Automated Parking Lot Activity Analysis

Seth Polsley
Sketch Recognition Lab
Texas A&M University
spolsley@tamu.edu

Abstract

Automated parking lot analysis is concerned with the development of algorithms and software systems that can perform unsupervised monitoring of parking lots. These systems offer convenience to drivers and powerful data analytics options to business owners and city planners in the form of parking space availability tracking, peak usage detection, and more. Unfortunately, commercial solutions are sensor-based, making them largely cost-prohibitive because of individual sensor installation and maintenance costs. In this work, the authors explore the application of computer vision to the field of parking lot analysis and introduce a simple system based on the combination of pattern recognition and heuristic techniques. Though not yet as accurate as a human observer, they show that such systems have enormous potential to the field, especially given their easy installation at little to no cost. Considering all of these benefits and the ongoing efforts to improve recognition rates in the computer vision community, such systems will soon be a realistic commercial option for parking lot analysis.

1. Introduction

1.1. Definition

Automated parking lot analysis describes a computer system capable of tracking and presenting, in real-time, information about parking space usage and availability. These systems have been provided some known information, like parking space location or a lot's layout, and then they provide based on that knowledge additional metrics, like open parking spaces or lot capacity. Parking tracking systems are growing in popularity, and there are multiple reasons pushing their adoption.

1.2. Motivation

The first significant reason for adoption of such systems is convenience for drivers. In large cities, near businesses where parking is limited, or university campuses, drivers often spend a lot of time searching for good parking spots. There is a balance between spots a driver considers acceptably close to his or her destination but that are reasonably accessible and easily able to be checked for availability. However, it is very difficult to achieve this balance, and many drivers end up circling blocks or parking lots in order to scan for open spots near their stop. This in turn causes more traffic that can back up parking further. The strategy also does not guarantee that the earliest one to a lot obtains the first open space, as that driver may be circling on the wrong side of the parking zone when a spot becomes available. An automated system that tracks parking availability could provide users an enormous increase in convenience by always showing drivers the nearest open spots to their goal. This does not ensure they will always get the best space, but it can significantly reduce circling and traffic congestion in addition boosting efficiency by getting people parked faster.

Another major reason for adoption of these systems is the applications to big data. In today's information age, data is collected and analyzed for everything possible. The goal of such analysis is to find the best improvements. In the case of parking lot analysis, this comes in the form of metrics for parking lot or business owners, civil engineers, city planners, and more. Such metrics would be peak usage periods, most desired spaces, which businesses consistently bring in the most traffic, or what times do certain businesses have the most costumers. Businesses have already shown an interest in these metrics by adopting technologies like door counters, which drive analytics that help managers choose the best times to schedule shift changes and the like. By collecting this data automatically at the parking lot itself, business owners have even more information to analyze. Further, valuable knowledge about lot quality can be obtained. For instance, if a certain region of the parking lot draws the most parking, it can be identified early as a place where maintenance should be performed next. In this scenario, the data applications of parking lot analysis and user convenience combine because a lot owner can find the optimal time to do maintenance, and those closures can be supplied to drivers in real-time.

1.3. Existing Limitations

With so many applications for parking lot data, it is easy to see why there is such a growing interest in the area today. Unfortunately, current commercial solutions to this problem suffer from several limitations. All commercial methods are based on sensor networks, expanded a single space at a time via installation of some sensor. In addition to the initial cost of adding a sensor on a space-by-space level, there are further ongoing concerns such as network maintenance, since each device much have a connection to the network, and battery tracking to replace or repair dead sensors. Ultimately, when considering deploying at a large scale, these solutions incur large overhead costs.

A very recent solution to the problem that is being studied primarily in academia is to use computer vision. By deploying recognition algorithms on existing surveillance systems, like lamp post cameras in parking lots, computer vision software could solve almost all the existing problems. These systems would be easy to install, massively-scalable, and essentially maintenance-free. In this work, we introduce the implementation of a simple computer-vision based parking lot activity analyzer that combines pattern recognition and heuristic techniques. The next section provides some background work before delving into implementation details. The paper closes with a discussion of results and major findings.

