Traffic Sign Recognition for Computer Vision Project-Based Learning

David Gerónimo, Joan Serrat, Antonio M. López, *Member, IEEE*, and Ramon Baldrich

***Abstract*—This paper presents a graduate course project on computer vision. The aim of the project is to detect and recognize traffic signs in video sequences recorded by an on-board vehicle camera. This is a demanding problem, given that traffic sign recognition is one of the most challenging problems for driving assistance systems. Equally, it is motivating for the students given that it is a real-life problem. Furthermore, it gives them the opportunity to appreciate the difficulty of real-world vision problems and to assess the extent to which this problem can be solved by modern computer vision and pattern classification techniques taught in the classroom. The learning objectives of the course are introduced, as are the constraints imposed on its design, such as the diversity of students' background and the amount of time they and their instructors dedicate to the course. The paper also describes the course contents, schedule, and how the project-based learning approach is applied. The outcomes of the course are discussed, including both the students' marks and their personal feedback.**

***Index Terms*—Computer vision (CV), Master's degree, project-based learning (PBL), traffic sign.**

I. Introduction

COMPUTER vision (CV) [1] is a subfield of artificial intelligence (AI) aimed at understanding still images and video sequences; examples of this include recognizing people or objects, navigating in an environment, reconstructing the three-dimensional shape of a scene, or controlling a device (such as when a robotic arm grasps an object). It is an interdisciplinary field that draws from computer science, signal processing, and a number of mathematical fields like geometry, statistics, and algebra.

During the last two decades, CV has progressively been incorporated into both undergraduate and graduate programs of computer science studies. In their excellent review on CV education, Bebis *et al.* [2] review various approaches to teaching CV. Most of the literature is focused on different approaches to integrating the topic into undergraduate courses. The most common approach is to follow the traditional course structure in which the students are presented with theoretical lectures and have short practical assignments. Another approach is to use students' specific knowledge in a given area (computer graphics, image processing, etc.) and then introduce the new CV concepts on top. This approach assumes that the students already have knowledge of a given topic, which is convenient for undergraduate studies in which students have to follow a given subject itinerary. Another interesting method for teaching CV is through interactive technology. As an example, Reimer *et al.* [3] propose a tangible interface that accelerates the process of understanding the components and results of a vision system for students who have little experience in programming. However, while this can be a convenient approach for undergraduate students, being able to program and optimize the algorithms is a compulsory skill for a graduate student. Another widely used approach is to integrate the study of computer vision with another related topic, for example fusing robotics and CV in the same course [4]. Similarly, Hoover [5] proposes introducing computer vision topics into an undergraduate embedded systems course. In order to attract students to the field, the introduction of image computation in early programming classes has been proposed [6]. Finally, one attractive approach to having the students learn the topic is by having them carry out a CV project that solves a given problem. This methodology, usually referred to as project-based learning (PBL), has been used in secondary-level studies [7], engineering courses [8]–[14], and also graduate courses [15], this last one being an entry-level project like the one presented here. PBL has several advantages over the other approaches.

- It allows the students to gain real experience in a topic.
- It brings together theoretical and practical concepts with a single goal, which enhances students' motivation.
- It allows the students to discover these new concepts for themselves if the course is well designed.

Furthermore, PBL suits both undergraduate courses and graduate courses in which the concepts are more advanced and specialized.

A. Motivation

This paper presents a project-based assignment that is the core of the *Introduction to Computer Vision* course of the Master's program in CV and AI at the Universitat Autònoma de Barcelona (UAB), Bellaterra, Spain. In this course, the students learn the basics of computer vision: image formation, image processing, descriptors, classification, and learning theory. Subjects such as tracking, motion analysis, and stereo are left to an *Advanced CV* course that addresses them in depth. The course is taught during the first term together with *Introduction to AI*, and followed by either *Advanced CV* or *Advanced AI*

Manuscript received August 27, 2012; revised November 07, 2012; accepted December 09, 2012. Date of publication February 01, 2013; date of current version July 31, 2013. This work was supported by Dr. M. Vanrell, coordinator of the Master's Program of the Department of Computer Science, Universitat Autònoma de Barcelona, and Project Consolider-Ingenio 2010 CSD2007-00018. The work of D. Gerónimo, J. Serrat, and A. M. López was supported by the Project TRA2011-29454-C03-01. The work of R. Baldrich was supported by the Project TIN201021771-C02-1.

The authors are with the Computer Vision Center and the Department of Computer Science, Universitat Autònoma de Barcelona, 08193 Bellaterra, Spain (e-mail: dgeronimo@cvc.uab.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TE.2013.2239997

and a more transversal course called *Introduction to Research and Development* that conveys skills such as scientific writing, ethics, and R&D project management. After completing these courses, the students spend one term developing their Master's thesis under the guidance of an experienced advisor—either a university researcher or an employee of a computer vision company. Once the students have a solid knowledge of both theory and practice, they can then propose a thesis subject. For more details of the Master's program, the reader can refer to the Web site of the Master's program in Computer Vision and Artificial Intelligence at the UAB [16].

