Practical object recognition in autonomous driving and beyond

Alex Teichman and Sebastian Thrun
Stanford University
Computer Science Department

***Abstract*—This paper is meant as an overview of the recent object recognition work done on Stanford's autonomous vehicle and the primary challenges along this particular path.**

The eventual goal is to provide practical object recognition systems that will enable new robotic applications such as autonomous taxis that recognize hailing pedestrians, personal robots that can learn about specific objects in your home, and automated farming equipment that is trained on-site to recognize the plants and materials that it must interact with.

Recent work has made some progress towards object recognition that could fulfill these goals, but advances in model-free segmentation and tracking algorithms are required for applicability beyond scenarios like driving in which model-free segmentation is often available. Additionally, online learning may be required to make use of the large amounts of labeled data made available by tracking-based semi-supervised learning.

I. Introduction

Object recognition is a key missing component of many socially relevant robotic systems. Recently, some progress has been made in this direction using Stanford's DARPA Urban Challenge vehicle, Junior [6], to recognize cars, pedestrians, and bicyclists in natural street scenes. This paper is an overview of that work and the primary challenges that lie ahead.

Junior, shown in Figure 1, is equipped with a Velodyne HDL-64E S2 rotating 64-beam laser range finder; its measurements are integrated over time while the vehicle is in motion with a tightly coupled GPS/IMU, the Applanix POS LV 420. The object recognition algorithms discussed here use just this depth data; a Ladybug 3 panoramic camera is mounted above the Velodyne, but so far is used only for visualization of the results.

Most of the technical results discussed here can be found in [10] or [9]. There are, of course, other valid approaches with different tradeoffs; these will be briefly discussed in Section V.

II. Long term vision

A. Object recognition for autonomous driving

According to the U.S. Department of Transportation, over 30,000 people were killed due to car accidents in the U.S. in 2009 [7]. Autonomous vehicles have the potential to significantly reduce this number. Additionally, wasted time spent commuting could be reduced, increasing overall productivity. Fuel efficiency could be increased by caravaning on highways, thus reducing CO_2 emissions.

The recent self-driving car project at Google [11] has made significant progress towards the long-term vision of autonomous vehicles. Their work has shown that, at least so far, detailed and highly-accurate object recognition is largely not required; it remains to be seen whether this is true for the extremely high-reliability systems that are necessary for real-world use.

Fig. 1: Junior, the autonomous driving platform.

Despite this, there are many areas where sophisticated object recognition methods would be beneficial. Both Google's and Junior's autonomous driving systems are based on building detailed maps of the world, then localizing to them during operation. As a result, construction zones present a significant challenge. Robust detection of traffic cones, construction equipment, or workers holding stop signs could significantly improve handling of these situations. Additionally, polite and safe behavior in complex intersections, one-lane roads with bi-directional traffic, etc., may also require advances in object recognition. Safety could be enhanced by learning behavioral models of different object types, allowing an autonomous vehicle to anticipate the actions of bicyclists versus cars at stop signs, for example. While not strictly necessary, a person on the street corner hailing an autonomous taxi should be recognized as such.

B. Object recognition for robotics in general

More broadly, special purpose and general purpose robots have the potential to revolutionize society similar to the way computers have. In farming, for example, a generic picking

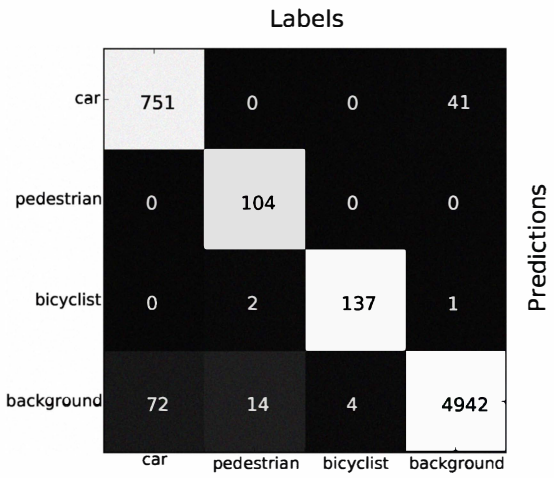

Fig. 2: Confusion matrix results from [10]. This is for the track classification task, which assumes correct segmentation and tracking.

device could be mass produced, then taught on-site by a farmer with no expertise in robotics to recognize the type of fruit it should be harvesting. Similarly, an autonomous tractor outfitted with a high power laser could be mass produced, then taught to recognize particular weeds versus crop plants at farms of widely different types, improving agricultural efficiency. In construction or manufacturing, generic delivery robots could be used to pick up raw materials and move them in to place; first, they must be taught to recognize these particular raw materials.