2. Previous Works

2.1. Commercial Solutions

As mentioned previously, most existing solutions to parking lot analysis are sensor-based. Commercial solutions include Nedap Identification's *SENSIT* technology[1], *Streetline* sensors[2], *Proxel*'s electromagnetic detectors[3], *Xvision Systems*' approach[4], and more. Most of these use infrared or some related method for detecting proximity of a car. When a car is determined to be in the space, that information is shared over the network to the controlling software.

Computer vision approaches differ from sensor-grid ones in a multitude of ways. First is the concern of physical installation of each sensor associated with sensor approaches. Many of the aforementioned sensors require permanent installment in the pavement that is not inexpensive. Even assuming all the sensors have been placed correctly and can accurately detect when cars are present, they must each have an individual connection to the network, otherwise the information they gather will be useless. This requires the installation of a new network or, possibly, extension of an existing one. Either way, after all of these initial setup costs, ongoing costs do not subside. There is the maintenance of each sensor. If one becomes faulty or its battery has died, it must be replaced, or if the installation method makes replacement difficult, it must be repaired, which could potentially be more expensive. Generally, the sensor-grid approach to parking lot tracking seems inferior to a computer vision one in nearly every way. By using existing surveillance systems, computer vision software could remove the need for individual-space installments as well as any ongoing maintenance costs. The only drawback to these systems is the existing level of recognition accuracy. When sensors are working properly, the sensor-based networks are completely accurate, whereas computer vision software may be prone to mistakes. However, researchers are beginning to study this problem in more detail, actively taking steps to improve the reliability of vision methods applied to this domain.

2.2. Research Solutions

Even much of the recent research in parking lot management is centered are sensor-based approaches in some way. Be it through gate sensors, individual space sensors, some sort of regionally-based wireless sensor, or other technology, multiple means have been explored in publications and patents [1][5][6][9][10][13].

Within the past decade or so, there has been more interest in camera-based systems relying on computer vision. In 1998, Wang and Hansen developed a system for analyzing parking lots using aerial imagery [11]. Although not directly applicable to modern parking lot surveillance, their work demonstrates the seed of the idea. Some work has been published specifically targeting industrial management [4][14], although these approaches also have limitations, such as only being tested on miniature-scale lots with fake cars. One of the more promising recent approaches was released in the form of a patent by Winter and Osterweil which is entirely camera-based for ordinary analysis [12]. Unfortunately, their method does have some installation cost in the form of requiring an initial 3D model of the parking lot to be monitored. Lin et al. developed a useful vision-based system which was field tested; however, their approach placed strict requirements on the camera layout given to the system [7]. While it is one of the least restrictive methods developed, forcing camera layout does carry implications in regards to deployment on existing lamp post surveillance cameras.

Two other papers worth mentioning have not been published yet to a peer-reviewed journal. Instead, they are papers written by graduate students for course projects tar-

[1] http://www.nedapidentification.com/products/sensit/

[2] http://www.streetline.com

[3] http://www.proxel.com/en/

[4] http://www.xvisionsystems.com/Products/Parking-Sensors/

geting the domain of automated parking lot analysis which have been published online as part of the course, not a scholarly work[5,6]. These works include some interesting approaches to making general-purpose parking lot tracking systems which have no restriction on the users, such as Gaussian ground pixel modeling techniques. Even though they are not published, they help to demonstrate current approaches to the problem as well as emphasizing the lack of good solutions that have led to the recent growth in the area.

3. System Design

The developed system was designed and implemented using Python 2.7 and OpenCV 2.4.6. This selection of tools was made for several reasons. First, OpenCV is a logical choice for computer vision work in application development given its expansive feature-set and wide support across multiple platforms. This works well with the choice of Python for development since OpenCV's Python bindings make direct calls to its lower-level C++ implementations. The Python code itself can be easily translated to C++ for performance and memory efficiency improvements, but in its original form, it provides a simple, interpreted means for research and development, similar to Matlab.

For training, testing, and analysis, the PKLot data set is used [3]. This data set is the first of its kind, and being released only in mid-2015, again shows how recent much of the interest in this area is. It features many images from three different parking lots, captured overhead from building-mounted or lamp post cameras. Images have been split according to weather conditions, and there are sequences for each lot over the course of multiple days with many images per day. Another feature of this data set is the number of manually-segmented parking spaces, both empty and occupied. It contains over a half million segmented spaces, providing a huge pool of images for training and testing machine learning algorithms.

