Semi-supervised Vehicle Recognition: An Approximate Region Constrained Approach

Rui Zhao[1,2], Zhihua Wei[1,3], Duoqian Miao[1], Yan Wu[1], and Lin Mei[2]

[1] Department of Computer Science and Technology, Tongji University,
201804, Shanghai, China
[2] The Third Research Institute of the Ministry of Public Security,
201204, Shanghai, China
[3] State Key Laboratory for Novel Software Technology, Nanjing University,
210093, Nanjing, China

Abstract. Semi-supervised learning attracts much concern because it can improve classification performance by using unlabeled examples. A novel semi-supervised classification algorithm SsL-ARC is proposed for real-time vehicle recognition. It makes use of the prior information of object vehicle moving trajectory as constraints to bootstrap the classifier in each iteration. Approximate region interval of trajectory are defined as constraints. Experiments on real world traffic surveillance videos are performed and the results verify that the proposed algorithm has the comparable performance to the state-of-the-art algorithms.

Keywords: Semi-supervised learning, object recognition, approximate region interval, constraints.

1 Introduction

Robust object recognition under real-world conditions is still a challenging task and limits the use of state-of-the-art methods in industry (e.g., video surveillance [1]). Recently, object recognition based on semi-supervised learning attracts much attention which exploits both labeled and unlabeled objects in learning classifier [2, 3]. It has been shown that for some kinds of problems, the unlabeled data can dramatically improve the performance of classifier. However, general semi-supervised learning algorithms [4] assume that the objects are independent so that they do not enable to exploit relationship between objects which might contain a large amount of information. For example, in a surveillance video, the certain vehicle location defines a trajectory which represents a kind of relation among the labeling of the video sequence. The objects close to the trajectory are positive examples; objects far away from the trajectory are negative ones.

In semi-supervised learning , more feasible strategy for labeling unlabeled examples is guided by some supervisory information [2]. This information may be in a form of labels associated with some examples or some forms of constraints [5]. The second is more general. Its basic idea is combining detector and

T. Li et al. (Eds.): RSKT 2012, LNAI 7414, pp. 161–166, 2012.

tracker [6] where the detector serves as initial model for semi-supervised learning. Object tracking could learn some information from underlying examples, i.e. estimating object location in frame-by-frame fashion. The object to be tracked can be viewed as a single labeled example and the video as unlabeled data. Many authors perform self-learning and co-training for tracking object [7–9]. This kind of approach predicts the position of the objects with a tracker and updates the model with positive examples that are close and negative examples that are far from the current position. The strategy is able to adapt the tracker to new appearances and background, but breaks down as soon as the tracker makes a mistake. In order to avoid above problem, Kalal proposed a bootstrapping method by using structure constraints named P-N learning [10]. This approach integrated tracker and detector and made them correcting each other. The approach demonstrated robust tracking performance in challenging conditions and partially motivated our research.

Basing on the above idea, the paper proposes a new paradigm for learning from dependent unlabeled objects. Relations between objects are used in parallel with classification algorithm to mutually rectify their errors. The relation among the objects is exploited by so called Approximate Region Constraints (ARC). That is, lower approximate region specifies the most frequently acceptable patterns of positive labels, i.e. objects on the moving trajectory. Upper approximate region specifies possibly acceptable pattern of positive labels, i.e. objects near to the trajectory.

The rest of the paper is organized as follows. Section 2 defines the vehicle recognition problem and formulates the semi-supervised algorithm based on ARC. Section 3 validates and analyzes the algorithm on real world traffic videos. The last section concludes and discusses the future works.

2 Semi-supervised Learning Based on Approximate Region Constraints (SsL-ARC)

2.1 SsL-ARC Algorithm Framework

Object detection problem could be regarded as an on-line learning process as follows. Let x be an image patch from a frame set X in videos and y be a label from lable-space $Y = \{1, -1\}$. A set of examples X_l and corresponding set of labels Y_l will be denoted as (X_l, Y_l) and called a labeled set. The task of vehicle recognition is to learn a classifier $f : X \to Y$ from a prior labeled set (X_l, Y_l) and bootstrap its performance by unlabeled data X_u.

A constrained boosting process is defined to verify the labels by classifier in accordance with the trajectory of object. Its framework is shown in Fig.1. Based on this idea, a SsL-ARC algorithm is proposed.