An introductory graduate course poses several specific challenges not encountered in a typical undergraduate course. First, the diversity of the students in terms of education and background in a graduate program is much wider than in an undergraduate course. The Master's is aimed at an international group who may have graduated in various disciplines, from mathematics to physics to computer science or electrical engineering. Hence, depending on their education and experience, they will be more or less comfortable with the mathematical foundations of computer vision, software development issues, or scientific programming. Second, the graduate students have two types of objectives: Some intend to pursue a Ph.D. program at the end of their Master's, while others just want an M.Sc. degree in computer vision in order to apply for positions in IT companies. The former are more interested in getting a sound understanding of the theoretical basis of the current state-of-the-art techniques, while the latter are interested in more practical matters. Third, although graduate students are usually more motivated than undergraduates, it is of key importance to present engaging assignments that match the aforementioned objectives.

B. Proposal

To the authors' knowledge, no literature exists that treats the use of PBL to meet the traditional requirements for CV education and the graduate-oriented challenges described above. In order to meet these challenges in the best way possible, a unique realistic medium-sized project is proposed here: traffic sign detection and the recognition of images extracted from video sequences recorded while driving in urban scenarios. Such a project requires the application of many of the low-level image processing techniques and the key components of an object detection/recognition system, that is, image local feature extraction for classifier learning and evaluation. The PBL approach presented here has three key factors that make it a very suitable strategy to address the challenges of the course.

- Although the traditional approach of using short exercises focused on particular parts of the syllabus can be an effective knowledge acquisition technique in many courses, PBL makes the students combine the concepts explained in theory lectures, improving their understanding of their application to real problems. This is an important factor in CV students' curricula. From the outset, the proposed project tackles a complex real problem, so every new algorithm implemented to extend the system is seen by the students as a way to improve *their* system, which ensures high levels of interest and engagement. By the end of the term, each group has a different and unique system.
- The project begins with only a basic skeleton structure and simple statements. Students thus gain a deeper understanding of the project and take incremental steps of progressive complexity. Each of these steps is based on concepts explained in the theory lectures.
- The project's being of medium size helps to maintain a balance between giving the students a well-defined plan (i.e., what, when, and how to implement the mandatory techniques) and giving them the freedom to implement novel, state-of-the-art techniques in the literature; e.g., proposing improvements to boost the system performance.

A project designed for an introductory course such as that presented in this paper differs from a traditional final-year project in two aspects. First, the techniques to be implemented in an introductory course are more constrained, given that the planning and modules are predefined. Second, the system's behavior in terms of detection performance and computational time is not considered when assigning marks, but it is an implicit aspect to be taken into account, as will be seen later. The aim is that the students explore and implement the techniques explained in theory classes and realize that these can be combined to solve a real problem. This is in contrast to focusing on a high-performance *commercial-type* product, as in final-year projects.

This paper details how the project has been designed in terms of schedule, material, and evaluation and discusses the relevant outcomes of the course during the 2011–2012 term. Following on from previous project-based-learning literature [8], this paper demonstrates that PBL is an effective strategy for teaching CV to graduate students.

The remainder of this paper is organized as follows. Section II describes the technical details of the project to be developed by the students, the weekly assignments, the milestones, datasets, etc., together with the assessment methodology. Section III summarizes the outcomes, i.e., how the course evolved in its last offering in 2011–2012 and the feedback received from students. Finally, conclusions and areas for improvement in next year's course are discussed in Section IV.

II. Traffic Sign Recognition Project

Traffic sign recognition (TSR), through vision or other sensors, is one of the applications belonging to the so-called advanced driving assistance systems (ADAS) field, which aims to improve driving safety by warning the driver of potential hazards [17]–[19]. The first commercial TSR systems to recognize speed limits appeared in high-end cars as recently as 2009. However, the solution to the problem of reliably recognizing all types of traffic signs under varying ambient lighting and weather conditions (daytime, nighttime, rain, fog, high/low beams, etc.) and different road types remains elusive. Modern methods, similar to other ADAS tasks [20], adopt a divide-and-conquer approach: first detection, then recognition. The output of the detection stage is a list of image bounding boxes, each containing a yet unrecognized traffic sign, while that of the recognition stage is a class label for each one ("stop," "give way," etc.), as illustrated in Fig. 1.

There are two main aspects that make TSR an ideal topic for a PBL strategy. First, students are motivated by a present-day technological problem, the solution to which can literally save