The program was developed from two aspects. First, the problem of identifying cars in a space is a machine learning one. This is a perfect example of how pattern recognition applies to this problem, and as the next section will explain in more detail, it is considered exclusively as such in this work. The second aspect is one of a knowledge-based or heuristic approach. Even though humans still use pattern recognition when finding an empty space, we do so with the assistance of lines. Parking lots are nearly all lined according to the same rules, and these lines act as guides to human users. Likewise, we can encode knowledge for the computer into these lines, such as determining the layout of spaces in the parking lot when there are no cars present.

[5] http://www.cs.cmu.edu/~epxing/Class/10701-06f/project-reports/wu_zhang.pdf

[6] http://cseweb.ucsd.edu/classes/wi07/cse190-a/reports/ntrue.pdf

4. System Implementation

4.1. Learning Approaches

4.1.1 Pixel-Based Model

To address the first problem of identifying whether or not a car has been detected in a space, we applied artificial neural networks and a sliding window algorithm. Our first model was the most simple one used in vision – to use the pixel data from a predefined window size directly. In this case, we used a window size of 36x48 pixels, yielding 5,184 input features when considering each of the RGB channels (36x48x3). This model was trained with 50,000 randomly-selected car/non-car images (relying on the labeling and segmenting provided in the PKLot dataset) and tested on another set of over 200,000 random images. The confusion matrix is shown in Figure 1 for the car (C) /non-car (NC) identification. Table 1 shows some of the relevant measures of this classifier.

This model performed surprisingly well at identifying non-car images, although it lacked accuracy regarding finding cars themselves. The high true negative rate can be attributed to the fact that much of the ground color in a parking lot is the same, due to equal fading of the pavement and the fact that these cameras capture mostly pavement with cars in spaces. As such, it seems reasonable that a pixel-based classifier could identify a non-car region, but the overall accuracy was still too low to use this model for recognizing cars, leading to the next approach.

4.1.2 Feature Descriptor Model

The second machine learning model also used neural networks on 36x48 windows; however, instead of using pixel values directly, feature descriptors were used. Specifically,

	C	NC
C	64.4%	35.6%
NC	19.3%	80.7%

Figure 1. The confusion matrix for the first model when distinguishing between cars (C) and non-car (NC) images.

Metric	Score
Precision	75.1%
Recall	64.4%
Accuracy	72.3%
F-Score	73.7%

Table 1. The first model's results in regards to important classification metrics.

Figure 2. An example car image with the SURF (left) and SIFT (right) keypoints overlaid; certainly some of the important regions are being found.

the top ten SURF and SIFT features were combined into a 1x20 input array to be used by the network for classification between car and non-car. SURF stands for Speeded-Up Robust Features and uses Wavelet responses along the horizontal and vertical directions to try to capture key regions of the image, referred to as ”keypoints” [2]. SIFT, short for Scale-Invariant Feature Transform, also attempts to extract image keypoints, an element common to many feature descriptors, but it uses layers of Gaussian-blurred versions of the image and extrema detection to try to find important regions [8]. Feature descriptors in general are intended to be compressed representations of images that contain the most significant portions of the image, somewhat like fancy edge detectors.

This approach intuitively makes more sense than just using the pixel values since cars inherently have features that make them identifiable to humans. For instance, every car has a windshield, doors, and tires. By attempting to capture this information in keypoints, we can assume car identification accuracy will increase. Figure 2 shows the SURF and SIFT keypoints on a single car image. While not perfect, we can clearly see that some important features like the windshield and edges along the car trim are being detected.

Again this model was trained on 50,000 random images and then tested on over 200,000 others. Figure 3 shows the confusion matrix of the results. Table 2 also shows the results of important classification measures.

Note the high recall for this model. It is very good at identifying if a car is in the image, which is promising for our purposes. Unfortunately, it is over-zealous in that it often classifies non-car regions as cars; we see this in its poor precision, which is lower than the precision of the first model. This is not particularly surprising because there are a lot of non-car elements of an image that may look like cars. For instance, foliage or any high entropy region may take on characteristics that look like tires or a windshield. We will see specific examples of this in the next section with a discussion of results. In the meantime, it is safe to say that while this model improves on the previous pixel-based one, it still suffers from limitations that contribute

	C	NC
C	93.6%	6.4%
NC	43.1%	56.9%

Figure 3. The confusion matrix for car (C) and non-car (NC) using the second, feature-descriptor-based model.