III. Desiderata

There is, of course, no one "right" list of desired properties for practical object recognition systems. This list represents the near- to medium-term horizon of what we feel would enable a number of socially-useful applications that are currently not possible.

- **Accurate**
- **Real-time capable**
- **Inherently multi-class** – For practical application, we would like to be able to recognize at least on the order of tens of classes.
- **Inherently multi-descriptor** – There are many useful cues in object recognition that come from diverse sources, including depth sensors, cameras, and radar; it is desirable to have a learning framework that can incorporate them all with little pain (*e.g.*, no hand-tuned scaling parameters).
- **Can learn without massive hand-labeled training sets**
- **Can add new object classes and descriptors without relearning from scratch**
- **Little manual feature engineering required**

IV. Overview of existing progress

We now consider recent object recognition work on Junior in the context of the desired properties of Section III.

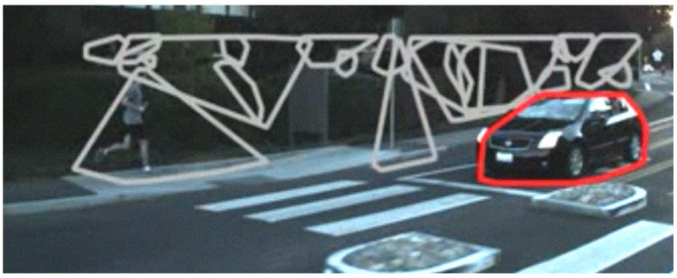

Fig. 3: Segmentation and tracking failures are the most common source of object recognition errors. In this example, a pedestrian gets segmented together with a large stretch of curb, resulting in a false negative, while a well-segmented car is correctly identified. Best viewed in color.

A. Supervised method

To summarize the current status, the system is real-time, accurate when given good segmentation and tracking, and inherently multi-class and multi-descriptor. Learning without massive hand-labeled datasets will be addressed in Section IV-B, and the remaining desired properties require work or have not yet been shown experimentally.

The laser-based object recognition algorithm used on Junior is broken down into three main components: segmentation, tracking, and track classification. Objects are segmented from the environment using depth information, then tracked with a simple Kalman filter. Thus, the segmentation and tracking methods are *model-free*, *i.e.* no object class model is used during these stages. Classification of tracked objects is achieved with a boosting algorithm applied across several high dimensional descriptor spaces which encode size, shape, and motion properties. This system is described in [10].

Figure 2 shows results for the track classification subproblem from [10]. The largest source of errors in the full object recognition problem is segmentation and tracking; see Figure 3. Cars and bicyclists actively avoid becoming segmented together with the environment, but this is unfortunately not generally true for pedestrians and other object classes that could be of interest. Quantitatively, the method achieved 98.5% track classification accuracy - that is, segmentation and tracking failures are *not* included - on a large test set from real street scenes; qualitatively, when considering the full object recognition problem, undersegmentation frequently results in false negatives of objects that get too close to other objects.

Maintaining real-time capability has (for now) dictated that this system remain feed-forward; that is, more mathematically-sophisticated methods that jointly consider segmentation, tracking, and classification do not yet seem real-time capable. Classifying only pre-segmented objects means that the system can spend more time on each candidate than, for example, the sliding window systems prevalent in computer vision.

B. Semi-supervised method

Model-free segmentation and tracking enables a highly effective method of learning object models without the need for

massive quantities of hand-labeled data. This method, known as tracking-based semi-supervised learning [9], iteratively a) learns a classifier, and b) collects new, *useful* training instances by using tracking information. For example, the method can learn to recognize half-occluded bicyclists from unlabeled tracks that include both unoccluded and half-occluded views. A video example of object recognition results using this method can be seen at [1].

This method has been shown to achieve the relatively high track classification accuracy of [10], but with only three hand-labeled training tracks of each object class. This has the potential to enable non-robotics-experts to teach robots about new objects; it is probably impractical to assume non-experts would have the time or ability to assemble large training sets by hand. This could be an essential ingredient to adapting robots to specific tasks at specific work sites.

V. Related work

Whereas our method relies on model-free segmentation and tracking, the tracking-by-detection approach involves running a detector on individual frames, then tracking the resulting detections. This approach is exemplified in the pedestrian detection work of [8]. As it is formulated in [9], tracking-based semi-supervised learning requires model-free segmentation and tracking; however, it might be possible to use the results of FlowBoost [2] to achieve similar gains in a tracking-by-detection framework, though this method requires sparse labels rather than completely unlabeled data.