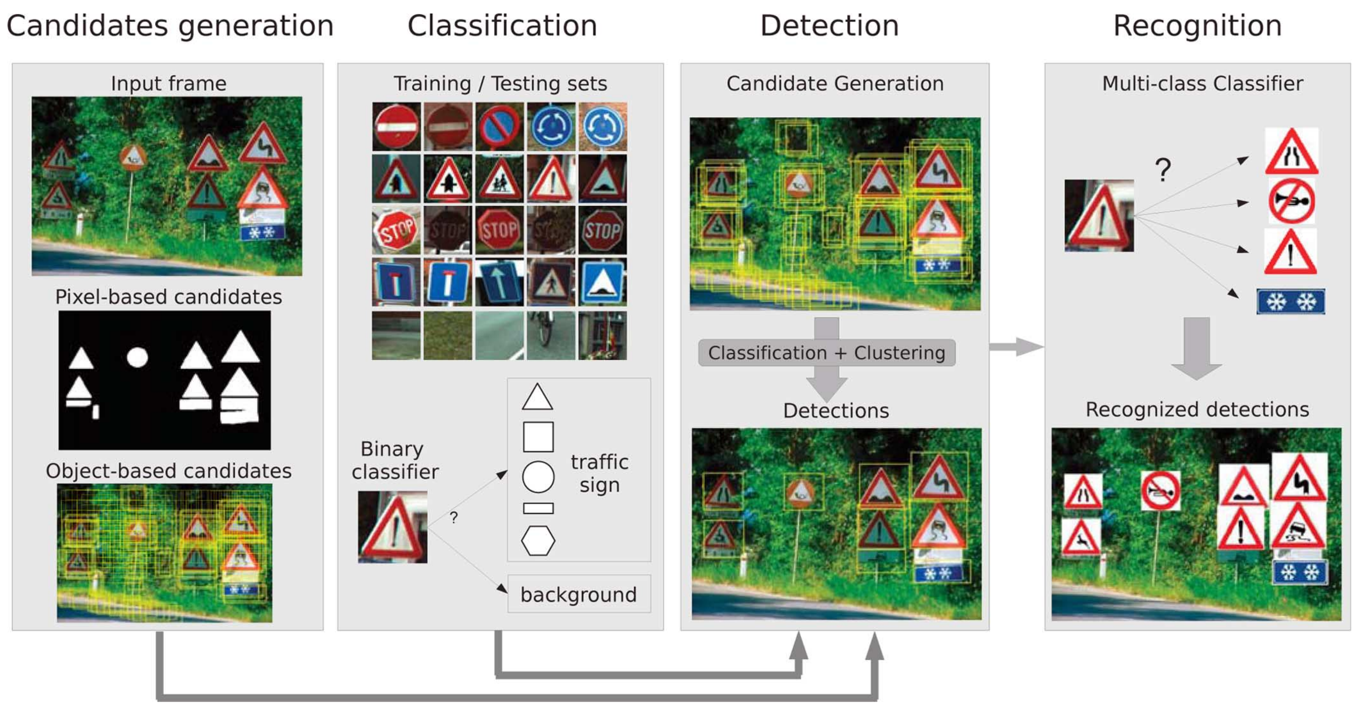

Fig. 1. Four project milestones of the traffic sign recognition prototype. Candidate generation of bounding boxes and the learned binary sign/background classifier are combined in order to create a detection subsystem. A multiclass sign classifier takes the positive bounding boxes and assigns a traffic sign label to each one.

lives. Furthermore, being a current research problem, many of the techniques to be tested (e.g., features, candidate generation algorithms, etc.) are not in textbooks, but rather in recent technical articles. This trains the students to acquire knowledge from other sources. Lecturers should also suggest up-to-date textbooks, mostly in the machine learning area [21], [22], which may be consulted for some of the algorithms. Second, given the modular nature of such a recognition system, the students are encouraged to incorporate new concepts in their solutions in a progressive manner, putting special emphasis on the path from intuitive solutions to complex state-of-the-art ones. In addition, the modular nature of the proposed project (containing algorithms for many different tasks, depending on the system stage) also allows the project to cover most of the concepts studied in the theory lectures.

Object detection problems other than TSR (e.g., pedestrian/vehicle detection, surveillance, visual object retrieval) would also fulfill the two aspects mentioned above. However, human-detection applications such as pedestrian detection or surveillance would be too challenging for entry-level students. Visual object retrieval is a compatible option, but TSR is by far more attractive: The students appreciate the benefits and potential of TSR in real life, which provides extra motivation and encouragement.

A. Schedule

The course is given over one 12-week term, with two theory lectures of 2 h and one project session of 1 h per week. Students organize themselves in teams of three and must attend all the sessions. The project development has four milestones, set at the start, corresponding to the increasing functionality of the software prototype for TSR. During the first week's project session, the problem is introduced, the MATLAB framework and datasets to be used to develop the project are provided (see the Appendix and Section II-C), and the goals (homework) for the next week are stated, as will be done each week. Note that the instructor does not guide the teams precisely on how to achieve the goals; instead, they are told *what* the goals are and are given some hints and explanations on the algorithms they may use. Furthermore, the theory lectures are synchronized with the project milestones, so the weekly homework corresponds with the required background study.

Next, the four milestones are presented, which are illustrated in Fig. 1 and detailed on the project Web site [23].

- *Candidate generation* (Weeks 2–4). Given an input image, candidate generation aims to propose bounding boxes that are likely to contain a traffic sign of some type. A bottom-up approach using the distinctive colors and geometric shapes of the signs is first suggested, which makes students learn how to apply useful image processing operators like color space conversions and thresholding. Given the variability of images, some of the approaches will most likely fail, so the students are advised to use a top-down sliding windows algorithm (scanning the image with all the possible candidate windows).
- *Binary classification* (Weeks 5–8). Candidate generation usually produces false positives together with the correctly framed signs, which introduces the need for a more sophisticated filtering step. The idea is that the students start by using very intuitive descriptors (e.g., mean grayscale) and simple classifiers (linear discriminant analysis) and end by testing complex features (e.g., histograms of oriented gradients, Haar-like features, etc.) and learning machines (support vector machines).