Metric	Score
Precision	66.3%
Recall	93.6%
Accuracy	74.3%
F-Score	77.6%

Table 2. The second model’s results in regards to important classification metrics.

to its overall low accuracy, which is about the same as the original network.

In light of the advantages and disadvantages tied to both models, we selected to use a combination of them. That is, since the pixel-based approach was good at determining non-cars, but the feature descriptors helped find cars, we consider both network classifications on a given window in the image. This helps reduce the misclassifications found in high entropy areas, but it still has some problems, as will be shown in the results.

4.2. Heuristic Approach

In addition to the computer vision and learning aspect of identifying cars, this work also considers the knowledge-based aspect of the problem. This is primarily motivated by the fact that parking lots are lined, providing guidance for humans but also carrying important information for the computer. In this case, we sought to use the lines of an image of empty or near-empty parking lots to automatically determine the layout of spaces in the lot. Having the lot layout is vital to being able to report open spots to users, just as relevant as being able to actually determine if a car is in a given spot. Other works seems to place restrictions on either the placement of cameras relative to the lot or in forcing the user to provide a model of the lot beforehand. Here, we consider an automated approach which tries to evaluate the lot layout using only knowledge from the lines.

Since there is no learning aspect in this part of the system yet, the algorithm can be very simply outlined as a series of image processing steps. All the steps listed below are performed on a grayscale version of the input image.

1. Ground Detection - The first step was detecting the ground pixels in the image to extract non-relevant portions of the image prior to edge detection. Some have

used Gaussians to model the pixel distribution to determine the most likely ground/pavement color. Here, we consider an even simpler approach

- We can assume that a camera facing the parking lot almost exclusively will contain mostly ground pixels for an empty or near-empty lot.
- Given that assumption, we take a histogram of the image and select a region of pixels surrounding the maximum value in the histogram according to some threshold.
- Pixels in that grayscale range are assumed to be ground pixels.

2. Edge Detection - Following the detection of ground data, we remove all extraneous pixels and perform edge detection to prepare for space detection.

    - Canny edge detection is applied first as a well-established means of extracting the strongest edges from an image. At this point, we initially tried to apply Hough Line detection directly to find spaces, but this resulted in only finding the parking rows, which had long edges associated with them. This led to the following sequence of steps for finding individual spaces.
    - A morphological closing was applied to the edges in order to clean them up for space detection.

3. Space Detection - As the final step, we sought to extract location information for the actual spaces in the parking lot.

    - Since the parking spaces themselves were too small to be detected with Hough Lines transformation on the whole image, a sliding window algorithm was employed over the image of edges
    - Hough Line detection was then run within each window. When two parallel lines are detected, the window is likely to contain a parking space. Unfortunately, Hough Line detection returns a lot of lines in noisy regions of the image, so this approach does include some detection of incorrect spaces. It does show a lot of promise though, as we will see in the next section.

5. Results

In this section, we will see how the previously-described methods perform on some sample images in the PKLot data set.

Figure 4 shows a parking lot on a sunny day, including a shadow from a nearby building. This shadow is one form of regular complications that can arise for a system designed for parking lot analysis. Figure 5 illustrates the result of the pixel-based model on Figure 4. Note that it performs fairly well at detecting cars versus non-cars in the sunlight, but the shadows do lead to multiple false positives. Conversely, Figure 6 shows the feature descriptor classifier output for the same image. In this case, cars are detected fairly well throughout, regardless of sunlight or shadow, but they are

Figure 4. An image of a parking lot. Notice how the two-tone effect introduced by the building shadow may further complicate the result.

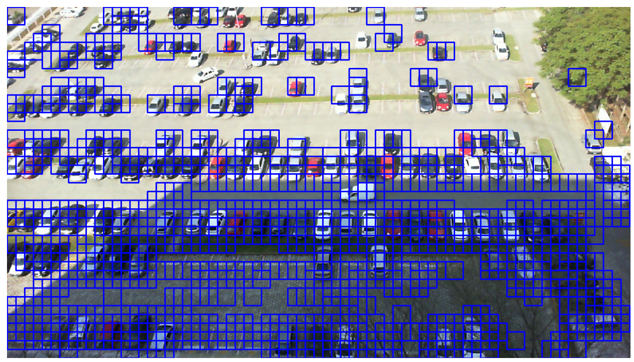

Figure 5. The pixel-based classifier applied to the parking lot.

