MVA2015 IAPR International Conference on Machine Vision Applications, May 18-22, 2015, Tokyo, JAPAN

# A Deep Reinforcement Learning Approach to Character Segmentation of License Plate Images

Farnaz Abtahi
The Graduate Center, CUNY,
New York, NY 10016, USA
fabtahi@gradcenter.cuny.edu

Zhigang Zhu
The City College and Graduate Center,
CUNY, New York, NY 10031, USA
zhu@cs.ccny.cuny.edu

Aaron M. Burry
Xerox PARC
Webster, NY 14580, USA
Aaron.Burry@xerox.com

## Abstract

*Automated license plate recognition (ALPR) has been applied to identify vehicles by their license plates and is critical in several important transportation applications. In order to achieve the recognition accuracy levels typically required in the market, it is necessary to obtain properly segmented characters. A standard method, projection-based segmentation, is challenged by substantial variation across the plate in the regions surrounding the characters. In this paper a reinforcement learning (RL) method is adapted to create a segmentation agent that can find appropriate segmentation paths that avoid characters, traversing from the top to the bottom of a cropped license plate image. Then a hybrid approach is proposed, leveraging the speed and simplicity of the projection-based segmentation technique along with the power of the RL method. The results of our experiments show significant improvement over the histogram projection currently used for character segmentation.*

## 1. Introduction

Automated license plate recognition (ALPR) using video/ image processing techniques has been applied to identify vehicles by their license plates and is critical in a number of important transportation applications. ALPR is considered a challenging problem in the field of machine vision and automation with various applications including law enforcement, parking lot ticketing systems, automated hand-free toll collection and automated vehicle access in secure establishments [1]. A key element of an ALPR system is character segmentation – extracting images of each individual character in preparation for an optical character recognition (OCR) step.

In order to achieve the required OCR accuracy levels, it is necessary to obtain properly segmented characters. A standard segmentation technique in the field of license plate recognition is a projection segmentation approach based on the vertical histogram of a license plate image [2]. Although this method is the simplest and most common segmentation approach in ALPR systems [3] and is also fast and mostly effective, it struggles with complex backgrounds and partial obstructions. It also requires the license plate image to be rectified. In order to improve the robustness of the overall ALPR system and extend its operating latitude to include other applications, improvements to character level segmentation are required.

In this paper we describe a machine learning approach to character level segmentation of license plate images. In particular, the reinforcement learning (RL) method is adapted to create a segmentation agent that can find appropriate segmentation paths that avoid characters. In a preferred embodiment, the results of an aggressive projection segmentation step are used as seed points for the RL method. The RL agent then determines which of the possible cut paths are valid and which are not. The results of our experiments show that our proposed approach is very promising and can achieve higher performance compared to existing projection-based segmentation methods.

## 2. Background and Related Work

The goal of character segmentation is to decompose an image of a sequence of symbols into sub-images of individual symbols (i.e. characters). In a classical OCR approach, segmentation is the initial step in a three-step procedure. Starting from a point in the image [4]: (1) find the next character, (2) extract features of the character image, and (3) find the symbol that best matches those features and output its identity. This sequence is repeated until all characters in the image are recognized, or no more characters are left. There are three major strategies for character segmentation; many hybrid approaches are combinations of these strategies [4]:

(1) The classical approach, in which the criterion for good segmentation is the agreement of general properties of the segments obtained with those expected for valid characters.
(2) Recognition-based segmentation, in which the system searches the image for components that match classes in its alphabet.
(3) Holistic methods, in which the system seeks to recognize words as a whole, thus, avoiding the need of segmentations.

Not many works have used machine learning techniques to address the character segmentation problem. Most published character segmentation methods rely upon heuristically derived rules and the use of classification techniques in segmentation is largely an unexplored area [5,6]. The approaches that were often selected as techniques to test against human crafted software modules that employ heuristic rules are usually backpropagation neural networks and Bayesian classifiers. In [6] Bayesian classifiers were constructed according to Duda and Hart [7]. The training data is used to estimate the a-posteriori probability of the occurrence of each class, given each input vector. During evaluation of the test sets, the a-posteriori probabilities are used to select a classification that minimizes the likelihood of error. Back propagation networks were built according to Rumelhart and McLelland [8]. Each network has an input node for each element of the input vector and an output node for each of the mutually exclusive classes.