- *Detection system* (Week 9). A detection system combines candidate generation and binary classification together with clustering (e.g., Mean Shift) to group overlapping positive candidates as single detections. At this milestone, the students only have one supervision session and must report results on whole video sequences after a 2-week break. Developing the system is an enjoyable task they can carry out on their own.
- *Recognition system* (Weeks 10–12). The final step consists in upgrading from traffic sign detection to sign recognition: deciding which specific sign ("stop," "give way," etc.) is contained in each positive detection. The simplest strategy is to apply the same binary classification algorithms as those used in step two in the context of one versus all multiclass classification, though more elaborate strategies like error correcting output codes (ECOCs) are also tested by fast-paced teams. Finally, the students have to test the whole system on a whole video sequence, just as they did with the detection system.

B. Weekly Work

During the course, 1 h per week is dedicated to monitoring and giving guidance to the otherwise independent student teams. In the first part of the session (40–45 min), the results obtained in the previous assignment are analyzed by the instructor. He or she will provide a summary of the common problems, the most interesting proposals, and a statistical performance of the teams. At this point, there will also be short presentations by the teams, submitted via e-mail beforehand, consisting of three slides summarizing the techniques implemented, the results, and the problems encountered. Two randomly chosen teams present their work (it is not mandatory that all the team members speak). This serves to facilitate communication among the groups and allows the instructor to spend the majority of this part of the session studying in greater depth both aspects specific to a particular team and aspects applicable to all. The instructor ensures a balance between the interest value of the presentation for the entire class and the frequency with which each team presents their work. The rest of the groups also participate with comments and questions, not only on the presenting team's work, but also with respect to their own systems. The submitted work is not evaluated weekly, but at the milestone checkpoints (see Section II-C).

In the second part of the session (15–20 min), the homework for the next week is explained. Usually, this is quite straightforward: The main ideas plus the evaluation criteria are explained, and since the required theoretical concepts are still fresh from the recent lectures, there are few questions. If any concept has not been explained (e.g., sliding windows), it is described in detail in this part of the session. In addition, the instructor gives special instructions on the algorithms that are particularly complicated to implement (e.g., integral image). Finally, it is worth noting that during the semester, the instructor has one additional hour per week for answering questions from individual groups or students, during which they can analyze their systems in greater depth.

Fig. 2. Samples from seven of the 14 traffic sign types learned by the multiclass classifier for sign recognition.

C. Dataset

The students are provided with a dataset of images both to facilitate the testing of each milestone and to serve as a common benchmark for all teams. In addition, they are freed from creating a manually annotated ground-truth set for detection and recognition, which would be very time-consuming. The dataset used for the project is based on the KUL traffic sign database [24], [25] created by Dr. Radu Timofte from the Katholieke Universiteit of Leuven, Leuven, Belgium. The dataset was adapted to the specific needs of the project, selecting the number of images and those traffic sign types with more samples. Three subsets are provided to the students:

- *Detection set:* 669 images of 1628×1236 pixels, with annotated ground-truth (bounding boxes and type of traffic signs present in the image) used in the candidates generation and detection/recognition systems milestones;
- *Classification set*: cropped images to be used for the binary classifier training and testing. In total, 3000 images in five groups (circles, inverted triangles, rectangles, octagons and triangles) plus a background subset;
- *Recognition set:* This is divided into training and testing and is used to build the multiclass classifier that recognizes the traffic sign type. The set is limited to 14 types of signs (Fig. 2), those with more samples. The size of this set is 1225 images.

The final recognition system, in addition to being evaluated on the detection set, is tested on a short sequence (200 frames) from the KUL dataset and on a challenging sequence (185 frames) from the CVC-02 pedestrian dataset [26]. The latter was recorded with a different camera in a different country (Spain) and thus contains slightly different signs and imaging conditions.

D. Evaluation

The project is graded as follows: 20% for each of the four milestone deliverables, 10% for two oral presentations, and 10% for peer evaluation. This amounts to 55% of the total course score, the remaining 45% being determined with a written exam.

Teams are evaluated at the four project milestones based on their deliverables. These consist of a 10-page report plus the source code of the corresponding project part. Each weekly assignment is graded with an E (not submitted), D (failed), C (meets mandatory requirements), B, or A (exceeds mandatory requirements).[1] During the first weeks, it is the instructor who suggests ways to go beyond the mandatory work. Subsequently, the students propose their own approaches, whose complexity and originality will determine their grade. As an

[1]For inclusion in their total marks for the course, students' project grades are converted to marks as: $A = 10$, $B = 8$, $C = 6$, $D = 4$, and $E = 0$.

example, the homework in week 8 consists of testing the performance of HOG + SVM on the classification dataset, which is the minimum requirement to get a C. Getting an A or a B depends on students' incorporating proposed improvements: the use of different color spaces to compute the HOG descriptor would be graded with a B, while the implementation of different visual features—e.g., local binary pattern (LBP), shape context, distance to border (DtB), or scale invariant feature transform (SIFT)—or the implementation of multipart classification would be graded with an A.