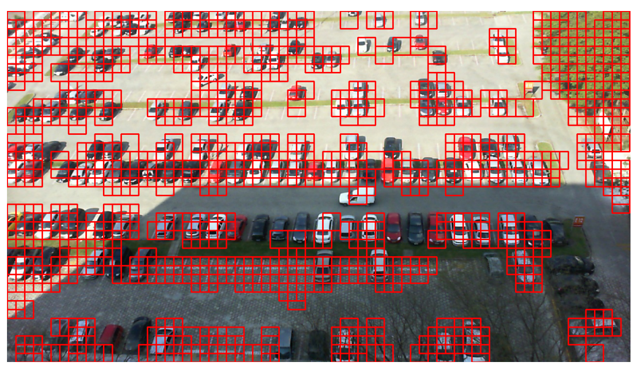

Figure 6. The feature descriptor classifier applied to the parking lot.

Figure 7. Applying both classifiers, combining the output according to a threshold and cleaning up overlapping windows.

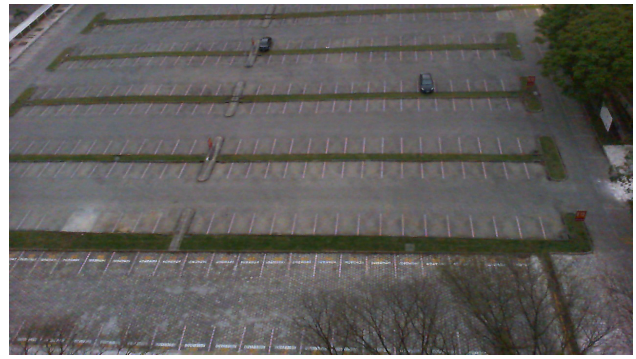

Figure 8. An image of the same parking lot when virtually empty for space detection.

heavily skewed in high entropy areas. The foliage in the upper right corner induces many false positives. This is understandable, since we can see in Figure 4 how regions of the tree leaves look like windshields and other car features. However, note that the pixel-based classifier performed very well over the areas of heavy foliage. By combining the two classifier outputs and cleaning up the resulting windows, we get the image shown in Figure 7. This image performs better than the two individual classifiers in shady or high entropy regions. While not perfect, it is a reasonable model for car detection just by building on the strengths of two simpler models.

To consider an example of the knowledge-based algorithm, we look at the same parking lot in a nearly empty state, as in Figure 8. The results of ground detection are shown in Figure 9, which shows that our assumption of the ground pixel being the most common is fairly reasonable for lot-directed cameras. The edge detection, seen in Figure 10, then performs well given the simplified image created with ground detection. If we attempt to apply Hough Line detection immediately to Figure 10, we see yield the result in Figure 11. While this seems to help find the parking lot's rows, it is ineffective at finding individual spaces. For that, we return to the edge-detected image and perform a morphological closing in preparation for space detection; see Figure 12. The final result of the sliding window and Hough Line detection can then be seen in Figure 13.

To put all of this capability together, we consider another image of the same parking lot when half-full, seen in Figure 14. Figure 15 shows the overlap of the car detection and space detection outputs. While imperfect, we see that largely, the regions with cars are blocked off with green boxes, while there are still white boxes unblocked at the portions with spaces but no cars. The bottom portion of the image is unfortunately inaccurate due to noise, which does affect the car detection result but is far more troubling for the space detection result.

Figure 9. The results of ground detection.

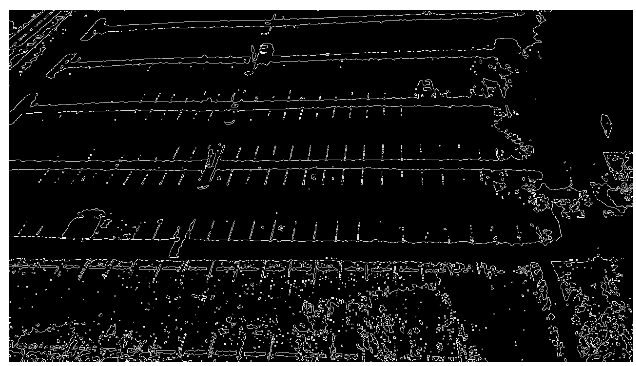

Figure 10. The results of edge detection.