## 3. Projection-based Method: a Baseline

The XLPR, an ALPR system developed by Xerox, is currently in real use and achieves state of the art performance on a number of real-world data sets obtained from several tollbooths across the US. The current segmentation subsystem within XLPR uses a vertical histogram projection to produce character boundaries (cuts), and uses local statistical information, such as median character spacing, to split large cuts (caused by combining characters) and to insert missing characters [9]. These operations require minimal computational resources and consequently are applied to each input image to achieve good character segmentation accuracy in real-time. No a-priori information is utilized in this method, thereby enabling robust performance over a variety of state logos, fonts and character spacing that exists in license plates in the US.

A key challenge for the basic projection-based segmentation technique is substantial variation across the plate in the regions surrounding the characters. For instance, in Figure 1, the partial obstruction near the center of the plate clearly presents a different local challenge as compared to other inter-character regions on the plate. It is exactly this type of local variability that the current projection-based segmentation approach struggles with.

Figure 1. Example of a difficult segmentation case.

Complex background pictorials are also particularly challenging as they too present local variations not easily overcome with a fixed segmentation threshold. Take for instance the image in Figure 2(a), showing a difficult license plate background. Note that the projection-based segmentation method failed to identify the segmentation boundaries between the Florida logo and the characters on either side. Adjusting the aggressiveness of the threshold for the projection-based segmentation method can help to prevent missed cuts. However it can also lead to over-segmentation of images, as seen in Figure 2(b). It is often extremely difficult at design time to find the right threshold setting such that we can reduce the under-segmentation cases (missed cuts) without inducing unwanted over-segmentation cases (split characters) across a large number of license plate images.

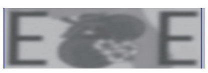
(a)  (b) 

Figure 2. Projection-based segmentation. (a) Miss and (b) over-segmentation example for a Florida plate.

## 4. Text Segmentation Using Deep Reinforcement Learning

In this paper, we use the Deep Reinforcement Learning (DeepRL) algorithm proposed in [10] which combines the classical reinforcement learning with deep belief networks (DBNs [11]). The diagram in Figure 4 summarizes the system architecture of the DeepRL method. The basic idea underlying combining DBNs with RL is to take advantage of the unsupervised pre-training phase in DBNs, and then use the DBN as the starting point for a neural network function approximator for representing the Q-function of the reinforcement learning. The algorithm is explained in more details below.

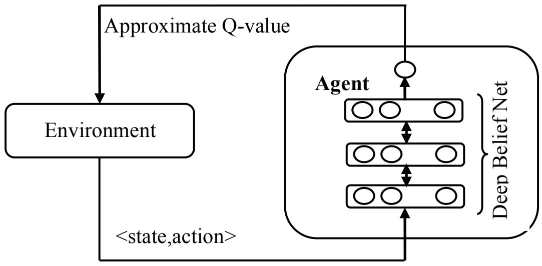



Figure 4. DeepRL system architecture.

### 4.1. DeepRL algorithm

The DeepRL algorithm consists of two main steps. First, the training set and the DBN are initialized. Depending on the setting that we would like to use in a particular experiment, we can use different initializations. The initial transition samples are a set of <state; action; target> tuples. If we decide to start with unsupervised pre-training, the DBN is pre-trained on the set of transition samples, without taking the target values (i.e., return estimates or the Q-value of each state) into account.

The second step of the algorithm is the reinforcement learning loop. From this point on, the algorithm works similar to the Neural Fitted Q-Iteration (NFQ) approach [12] and the DBN weights are used as the initial configuration of a regular neural network value function approximator. This part of the algorithm begins with using the current Q-function for a greedy policy, which is run in the environment to gather an additional set of experiences, which are then attached to the initial training set.