One of the course objectives is to have the students explore many of the different techniques explained in the theory lectures. The idea of evaluating the teams exclusively according to practical issues like fast processing and detection rates was discarded. In fact, the computation time restriction is naturally limited by their having to present their results weekly, which forces them to optimize their algorithms. In addition, simply providing the teams' performance statistics fosters competition and thereby boosts variety of the solutions.

In the first deliverable, the length or organization of the report is not evaluated. Instead, the idea is to make the students explain and assess the implemented techniques. From the second milestone on, once students have already settled their frameworks and feel comfortable with the technical part, technical writing is also evaluated: document organization, comprehensiveness, style, references, and synthesis. Some of these concepts are taught in the parallel Master's course *Introduction to R&D* [16]. The students are provided with a set of technical writing guidelines and a LaTeX template.

Each team has to make two 10-min oral presentations, with a further 5 min for discussion during which all the members are required to speak. The first presentation is on the candidate generation milestone and the second on the detection or the recognition part. These presentations serve to evaluate two different skills necessary for any engineer: technical soundness and communication. The technical aspects are assessed by the instructor, while the oral presentation aspects (vocabulary, grammar, expression) are assessed by an English language instructor. Each member of the group has to present a part of their work, but the final mark is not set individually.

Finally, each student is provided with a confidential form to assess the contribution to the project of each of his/her team partners. This consists of simple, direct, unambiguous, multiple-choice statements such as *Student X worked more/same/less than me*. In this way, the chances of getting an irrelevant or biased evaluation is minimized. This evaluation is also useful to identify students who do not contribute sufficiently to the development of the project, in which case they can be penalized in their marks or have an individual evaluation.

III. Discussion

This section presents the outcomes of the course for the 2011–2012 year. They are divided as follows: an analysis of grades and student feedback, remarks from the instructors that are likely to be useful for educators willing to implement similar courses in their institutions, and issues regarding the students' performance in the theory exam.

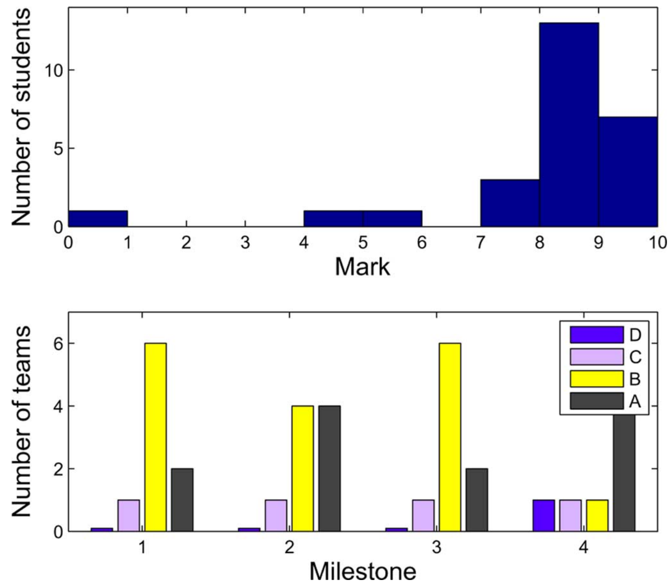

Fig. 3. *(top)* Distribution of the students' marks after taking into account deliverables, oral presentations, and intragroup evaluation. *(bottom)* Distribution of the grades (see legend) of the deliverables at each milestone.

A. Marks

Fig. 3 (top) shows the distribution of the final student marks according to the weighting specified in Section II-D. As can be seen, the marks are high (a mean average of 8.42 and a standard deviation of 1.08 over 10), which means that in general the teams fulfilled most of the requirements and proposed interesting improvements. Fig. 3 (bottom) illustrates the increase in most teams' grades during the four stages. This increase is due to two factors. First, the students initially had to adapt to the weekly milestones scheme, the working framework, and the language (while almost everybody had medium or high programming skills, 44% of the students had little experience in MATLAB). Second, they required the first evaluation to understand the type and extent of improvements they had to implement in order to obtain the highest marks.

B. Student Feedback

The feedback from the students was very good. They were engaged by the weekly assignments and enjoyed the project in spite of the amount of work it represented. An indication that the course succeeded in motivating students is that several of them changed their enrolment from *Advanced AI* to the *Advanced CV* course for the second term of the year.

At the end of the course, the students were asked to fill in an anonymous assessment form on the project. Based on this form and on personal communication, some interesting issues are worth highlighting. On the one hand, the students valued the continuity of the project, its relevance to the theory lectures (68%), and the appropriate level of difficulty (87%), the latter being one of the main concerns during its design. On the other hand, while the discussion sessions were very highly valued with respect to their usefulness in developing the project (93%), some students complained of insufficient time being dedicated to the explanation of the weekly homework and the subgoals during these sessions (37%). Keeping an appropriate balance between the time spent in discussion and homework explanations is difficult. In fact, the better-performing teams wanted more discussion on the improvements presented by other groups, which could help them to learn new techniques and methods for their experiments. Other teams preferred a longer and more detailed explanation of the mandatory homework to be done by the next week.