6. Future Work

While the results are promising so far, there is still much work to be done to improve the system. Specifically, the outputs of both parts of the algorithm need to be combined more directly, allowing the system to specify exactly how many spaces there are in the image and how many are available. This would lead into future development of a user interface through which drivers could access parking availability information for the parking lot and owners or plan-

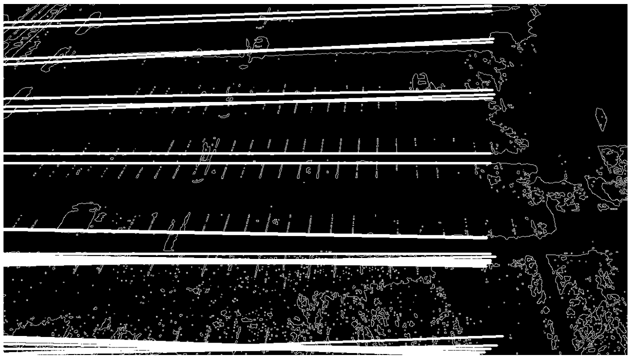

Figure 11. An attempt to extract parking lot layout using Hough Line detection directly on the edge-detected image. Notice that only long lines, like parking rows, can be detected.

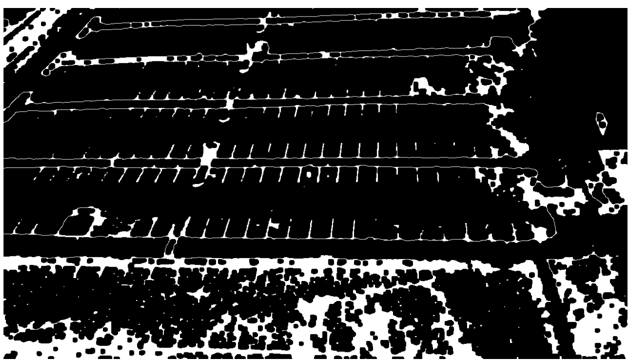

Figure 12. The result of closing the edge-detected lot.

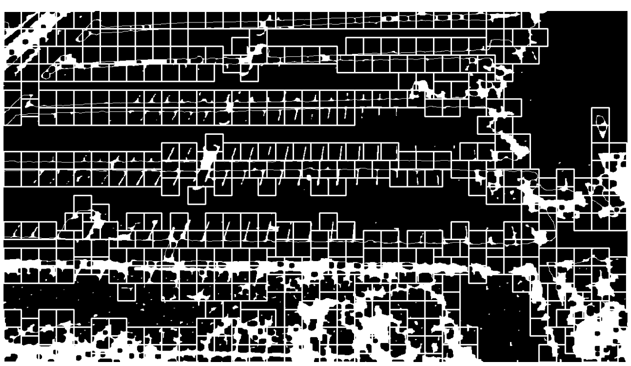

Figure 13. The final space detection result, combining sliding windows with Hough Line detection.

ners could get analytical data. Fortunately, the frontend and backend of such a system can be largely uncoupled so long as the interface between the two remains the same. This work has focused almost exclusively on the backend recognition, and there is much room for improvement in that regard as well. The car detection could be improved with a better classifier, perhaps using some new car-specific features descriptors. Furthermore, the space detection technique could be improved, and it is not unreasonable to expect that the user could provide some help to the algorithm at this point. Since the layout would only need to be constructed once at the time of installation, the user could identify incorrect spaces at that time. This does add some supervision, but it is simpler than requiring the user to generate a complete 3D model beforehand while still ensuring accurate layout for the system's future use.

