning a diagnosis to a given patient as described by observed characteristics of the patient (gender, blood pressure, presence or absence of certain symptoms, etc.).

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.2, Pages : 07-13 (2013)
Special Issue of NCRTECE 2013 - Held during 8-9 February, 2013 in SMK Fomra Institute of Technology, OMR, Thaiyur, Kelambakkam, Chennai

The final stage of our cascade detection system is target classification. A binary classifier assigns each of the detection results of the previous stages into vehicle or background categories and further reduces the false alarm rate. This process begins by obtaining eight additional windows surrounding the initial detection location obtained from the previous stages. These neighboring windows are selected using a window displaced 25 pixels in the vertical and/or horizontal directions. All nine of the rectangular areas are then analyzed. If any one window around a detection result is classified as a target, then the entire area is detected as a target, otherwise it is classified as background.

This stage is significantly more complex for each window but analyzes a much smaller number of windows compared to the first stage of the algorithm. This section compares two feature extraction methods (Histogram of Oriented Gradients and Histogram of Gabor coefficients) and several classification techniques (nearest neighbor, decision trees, random trees and support vector machines) for the task of vehicle detection.

I.HISTOGRAM OF ORIENTED GRADIENTS

The feature extraction method used here is based on the Histogram of Oriented Gradients (HoG) approach Histogram of Oriented Gradients (HOG) are feature descriptors used in computer vision and image processing for the purpose of object detection. The technique counts occurrences of gradient orientation in localized portions of an image. This method is similar to that of edge orientation histograms, scale-invariant feature transform descriptors, and shape contexts, but differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy.

The essential thought behind the Histogram of Oriented Gradient descriptors is that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. The implementation of these descriptors can be achieved by dividing the image into small connected regions, called cells, and for each cell compiling a histogram of gradient directions or edge orientations for the pixels within the cell. The combination of these histograms then represents the descriptor. For improved accuracy, the local histograms can be contrast-normalized by calculating a measure of the intensity across a larger region of the image, called a block, and then using this value to normalize all cells within the block. This normalization results in better invariance to changes in illumination or shadowing.

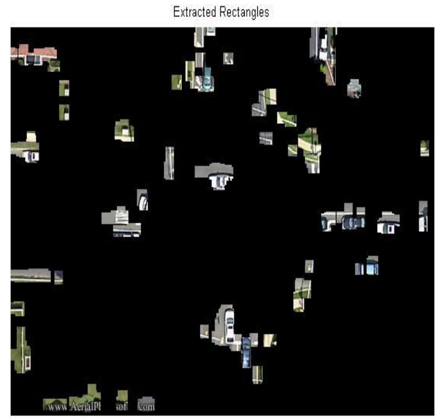

Fig 2.7 Feature Extraction

The HOG descriptor maintains a few key advantages over other descriptor methods. Since the HOG descriptor operates on localized cells, the method upholds invariance to geometric and photometric transformations, except for object orientation. Such changes would only appear in larger spatial regions. Moreover, as Dalal and Triggs discovered, coarse spatial sampling, fine orientation sampling, and strong local photometric normalization permits the individual body movement of pedestrians to be ignored so long as they maintain a roughly upright position. The HOG descriptor is thus particularly suited for human detection in images.

II.HISTOGRAM OF GABOR COEFFICIENTS

The other feature extraction algorithm was a bank of Gabor kernels. The bank contained a total of sixteen filters constructed using combinations of four orientations and four phase offsets. After filtering, a histogram of the filtered image was constructed which was used as the final feature vector for the region. For the most part, the histograms of vehicles are concentrated in the center, while the histograms for the background are more spread out.

III.CLASSIFIERS

The above features were tested using k-Nearest Neighbors (k-NN) ,Random Forests (RF) and Support Vector Machines (SVM) classification techniques.

A.K-NEAREST NEIGHBORS

The *k*-nearest neighbor algorithm (*k*-NN) is a method for classifying objects based on closest training examples in the feature space. *k*-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The *k*-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its *k* nearest neighbors (*k* is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of its nearest neighbor.

The same method can be used for regression, by simply assigning the property value for the object to be the average of the values of its *k* nearest neighbors. It can be useful to weight the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones.

The neighbors are taken from a set of objects for which the correct classification (or, in the case of regression, the value of the property) is known. This can be thought of as the training set for the algorithm, though no explicit training step is required. The *k*-nearest neighbor algorithm is sensitive to the local structure of the data.