Afterwards, the combined set is used to update the network and get a new estimate of the Q-function. This is done by using the current Q-function to recalculate the target values for every experience tuple in the updated training set, and then stochastic gradient descent is applied to update the value function outputs [12]. These steps are repeated N times, or until the Q-function converges and the updated targets are successfully learned.

### 4.2. Text segmentation using DeepRL

Our proposed method uses DeepRL approach to identify the proper segmentation paths (cuts) from top to bottom in a cropped license plate image. The character segmentation problem is formulated into an extended version of the “Puddle World”, which is a benchmark problem in the RL literature: (1) Characters count as puddles, and are to be avoided; (2) Starting from a certain point on the top border of the text area of a license plate, the goal is to reach the bottom border as fast as possible, without entering the body of any of the characters; (3) Moving from a dark area (potentially a character) to a lighter area is always preferred.

Starting from a point on the top border of the license plate image, 4 moves are possible: up, down, right, and left. A neighborhood of size N×N is used as a “field of view”. The mean gray value of the neighborhood as seen looking in the four possible directions is compared to that of the neighborhood centered on the current pixel location. Since characters are typically seen as dark on lighter backgrounds, we define rewards such that moving from light to dark is considered a bad move, while moving from dark to light or staying in a light region are good. Examples of good and bad moves are shown in Figure 3. In addition, since we are attempting to draw segmentation cuts from top to bottom, moving down is preferred (rewarded) over moving up (punished).

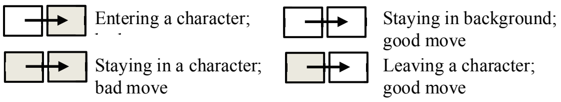



Figure 3. Possible moves from a pixel/window to its neighbors.

During the training phase, the agent attempts to traverse from the top of the license plate image to the bottom. Rewards are provided according to the number of "good" and "bad" moves that were made. At the end of training, the agent has learned to associate its current state and its "sensory input" (what it "sees" in each of the four possible move directions) with a preferred action. At runtime, the RL agent uses the associations learned in the training phase to traverse from the top to bottom of a given license plate image in real-time. So, despite the training phase that is performed offline on a batch of training samples, the test time is in milliseconds and can be performed in realtime.

### 4.3. A hybrid method

The RL based technique outlined above requires that the agent be initialized to a given starting location. One approach would be to simply have the agent attempt to draw segmentation paths starting at every possible pixel location in the top row of the image. However, this would not be very efficient. In a preferred embodiment, the projection-based segmentation method is used to provide seed locations for the RL method. In particular, the projection-based segmentation is performed with a larger than normal threshold. Recall that in the projection-based method, any column whose normalized vertical sum is less than the threshold is a candidate cut. Thus, the higher than normal threshold means that we will tend to over-segment the image. The RL technique then serves to screen off unwanted segmentation cuts that are deemed invalid. This hybrid approach leverages the speed and simplicity of the projection-based segmentation technique along with the power of the RL method.

## 5. Experimental Results

We have performed three sets of experiments. In all the experiments, we used image samples of simulated Florida license plates that have been used in algorithms applied to real-world OCR in XLPR. Even though the images are synthetic, they are highly representative of the real license plate images. They cover poor image resolution, blurriness, poor lighting, low contrast and occlusion. In particular, the logo is well represented in the simulated images. Note that many existing transportation solutions leverage near-infrared illumination for both stealth and minimizing driver distraction when acquiring images of license plates. Since these capture systems result in grayscale output images, the synthetic images used in this study likewise do not include color information and even if they did, this information would not have been reliable enough to be used for segmentation purposes since the Florida license plate in fact uses green characters overlaying a green logo.

Ground truth data was provided for the samples, i.e. a binary mask image in which only the characters are included and the logo or background image is omitted. These binary masks are only used to automatically count the number of correct vs. incorrect segmentation lines/ paths and are not utilized for training the system. We perform our test on a set of 2-character images cut from license plates with logos (such as in Figure 2), which is considered a very hard case for segmentation and has the most problems in the projection-based segmentation.