Another popular complaint concerned the ambiguity when evaluating the work, specifically the difference between A and B grades. While the students were informed at the beginning of the course that the grading policy would take into account additional improvements (i.e., a subjective mark depending on the complexity), some teams who implemented minimum improvements were not satisfied with their marks being lower than those for teams with much more relevant proposals. This was rapidly amended by stating specific objectives to be met to achieve the possible improvements. For instance, in week 9 (sign detection task), the number of required implementations of candidate generation and classification algorithms for the B and C grades was specified, instead of leaving it open, and A was reserved for very original proposals, such as the use of a new feature descriptor or using part-based classification.

The students also provided information about the time they spent on the project each week: 6% spent from 5 to 10 h, 38% from 10 to 15 h, 25% from 15 to 20 h, and 31% spent more than 20 h a week on the project. Given that the whole course is worth 10 ECTS (25 hours per ECTS credit), which corresponds to around 50 h of lectures and 200 h of work out of class, these figures suggest an appropriate correspondence between the ECTS available and the students' effort.

C. Instructors' Remarks

The oral presentations were too tiring for both instructors and students due to the number of teams, the limited time, and the similar content being presented, in addition to the instructors being too generous in presentation timing. This should be amended in future offerings.

Specifying a clear penalty when deadlines are not met is also an important point. Given that there are fewer groups than in an undergraduate course, the due dates were sometimes extended. By clearly stating the penalty (e.g., late submissions are not accepted), evaluation becomes easier for the instructor and fairer to the students.

Finally, in future courses, it would be desirable to go into more detail on topics that were only touched upon, such as the suitability of the different evaluation measurements, or the analysis of performance curves and limit the treatment of others, such as template matching or linear discriminant analysis.

D. Theory Exam

Although the students became very capable of analyzing a real problem and solving it (managing overfitting, dealing with ground-truth limitations, researching current literature, etc.), their marks in the theory exam were unexpectedly low (a mean average of 4.53 with a standard deviation of 1.87) and had low correlation with the project marks (0.3). While this paper focuses on a PBL project and not on the theoretical part of the course, it is worth analyzing the ways in which the project could be modified to improve the students' performance in the exam.

First, it is clear that the project is more engaging to students than are theoretical studies, but it is also time-consuming. Therefore, at the end of each week, they may have greatly improved their system, but also may have ignored the theoretical part of the course to some extent. Second, some of the answers in the exam suggest that although the students understand the techniques while implementing them for the project, they do not really understand their relevance in spite of their being highly related to the theory. While it is virtually essential to put theory into practice in order to master it, it is also important to ensure that the theoretical concepts behind the practical applications are understood in order to obtain a solid knowledge of the subject. These problems could be resolved in part by eliminating elements of the project, such as assignments in the candidate generation or classification stages, and/or some of the reports or oral presentations. This would not affect the structure and attractiveness of the project and would allow the inclusion of midterm theory exams. This would not only leave students more time for studying, but would also motivate them to learn the theory behind the implemented techniques, which is what will be evaluated in the exam. Furthermore, the inclusion of some project-related problems focused on the theoretical foundations of the implemented techniques would also encourage the students to study more theory. These problems could also be used as an additional part of the evaluation.

IV. Conclusion

A project on traffic sign detection and recognition for an introductory computer vision graduate course has been presented. The outcomes of the 2011–2012 year were excellent: The students were satisfied both with their results and with the variety of techniques they had put into practice, most of them related to the theory lectures. By the end of the course, they have developed a relatively complex vision system, having researched numerous CV fields, and written a number of scientific and technical documents.

The three main aspects that should be considered by educators aiming to implement similar PBL approaches at graduate level are as follows.

- First, an analysis of the course syllabus and the skills to be acquired by the students is crucial when deciding if PBL is the most convenient approach. The ideal case involves a wide syllabus containing concepts with direct implementation, in which practical experience of programming and teamwork are basic required skills. On the contrary, courses with very theoretical or specialized syllabi should be implemented using approaches more focused on exams or with a number of separate practical assignments, respectively.
- Second, current-day real-life problems not only motivate the students, with a subsequent improvement of their expected results, but also emphasize the need to understand the theoretical concepts to develop the project. In addition, they also require the students to read research articles in addition to textbooks, a desirable skill for graduate students.