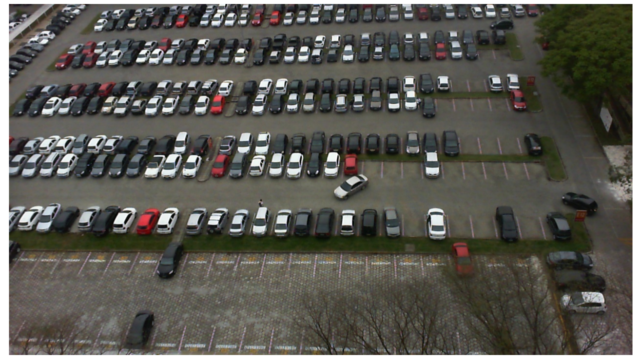

Figure 14. Another image of the same lot half-full.

Figure 15. The output of both car and space detection.

7. Conclusion

Automated parking lot analysis systems offer many advantages to drivers, business owners, engineers, and others. Drivers would experience most of these advantages in the form of up-to-date information about space availability to help them find the best parking space near their destination. The metrics these systems produce can be used to better maintain existing lots, keep drivers informed of closures, track business performance, and more through the options provided by big data analytics. Unfortunately, current commercial solutions are not very scalable, being both expense and time-consuming to install and maintain. Vision-based approaches offer a much better way, being fast, easy to deploy on existing systems, and free from nearly any cost. While there isn't a vision-based competitor yet that can compete in terms of accuracy with the sensor-based commercial solutions, we have seen in this work how simple models built on applications of pattern recognition and domain-specific knowledge can be used to build a fairly powerful parking lot analyzer. Through advancing work on this system and the ongoing improvements in computer vision in other domains, it is very likely that we will see vision-based parking lot analysis software being widely deployed by governments, schools, and businesses in the near future.

References

[1] K. Aravind, T. Chakravarty, M. G. Chandra, and P. Balamuralidhar. On the architecture of vehicle tracking system using wireless sensor devices. In *Ultra Modern Telecommunications & Workshops, 2009. ICUMT'09. International Conference on*, pages 1–5. IEEE, 2009.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[3] P. R. de Almeida, L. S. Oliveira, A. S. Britto, E. J. Silva, and A. L. Koerich. Pklot–a robust dataset for parking lot classification. *Expert Systems with Applications*, 42(11):4937–4949, 2015.

[4] T. Fabian. An algorithm for parking lot occupation detection. In *Computer Information Systems and Industrial Management Applications, 2008. CISIM'08. 7th*, pages 165–170. IEEE, 2008.

[5] B. O. Hall. Computerized parking facility management system, Jan. 22 2002. US Patent 6,340,935.

[6] A. Lim, Q. Yang, K. Casey, and R.-k. Neelisetti. Real-time target tracking with cpa algorithm in wireless sensor networks. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2008. SECON'08. 5th Annual IEEE Communications Society Conference on*, pages 305–313. IEEE, 2008.

[7] S.-F. Lin, Y.-Y. Chen, and S.-C. Liu. A vision-based parking lot management system. In *Systems, Man and Cybernetics, 2006. SMC'06. IEEE International Conference on*, volume 4, pages 2897–2902. IEEE, 2006.

[8] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[9] J. K. Suhr and H. G. Jung. Sensor fusion-based vacant parking slot detection and tracking. *Intelligent Transportation Systems, IEEE Transactions on*, 15(1):21–36, 2014.

[10] V. W. Tang, Y. Zheng, and J. Cao. An intelligent car park management system based on wireless sensor networks. In *Pervasive Computing and Applications, 2006 1st International Symposium on*, pages 65–70. IEEE, 2006.

[11] X. Wang and A. R. Hanson. Parking lot analysis and visualization from aerial images. In *Applications of Computer Vision, 1998. WACV'98. Proceedings., Fourth IEEE Workshop on*, pages 36–41. IEEE, 1998.

[12] M. Winter and J. Osterweil. Apparatus and method for sensing the occupancy status of parking spaces in a parking lot, Oct. 3 2006. US Patent 7,116,246.

[13] S.-e. Yoo, P. K. Chong, T. Kim, J. Kang, D. Kim, C. Shin, K. Sung, and B. Jang. Pgs: Parking guidance system based on wireless sensor network. In *Wireless Pervasive Computing, 2008. ISWPC 2008. 3rd International Symposium on*, pages 218–222. IEEE, 2008.

[14] R. Yusnita, F. Norbaya, and N. Basharuddin. Intelligent parking space detection system based on image processing. *International Journal of Innovation, Management and Technology*, 3(3):232–235, 2012.