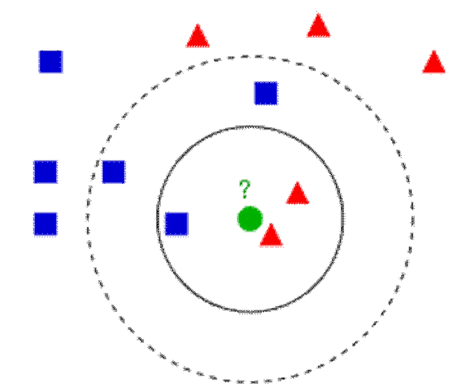

Fig 2.8 Example of *k*-NN classification. The test sample (green circle) should be classified either to the first class of blue squares or to the second class of red triangles. If $k = 3$(solid line circle) it is assigned to the second class because there are 2 triangles and only 1 square inside the inner circle. If $k = 5$(dashed line circle) it is assigned to the first class (3 squares vs. 2 triangles inside the outer circle.

B.RANDOM FOREST

Random forest (or random forests) is an ensemble classifier that consists of many decision trees and outputs the class that is the mode of the classes output by individual trees.

Each tree is constructed using the following algorithm:

1.Let the number of training cases be *N*, and the number of variables in the classifier be *M*.

2.We are told the number *m* of input variables to be used to determine the decision at a node of the tree; *m* should be much less than *M*.

3.Choose a training set for this tree by choosing *n* times with replacement from all *N* available training cases (i.e., take a bootstrap sample). Use the rest of the cases to estimate the error of the tree, by predicting their classes.

4.For each node of the tree, randomly choose *m* variables on which to base the decision at that node. Calculate the best split based on these *m* variables in the training set.

5.Each tree is fully grown and not pruned (as may be done in constructing a normal tree classifier).

For prediction a new sample is pushed down the tree. It is assigned the label of the training sample in the terminal node it ends up in. This procedure is iterated over all trees in the ensemble, and the mode vote of all trees is reported as the random forest prediction.

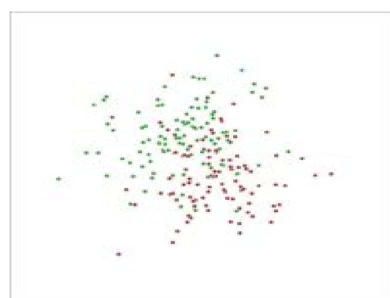

Fig 2.9a Training data consisting of two Gaussian point clouds.

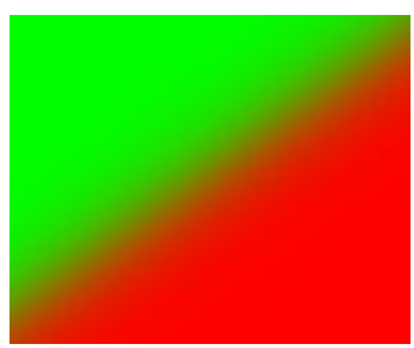

Fig 2.10bFor comparison, a logistic regression model was also trained on the same data.

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.2, Pages : 07-13 (2013)
Special Issue of NCRTECE 2013 - Held during 8-9 February, 2013 in SMK Fomra Institute of Technology, OMR, Thaiyur, Kelambakkam, Chennai

C.SUPPORT VECTOR MACHINE

Support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. The basic SVM takes a set of input data and predicts, for each given input, which of two possible classes forms the output, making it a non-probabilistic binary linear classifier. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other.

An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

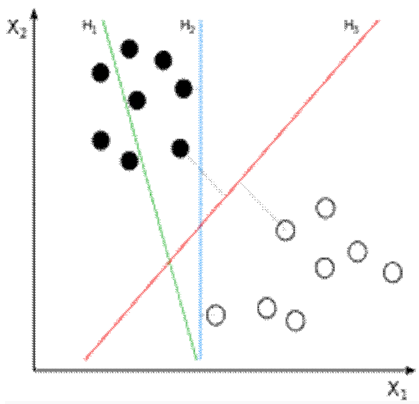

Fig 2.11a H_1 does not separate the classes. H_2does, but only with a small margin. H_3separates them with the maximum margin.