TABLE I
THEORY AND PROJECT CONTENTS OF THE COURSE. PERFEVALPIX AND PERFEVALOBJ ARE INTRODUCED IN WEEKS 2 AND 3 AND THEN USED THROUGHOUT THE COURSE

Week	Lecture	Project Topic	Techniques to Implement	Provided Code
1	Color Im. Form., Low Level Image Rep.	-	-	-
2	Geometric Transforms, Image Feats.	Color / Morphology	Color spaces (RGB,Opp,LAB...)	`colorspace, PerfEvalPix*`
3	Local Image Descriptors	Windows Generation	CCL, Sliding Windows	`PerfEvalObj*`
4	Bayesian Theory, Statistical Learning	Geometric Heuristics	Hough, Circular Hough	`CircHough`
5	Dimension. Reduction, Expect. Max.	Template Matching	Chamfer Dist	-
6	Support Vector Machines	Global Descriptors	Gradient Orientation, LDA	`LDA`
7	Markov Random Fields (MRF)	Local Descriptors I	Haar, SVM	`SVM`
8	Multiclass, Inference on MRFs	Local Descriptors II	HOG, SVM	`HOG`
9	Pattern Recognition in Practice	Sign Detection	MeanShift	`MeanShiftClustering`
10	Ensemble Class., Multiple Instance	Multiclass	One versus All	-
11	Kernel Methods	Sign Recognition	Full system	-

- Finally, planning the project so that the students encounter the concepts naturally, as they carry it out, is the most suitable approach. Most of the systems can be divided into different stages that can be used as guidelines to plan and schedule the theoretical lectures. This aspect is perhaps the most difficult to be implemented: to make the first algorithms based on educated guesses, while the new solutions are built on top of their predecessors. PBL represents the most convenient approach to tackle this aspect, but the instructors have to take into account the characteristics of the student demographic (e.g., different backgrounds), the course level (e.g., degree), and the coverage of the theory contents in order to tailor the course's design to the specific requirements.

The complete project materials have been made public on the project Web site [23]. They may serve either as an inspiring resource for other educators together with the experience described in this paoer, or they can be directly applied not only in other graduate CV courses, but also in pattern recognition or machine learning courses, just as when adopting a textbook as a course guide. The proposed project is the result of the authors' experience during 5 years in graduate education, the thorough design of the plan, templates, rubrics, and datasets to implement such a project. Supporting the effectiveness of this project design, these materials were successfully used by other instructors from the same institution in the 2012–2013 offering of the course, who found the preparatory work to have been much less than expected thanks to the materials provided and the detailed course organization.

APPENDIX

Material publicly available to students and instructors [16]:

- project assignment;
- project planning and contents;
- templates and documents: progress slides, report, intragroup evaluation;
- rubrics: reports and oral presentations;
- source code: MATLAB skeletons (detector and classifier), third-part MATLAB code from MathWorks and other sources (Circular Hough, Colorpsace, Harris, HOG, LDA, LibSVM, MeanShift);
- datasets: detection, classification, recognition, and video sequences (based on KUL Dataset [24], [25]).

Table I lists the contents of the theory classes and the project (note that week 12 is reserved for study time for the exam).

ACKNOWLEDGMENT

The authors would like to thank Dr. R. Timofte for making the KUL Belgium Traffic Signs and Classification Benchmark datasets publicly available. These datasets were subsequently adapted for teaching purposes.

REFERENCES

[1] R. Szeliski, *Computer Vision: Algorithms and Applications*. New York, NY, USA: Springer, 2010.
[2] G. Bebis, D. Egbert, and M. Shah, "Review of computer vision education," *IEEE Trans. Educ.*, vol. 46, no. 1, pp. 2–21, Feb. 2003.
[3] P. Reimer, A. Albu, and G. Tzanetakis, "Raydiance: A tangible interface for teaching computer vision," in *Proc. 7th ISVC*, Las Vegas, NV, USA, 2011, vol. 2, pp. 259–269.
[4] G. Cielniak, N. Bellotto, and T. Duckett, "Integrating mobile robotics and vision with undergraduate computer science," *IEEE Trans. Educ.*, vol. 56, no. 1, pp. 48–53, Feb. 2013.
[5] A. Hoover, "Computer vision in undergraduate education: Modern embedded computing," *IEEE Trans. Educ.*, vol. 46, no. 2, pp. 235–240, May 2003.
[6] C. Olson, "Encouraging the development of undergraduate researchers in computer vision," *ACM SIGCSE Bull.*, vol. 44, no. 2, pp. 255–259, 2006.
[7] R. Capraro and S. Slough, *Project-Based Learning*. Rotterdam, The Netherlands: Sense, 2009.
[8] L. Fernandez-Samaca and J. Ramírez, "An approach to applying project-based learning in engineering courses," in *Proc. IEEE ANDESCON*, Bototá, Colombia, 2010, pp. 1–6.
[9] A. Price, R. Rimington, M. T. Chew, and S. Demidenko, "Project-based learning in robotics and electronics in undergraduate engineering program setting," in *Proc. IEEE Int. Symp. Electron. Design, Test, Appl.*, Ho Chi Minh City, Vietnam, 2010, pp. 188–193.
[10] R. Hsu, "Project based learning as a pedagogical tool for embedded system education," in *Proc. 3rd Int. Conf. Inf. Technol., Res. Educ.*, Hsinchu, Taiwan, 2005, pp. 362–366.
[11] J. Macias-Guarasa, "A project-based learning approach to design electronic systems curricula," *IEEE Trans. Educ.*, vol. 49, no. 3, pp. 389–397, Aug. 2006.
[12] J. Venkataraman, "Project based electromagnetics education," in *Proc. Appl. Electron. Conf.*, 2009, pp. 1–4.
[13] A. Breiter, G. Fey, and R. Drechsler, "Project-based learning in student team in computer science education," *Electron. Energetics*, vol. 18, no. 2, pp. 165–180, 2005.
[14] F. Martinez, L. Herrero, and S. de Pablo, "Project-based learning and rubrics in the teaching of power supplies and photovoltaic electricity," *IEEE Trans. Educ.*, vol. 54, no. 1, pp. 87–96, Feb. 2011.
[15] J. Nelson, "Using project-based assignments in graduate-level digital signal processing course," in *Proc. Digital Signal Process. Workshop & 4th IEEE Signal Process. Educ. Workshop*, 2006, pp. 135–140.
[16] Universitat Autònoma de Barcelona, Bellaterra, Spain, "Website of the UAB Master on Computer Vision and Artificial Intelligence," 2012 [Online]. Available: http://mcvai.uab.cat
[17] J. Khan, S. Bhuiyan, and R. Adhami, "Image segmentation and shape analysis for road-sign detection," *IEEE Trans. Intell. Transport. Syst.*, vol. 12, no. 1, pp. 83–96, Mar. 2011.