The boosting algorithm we use was specifically designed to work well with diverse, high dimensional descriptor spaces. One alternative approach from the SVM literature is multiple kernel learning (MKL), first given an efficient implementation in [3], in which an SVM uses multiple kernels to intelligently combine different descriptor spaces into its classifications. An advantage of boosting here is that the math of combining predictions over the course of a track using a discrete Bayes filter [10] depends on the classifier outputting a log odds estimate, which boosting produces. [1]

VI. Challenges and opportunities

A. *Segmentation and tracking*

The primary challenge evident from the results in [10] and [9] is in developing fast, effective algorithms for model-free segmentation and tracking. The object recognition algorithms discussed in this paper are only applicable to cases where model-free segmentation and tracking is at least somewhat reliable. This is not the case in, for example, the cluttered indoor environments that a household robot such as Willow Garage's PR2 would commonly operate in. Additionally, while there are many cases in which simple depth segmentation works in street scenes, there are many that do not. A hailing pedestrian could probably be robustly recognized, but only if he was not touching a bush or lamppost.

[1] While the margin output from SVMs would likely be effective in practice, it is not a log odds estimate in the way the output from boosting or logistic regression is; see [5].

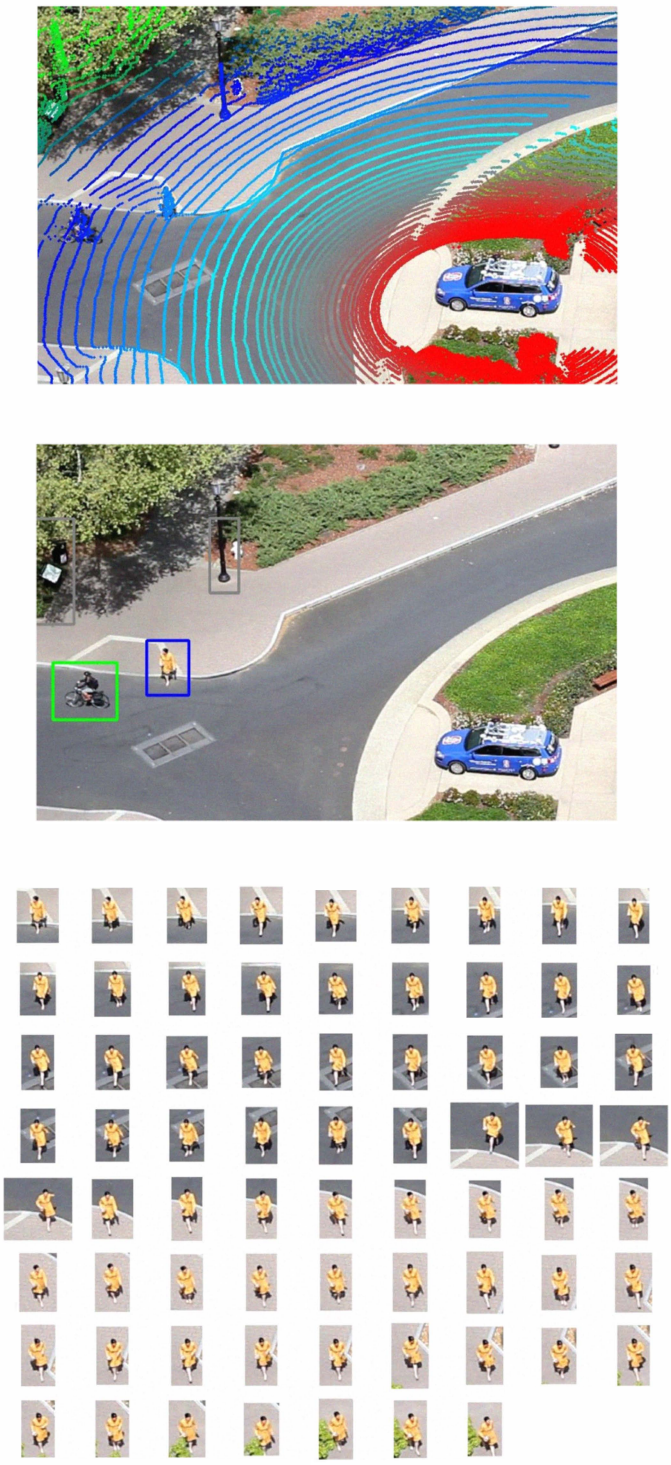

Fig. 4: The laser-based classifier can produce large quantities of automatically labeled training data for a more economical vision-based classifier to learn from. Best viewed in color.