Classifier. Object detection is performed by classification method that decides about presence of an object in an input frame and determines its location. Based on the "bag of words" ideology, an object could be described by the collection of

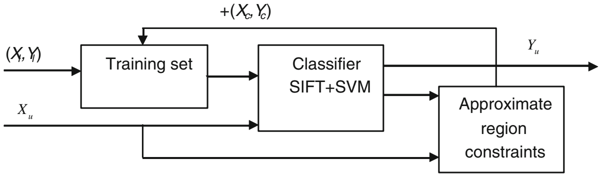

Fig. 1. SsL-ARC algorithm framework

interest points which are perceptually salient [11]. As a result, the Harris-Affine corner measure is used to find the salient feature and SIFT [12] is used for salient point description. In training phase, the Harris-Affine detector detects several salient points and the SIFT descriptor yields a 128 dimensional vector for each of these keypoints. This descriptor along with the scale information of the respective keypoint forms a 131 dimensional pattern. The labeled object represented by the feature vector was assigned a positive label and used for training. Core Vector Machine (CVM) is adopted as classifier which is a recently proposed flavor of SVM [13]. In testing phase, for a patch in a frame, all the salient keypoints were classified. If the number of keypoints belonging to positive class exceeding 50% of the total number of keypoints, the patch is labeled as positive.

Constraints. Selecting a single patch in the first frame as the labeled object, we could draw a trajectory curve in the video volume based on the fact that a single object appears in one location only in a given frame. The curve is obtained by a CamShift tracker which follows the selected object from frame to frame [14]. CamShift essentially climbs the gradient of a back projected probability distribution computed from rescaled color histograms and looks for the nearest peak in an axis-aligned search window. These occur when objects in video sequences are being tracked and the object moves so that the size and location of the probability distribution changes in time. CamShift which is originated from mean shift uses continuously adaptive probability distributions so that it adjusts the size and angle of the target rectangle each time it shifts. It does this by selecting the scale and orientation that are the best fit to the target-probability pixels inside the new rectangle location.

2.2 Approximate Region Constraints

Assume that the location of trajectory in time k is (m_0^k, n_0^k), the centroid of the patch detected by the classifier in time k is (m_1^k, n_1^k), the distance between the centroid of detected patch and the trajectory is as follows.

$$D(k) = \sqrt{(m_1^k - m_0^k)^2 + (n_1^k - n_0^k)^2} \tag{1}$$

Approximate region interval (Ari) based on interval set [15] is defined according to the trajectory obtained by the tracker. The definition of an interval set $Ari_i(a, b)$ of object x_i and its pair of threshold [16] is as follows.

Definition 1. For an vehicle object $x_i \in X_t$, its approximate region interval $Ari_i(a,b) = [\alpha_i^{(a,b)}, \beta_i^{(a,b)}]$, $[a,b]$ is the time span of a video sequence, $k \in [a,b]$.

$D(k) \le \alpha_i^k$ is defined as the lower approximate region of object x_i which denotes the frequent occurrence region of the vehicle object x_i;

$D(k) \le \beta_i^k$ is defined as the upper approximate region of object x_i which denotes the possible occurrence region of the vehicle x_i;

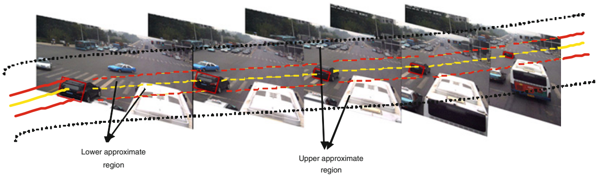

Fig. 2. Rough set approximate region of vehicle trajectory

The lower approximate region and upper approximate region of the trajectory could be described as Fig.3. Approximate region constraint function accepts a group of examples with labels given by the detector (X_u^k, Y_u^k) and output s subset of the group with changed labels (X_c^k, Y_c^k). The lower approximate region constraint which is denoted as L-constraint, is used to identify examples that have been labeled negative by the classifier but it belong to the lower approximate region of trajectory. In iteration k, L-constraint add $n^+(k)$ examples to the training set with labels changed to positive. On the other hand, the upper approximate region constraint which is named U-constraint, is used to identify examples that have been labeled as positive but beyond the upper approximate region of trajectory. In iteration k, U-constraint add $n^-(k)$ examples to the training set with labels changed to negative. These constraints enlarge the pool of positive and negative training set and thus improve the discriminative ability of classifier.

3 Experiments

The proposed algorithm is tested on 6 video sequences originates from surveillance videos of Shanghai Bureau of Urban traffic Management.

The algorithm SsL-ARC is initialized in the first frame by learning the Initial Detector noted as Ini-Detec and sets the initial position of the tracker. For each frame, the detector and the tracker find the location(s) of the object. The tracker finds the trajectory of given object in the continuous frames while the detector recognizes all patches similar to the given object by classifier. The objects close to the trajectory and far away from the trajectory are used as positive examples and negative examples, respectively.

The detector is implemented by integrating the source of Harris-Affine corner measure, SIFT descriptor [17] and Core Vector Machine [13] with the default parameters. Gaussian kernel is selected in CVM. The performance of the SsL-ARC algorithm is evaluated by precision P, recall R and F-measure since the algorithm is a boosted classification algorithm in nature. P is the number of correct detections divided by number of all detections, R is the number of correct detections divided by the number of object occurrences that should have been detected. F combines these two measures.

The performance of Ini-detec, Boost-detec and constrained tracker noted as Cons-Track on 6 sequences is listed in Table1. The performance is measured by P, R and F-measure averaged over time.