The problem with looking one pixel ahead is that the search performed on the image is too local. If there is a logo or noise in the background, it is either confused with the characters and is tried to be avoided (not really a failure), or we get stuck in the logo, because all moves are equally bad. Therefore in all our experiments we increase the size of the window to 3x3 to get a broader view of the surroundings. We reward a move if the average intensity over the entire block increases (brighter).

In the first experiment, we tested the basic DeepRL method on a dataset of 52 images similar to the "E-logo-E" stress case for all 26 letters (two images per letter, one used for training and the other one for testing), for 150 random starting points on the top border. Three types of failures occur: (1) Letters with stokes (E, L, …): the stroke does not significantly affect the average of a block and might be cut through. (2) Getting stuck in the logo: because the window size is not large enough to contain a way out of the logo. (3) Getting stuck inside a character (H, U, …): happens because the algorithm always prefers to move down. Table 1 shows the segmentation results for this dataset (percentage of segmentation paths falling in different categories). As we can see, even though the dataset is relatively small, the results are still acceptable.

Table 1. Performance of basic DeepRL method

| Cut through characters | Stuck in a character | Stuck in the logo | Successful segmentation |
|---|---|---|---|
| 2.5% | 5% | 8% | 84.5% |

In the second experiment, we test the hybrid method on the same dataset as the first experiment. For each image, we use a value for threshold that results in the best set of seed points for segmentation. Usually a higher threshold is better as it generates extra cut points, some of which are eliminated by the DeepRL segmentation phase, whereas a lower threshold will fail to find a number of necessary cut points. Some examples of the type of results that we get using the hybrid method compared with the projection-based approach can be seen in Figure 4. Here, note that the agent gets stuck within the "well" of the U's. Rather than forcing an over-segmentation, the agent is allowed to stay within this region as an indication that a suitable segmentation path cannot be found from the given starting location.

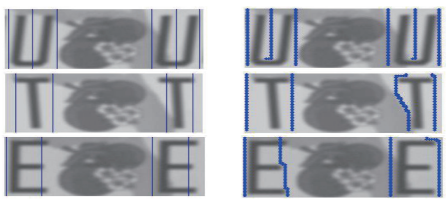


Figure 4. Initial projection-based results (left) vs. the results from the proposed hybrid method (right).

Note that we can choose different thresholds for the projection-based method and that results in different sets of initial cut points. The number of successful and failed cuts performed by the projection-based vs. hybrid method using a low, medium and high threshold are listed in Table 2. As explained in section 4.3, for the hybrid approach, we used a high threshold to get an over-segmented image to start with.

Table 2. Number of different types of cuts performed by the projection-based vs. the hybrid technique.

| | Projection Threshold | | | |
|---|---|---|---|---|
| | Low | Medium | High | Hybrid RL Method |
| Missed cuts (false negative) | 44 | 30 | 13 | 4 |
| Cuts into characters (false positive) | 0 | 22 | 39 | 2 |
| Correct Cuts (true positive) | 60 | 74 | 91 | 100 |

Table 3 shows the precision (true positive divided by the sum of true positive and false positive) and recall (true positive divided by the sum of true positive and false negative) on the dataset for the projection-based method using the same low, medium and high threshold and also the hybrid method. Since there is no training phase for the projection-based technique, only the test set has been used to be consistent with the way we test the hybrid method. These results show the proposed method to be significantly more robust than the existing projection-based segmentation technique.

Table 3. Performance of the projection-based technique vs. hybrid method.

| | Projection Threshold | | | |
|---|---|---|---|---|
| | Low | Medium | High | Hybrid RL Method |
| Precision | 100.0% | 77.1% | 70.0% | 98.0% |
| Recall | 57.7% | 71.2% | 87.5% | 96.2% |

In order to see the effect of training on a larger dataset, in the third experiment we apply the same hybrid method on a dataset with 50 Florida license plate images per letter (a total of 1300 images). In Hybrid method, 70% of the data is used for training and the remaining 30% is used for testing. The same test set is used for testing the projection-based technique. The results are summarized in Table 4 and 5. As we can see, there is almost no drop in either precision or recall.