[18] S. Maldonado-Bascón, S. Lafuente-Arroyo, P. Gil-Jiménez, H. Gómez-Moreno, and F. López-Ferreras, “Road-sign detection and recognition based on support vector machines,” *IEEE Trans. Intell. Transport. Syst.*, vol. 8, no. 2, pp. 264–278, Jun. 2007.
[19] A. de la Escalera, J. Armingol, and M. Mata, “Traffic sign recognition and analysis for intelligence vehicles,” *Image Vision Comput.*, vol. 21, pp. 247–258, 2003.
[20] D. Gerónimo, A. López, A. Sappa, and T. Graf, “Survey of pedestrian detection for advanced driver assistance systems,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 7, pp. 1239–1258, Jul. 2010.
[21] C. Bishop*, Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.
[22] D. Forsyth*, Computer Vision: A Modern Approach*. Upper Saddle River, NJ, USA: Prentice Hall, 2011.
[23] D. Gerónimo, “M.Sc. project Website 2012,” 2012 [Online]. Available: http://mcvai.uab.cat/content/mscproject2012
[24] R. Timofte, “KUL Belgium traffic signs and classification benchmark datasets,” 2012 [Online]. Available: http://homes.esat.kuleuven.be/~rtimofte
[25] R. Timofte, K. Zimmermann, and L. Van Gool, “Multi-view traffic sign detection, recognition, and 3D localisation,” *Mach. Vision Appl.*, 2011.
[26] D. Gerónimo, “CVC-02 pedestrian dataset,” 2009 [Online]. Available: http://www.cvc.uab.es/adas/datasets

David Gerónimo received the B.Sc. degree in computer science and M.Sc. degree and Ph.D. degree (with honors) in computer vision from the Universitat Autònoma de Barcelona (UAB), Cerdanyola, Spain, in 2004, 2006, and 2010, respectively.

He is currently a Postdoctoral Researcher and Associate Lecturer with the Computer Science Department, UAB, and the Computer Vision Center. His research interests include object detection, recognition and tracking, feature detection and description, and scene understanding.

Joan Serrat received the Ph.D. degree in computer science from the Universitat Autònoma de Barcelona (UAB), Bellaterra, Spain, in 1990.

He is currently an Associate Professor with the Computer Science Department, UAB, and a member of the Computer Vision Center. At UAB, he is in charge of two undergraduate courses on Software Engineering and Image Processing and co-coordinates the introductory module of the Master on Computer Vision. He has coauthored more than 40 papers and three patents. His current research interest is the application of probabilistic graphical models to computer vision problems such as feature matching, tracking, and video alignment.

Dr. Serrat has been a Board Member of the Spanish Chapter of the International Association for Pattern Recognition.

Antonio M. López (M’10) received the B.Sc. degree in computer science from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 1992, and the M.Sc. and Ph.D. degrees in image processing and artificial intelligence from the Universitat Autònoma de Barcelona (UAB), Bellaterra, Spain, in 1994 and 2000, respectively.

Since 1992, he has been giving lectures with the Computer Science Department, UAB, where he currently is an Associate Professor. In 1996, he participated in the foundation of the Computer Vision Center at the UAB, where he has held different institutional responsibilities, currently leading the research group on advanced driver assistance systems (ADAS) by computer vision. He has led public and private projects and is a coauthor of more than 100 papers, all in the field of computer vision.

Ramon Baldrich received the Ph.D. degree in computer science from the Universitat Autònoma de Barcelona (UAB), Bellaterra, Spain, in 2001.

He is an Associate Professor with the Computer Science Department, UAB, and is attached to the Computer Vision Center as a Researcher. His research interests are mainly focused in color treatment in computer vision problems, including color constancy, color induction, and image shadows.