From the Table 1, we could observed that the Ini-detec has high precision rate while very low recall rate. the Boost-detec has a significant increase on recall rate. The SsL-ARC algorithm is compared with the famous P-N Learning algorithm [10] on the same video sequences and has better performance on three sequences that occupy half of the dataset. In our experiments, the proposed algorithm shows better performance than P-N Learning.

Table 1. Performance analysis of SsL-ARC

Sequence	Frame	SsL-ARC algorithm(P/R/F)			P-N Learning(P/R/F)
		Ini-detec	Boost-detec	Cons-tracker	
Bus	552	1.00/0.02/0.03	**0.91/0.45/0.60**	0.83/0.41/0.55	0.90/0.38/0.53
Crossing1	705	1.00/0.01/0.02	0.87/0.32/0.47	0.85/0.56/0.67	**0.87/0.45/0.59**
Crossing2	720	1.00/0.06/0.12	**0.90/0.75/0.82**	0.63/0.78/0.70	0.88/0.67/0.76
Evening	487	1.00/0.04/0.08	0.91/0.43/0.58	0.80/0.37/0.51	**0.91/0.48/0.63**
Night1	455	0.76/0.01/0.02	**0.55/0.20/0.29**	0.23/0.22/0.22	0.44/0.18/0.26
Night2	462	1.00/0.06/0.12	0.92/0.65/0.76	0.89/0.8/0.84	**0.90/0.68/0.77**

4 Conclusions

A boosting semi-supervised learning algorithm SsL-ARC constrained by approximate region interval is proposed. The constraints are obtained by defining lower approximate region and upper approximate region of given object trajectory. The initial detector which is based on SIFT features and SVM classifier is bootstrapped by a feedback from these constraints. The algorithm is applied to the problem of real-time vehicle object recognition. Experiments on real world surveillance videos show that the proposed algorithm has better performance compared to the state-of-the-art methods. Further work may focus on defining more refined constraints.

Acknowledgments. This research is supported in part by the National Natural Science Foundation of China under Grants No.60970061, Key Program supported by the Shanghai Committee of Science and Technology under Grant No.10511500700, the opening project of State Key Laboratory for Novel Software Technology in Nanjing University (No.KFKT2011027) and the opening project of Shanghai Key Laboratory of Digital Media Processing and Transmission (No.2011KF03).

References

1. Dee, H., Velastin, S.: How close are we to solving the problem of automated visual surveillance? Mach. Vison. Appl. 19(5-6), 329–343 (2008)
2. Chapelle, O., Schôlkopf, B., Zien, A.: Semi-Supervised Learning. MIT Press, Cambridge (2006)
3. Zhu, X., Goldberg, A.: Introduction to semi-supervised learning. Morgan Claypool Publishers, USA (2009)
4. Nigam, K., McCallum, A., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using EM. Mach. learn. 39(2), 103–134 (2000)
5. Abu-Mostafa, Y.: Machines that learn from hints. Scientific American 272(4), 64–71 (1995)
6. Li, Y., Ai, H., Yamashita, T., Lao, S., Kawade, M.: Tracking in low frame rate video: A cascade particle filter with discriminative observers of different lifespans. In: 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE Press, Minneapolis (2007)
7. Grabner, H., Bischof, H.: On-line boosting and vision. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 260–267. IEEE Press, New York (2006)
8. Avidan, S.: Ensemble tracking. IEEE T. Pattern Anal. 29(2), 261–271 (2007)
9. Yu, Q., Dinh, T.B., Medioni, G.: Online Tracking and Reacquisition Using Co-trained Generative and Discriminative Trackers. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 678–691. Springer, Heidelberg (2008)
10. Kalal, Z., Matas, J., Mikolajczyk, K.: P-N Learning: Bootstrapping Binary Classifiers by Structural Constraints. In: 23rd IEEE Conference on Computer Vision and Pttern Recognition, pp. 13–18. IEEE Press, San Francisco (2010)
11. Sivic, J., Russell, B., Efros, A., Zisserman, A., Freeman, W.: Discovering Objects and their Localization in Images. In: 10th IEEE International Conference on Computer Vision, pp. 370–377 (2005)
12. Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision 60(2), 91–110 (2004)
13. Tsang, I.W., Kwok, J.T., Cheung, P.M.: Core vector machines: Fast SVM training on very large data sets. J. Mach. Learn. Res. 6, 363–392 (2005)
14. Bradski, G.R.: Computer Vision Face Tracking For Use in a Perceptual. User Interface. Intel Technology Journal, 2nd Quarter (1998)
15. Yao, Y.Y.: Interval-set algebra for qualitative knowledge representation. In: 5th International Conference on Computing and Information, pp. 370–374 (1993)
16. Yao, Y.Y.: Three-way decisions with probabilistic rough sets. Information Sciences 180(3), 341–353 (2010)
17. Affine covariant region detectors, `http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html`