B. *Online learning*

Tracking-based semi-supervised learning has presented an additional challenge: it is easy enough to acquire millions of training instances that it may be necessary to consider new, online algorithms that are designed for it. Currently, the boosting algorithm of [10] and [9] requires all training examples to be loaded into main memory during the learning process. There are a number of tricks one could apply to alleviate this problem, such as random projections or hashing to compress the descriptors, but ultimately it seems that online learning algorithms will be desirable. To put rough numbers to the problem, less than ten hours worth of unlabeled data frequently results in exceeding 32GB of RAM to store the inducted training examples.

To consider lifelong learning, and especially learning using the data made available from *fleets* of robots, it will probably be necessary to use constant space, linear time training algorithms. The good news is that we can now consider algorithms that might fail miserably with small amounts of training data, but have the desired online properties and work acceptably with large amounts of training data.

C. *Bootstrapping*

The impact of object recognition methods that use laser range finders is currently limited due to the high price (tens of thousands of dollars) of the sensors. For indoor applications, the economical (∼$150) Microsoft Kinect is an exciting development, but is inapplicable to outdoor and longer-range systems. It might be possible, however, to bootstrap a more economical camera-based object recognition system using large quantities of automatically labeled objects produced from a single laser-based system.

Two decades ago, statistical machine translation became viable for the first time, at least partly due to the discovery of a new source of large quantities of labeled data - the French and English records of the proceedings of Canadian Parliament [4]; today, related methods are socially relevant in the form of online translation engines. One could imagine a similar situation in which extremely large quantities of labeled images produced by a single, expensive laser-based method could make certain vision-based applications viable.

Currently, the largest category of failures in Junior's object recognition are false negatives due to undersegmentation. For the task of building large sets of labeled images, these types of errors are less serious than false positives or false negatives that are directly caused by content of the object. Figure 4 shows an example of collecting overhead views of pedestrians automatically by using object recognition on Junior, then projecting these detections into the external camera above. While this system is not completely free of bias - seated pedestrians, for example, cannot be recognized because they cannot currently be segmented - it is likely to produce data that is complete enough to be of some use.

D. *Automated descriptor tuning*

In our recent work, all descriptors have been hand-tuned. This task is tedious, time consuming, and probably would not be done by non-robotics-experts, but is an important part of achieving good performance. When our hypothetical farmer buys a new robotic picking device, he will need to train it to recognize the particular fruit that he grows, and this may require variations on the existing descriptor set to do well.

As a result, there is a need for a method which will tune these parameters automatically. Automated feature selection has been well-studied in the machine learning literature, but for robotics applications we need one in particular that will respect the real-time operation of the system as well as the final accuracy, ideally while considering the caching structure of the descriptor pipeline.

ACKNOWLEDGMENTS

The authors would like to thank Mike Sokolsky for maintenance of the autonomous vehicle.

REFERENCES

[1] Video output of object recognition using tracking-based semi-supervised learning. URL http://cs.stanford.edu/people/teichman/rss2011.html.

[2] Karim Ali, David Hasler, and Francois Fleuret. Flowboost - appearance learning from sparsely annotated video. In *Computer Vision and Pattern Recognition*, 2011.

[3] Francis Bach, Gert Lanckriet, and Michael Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In *International Conference on Machine Learning*, 2004.

[4] Peter Brown, John Cocke, Stephen Della Pietra, Vincent Della Pietra, Fredrick Jelinek, John Lafferty, Robert Mercer, Paul Roossin, and Thomas Watson. A statistical approach to machine translation. *Journal of Computational Linguistics*, 16(2):79–85, 1990.

[5] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Special invited paper. additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, 28(2):337 – 374, 2000.

[6] Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Soeren Kammel, J Zico Kolter, Dirk Langer, Oliver Pink, Vaughan Pratt, Michael Sokolsky, Ganymed Stanek, David Stavens, Alex Teichman, Moritz Werling, and Sebastian Thrun. Towards Fully Autonomous Driving: Systems and Algorithms. In *Intelligent Vehicles Symposium*, 2011.

[7] U.S. Department of Transportation. Traffic safety facts 2009 (early edition).

[8] Luciano Spinello, Matthias Luber, and Kai O Arras. Tracking People in 3D Using a Bottom-Up Top-Down Detector. In *International Conference on Robotics and Automation*, 2011.

[9] Alex Teichman and Sebastian Thrun. Tracking-based semi-supervised learning. In *Robotics: Science and Systems*, 2011.

[10] Alex Teichman, Jesse Levinson, and Sebastian Thrun. Towards 3D object recognition via classification of arbitrary object tracks. In *International Conference on Robotics and Automation*, 2011.

[11] Chris Urmson. The google self-driving car project. Talk at Robotics: Science and Systems, 2011.