Table 4. Number of cuts performed by the projection-based vs. the hybrid technique for the large dataset.

| | Projection Threshold | | | |
|---|---|---|---|---|
| | Low | Medium | High | Hybrid RL Method |
| Missed cuts (false negative) | 794 | 553 | 235 | 88 |
| Cuts into characters (false positive) | 26 | 447 | 1062 | 59 |
| Correct Cuts (true positive) | 766 | 1007 | 1325 | 1472 |

Table 5. Performance of the projection-based technique vs. hybrid method for the large dataset.

| | Projection Threshold | | | |
|---|---|---|---|---|
| | Low | Medium | High | Hybrid RL Method |
| Precision | 96.7% | 69.3% | 55.5% | 96.1% |
| Recall | 49.1% | 64.6% | 84.9% | 94.4% |

## 6. Conclusions and Future Work

In this paper, we applied DeepRL to improve the character segmentation unit of the ALPR systems. The results of our experiments show significant improvement over the histogram projection method currently used for character segmentation. The performance is expected to improve even more if we: (1) use more informative features than the average intensity; (2) try different window and shift sizes to find the best combination; and (3) learn the structure of the characters to be able to recognize them as a whole and not at a pixel level [13]. These will be our future work in further improving the accuracy of proposed approach. Meanwhile, we will also need to optimize the computational efficiency of the DeepRL approach by simultaneously starting the paths from all seed points on each test image, thus parallelizing the process of finding the segmentation paths.

**Acknowledgments:** This work has been supported by the Xerox Summer Internship Program, and the National Science Foundation (Award #EFRI-1137172). Major part of the work was performed when the first author was visiting Xerox's PARC research site in Webster, NY.

## References

[1] S. Roy, "Vehicle License Plate Extraction and Recognition", Technical Report, Dept. of Computer Science and Engineering, IIT, Kharagpur, India, 2007.

[2] A. M. Burry, C. Fillion, V. Kozitsky, "Robost Character Segmentation for License Plate Images", U.S. Patent Application No: 13,539,739, July 2, 2012.

[3] S. Du, M. Ibrahim, M. Shehata, W. Badawy, "Automatic License Plate Recognition (ALPR): A State-of-the-Art Review", IEEE Trans. on Circ. and Sys. for Video Tech., vol. 23, issue 2, pp. 311-325, Feb. 2013.

[4] R. G. Casey, E. Lecolinet, "A Survey of Methods and Strategies in Character Segmentation", IEEE Trans PAMI, 1996.

[5] T. Nirav, et al, "A Survey on Offline-Methods of Character Segmentation", Proc. Int. Conf. Advances in Computer Science and Electronics Engineering, India, 2012: pp45-51.

[6] R. W. Smith, J. F. McNamara and D. S. Bradburn, "Pattern Classification Techniques Applied to Character Segmentation", Proc. 5th USPS Advanced Technology Conference, Washington, D.C., 1992.

[7] R. O. Duda and P. E. Hart, "Pattern Classification and Scene Analysis", John Wiley and Sons, 1973.

[8] J. L. McClelland, D. E. Rumelhart and PDP Research Group, "Parallel Distributed Processing", MIT Press, 1986.

[9] T. D. Duan, et al, "Building an Automatic Vehicle License-Plate Recognition System", Intl. Conf. in Computer Science (RIVF'05), Can Tho, Vietnam, 2005.

[10] F. Abtahi and I. Fasel, "Deep Belief Nets as Function Approximators for Reinforcement Learning", Proc. AAAI Workshop on Lifelong Learning from Sensorimotor Experience, 2011.

[11] G. E. Hinton, S. Osindero and Y. Teh, "A Fast Learning Algorithm for Deep Belief Nets", Neural Computation, 18:1527–1554, 2006.

[12] M. Riedmiller, "Neural Fitted Q-Iteration − First Experiences with a Data Efficient Neural Reinforcement Learning Method", ECML 2005, 317–328, 2005.

[13] H. Lee, R. Grosse, R. Ranganath, A. Y. Ng, "Unsupervised Learning of Hierarchical Representations with Convolutional Deep Belief Networks", Communications of the ACM, 54(10), 2011: 95-103.