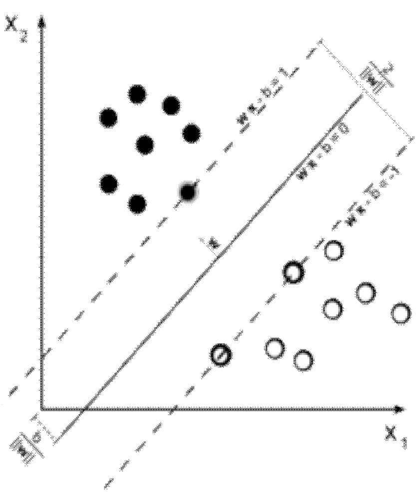

Fig 2.11bMaximum-margin hyperplane and margins for an SVM trained with samples from two classes. Samples on the margin are called the support vectors.

A support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

III.RESULTS

The results of the overall cascade system is obtained at the output end after the initial fast detection stage and final classification stage.

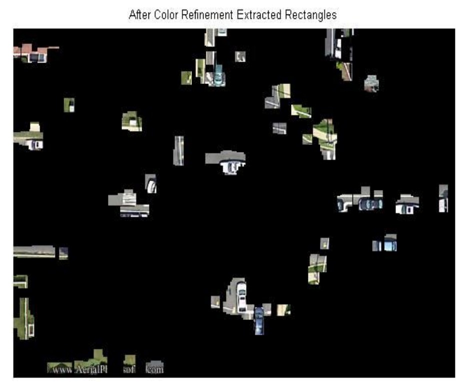

Fig 2.12 Color refined extracted rectangles

The above figure shows the final output of the overall cascade system with certain number of false alarms.

IV.CONCLUSION AND FUTUREWORK

The approach consists of a cascade detection algorithm with the first stage serving as a fast detection solution that rejects most of the background and selects patterns corresponding to man made objects. The patterns selected by this stage are further refined in the second stage using image classification techniques. The experiments for this stage compared four classification methods (KNN, SVM, decision trees and random trees) and two feature extraction techniques (histogram of gradients and Gabor coefficients). The system achieves best overall results using Gabor derived histograms and random trees classifiers.

The future research will be performed primarily towards improving the accuracy of the classification stage by replacing the binary classifiers with neural network based classifiers. And the number of false alarms are reduced by adding local binary patterns to the histogram of oriented gradients thereby increasing the output efficiency.

ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.2 , No.2, Pages : 07-13 (2013)
Special Issue of NCRTECE 2013 - Held during 8-9 February, 2013 in SMK Fomra Institute of Technology, OMR, Thaiyur, Kelambakkam, Chennai

REFERENCES

[1]Joshua Gleason, Ara V. Nefian, Xavier Bouyssounousse, Terry Fong and George Bebis 2011

[2] G´erard Biau, Luc Devroye, and G´abor Lugosi. Consistency of random forests and other averaging classifiers. *J. Mach. Learn. Res.*, 9:2015–2033, 2008.

[3] Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[4] T. Cover and P. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21 – 27, jan. 1967.

[5] Navneet Dalal and Bill Triggs. Histogram of Oriented Gradients for Human Detection. *IEEE Computer Society on Computer Vision and Pattern Recognition(CVPR'05)*, 2005.

[6] M.A. Hearst, S.T. Dumais, E. Osman, J. Platt, and B. Scholkopf. Support vector machines. *Intelligent Systems and their Applications, IEEE*, 13(4):18 –28, jul. 1998.

[7] Stefan Hinz. Detection of Vehicles and Vehicle Queues in High Resolution Aerial Images. In *Photogrammetrie - Fernerkunding -Geoinformation (PFG) 3*, pages 203–215, 2004.

[8] ZuWhan Kim and Jitendra Malik. Fast Vehicle Detection with Probabilistic Feature Grouping and its Application to Vehicle Tracking. *Computer Vision, IEEE International Conference on*, 1:524, 2003.

[9] Krystian Mikolajczyk and Cordelia Schmid. Scale and Affine Invariant Interest Point Detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[10] Uday Rajanna, Ali Erol, and George Bebis. A Comparative Study on Feature Extraction for Fingerprint Classification and Performance Improvements Using Rank-Level Fusion. *Pattern Analysis and Applications*, 13:263–272, 2010.

[11] Paul Viola and Michael J. Jones. Robust Real-Time Face Detection. *International Journal of Computer Vision*, 57(2):137–154, 2004.

[12] Sun Zehang, George Bebis, and Ronald Miller. Monocular Precrash Vehicle Detection: Features and Classifiers. *IEEE Transactions on Image Processing*, 15(7):2019–2034, July 2006.

[13] Tao Zhao and Ram Nevatia. Car Detection in Low Resolution Aerial Images. In *Image and Vision Computing*, pages 710–717, 2001.