Segmentation- and Annotation-Free License Plate Recognition With Deep Localization and Failure Identification

Orhan Bulan, *Senior Member, IEEE*, Vladimir Kozitsky, *Member, IEEE*, Palghat Ramesh, *Member, IEEE*, and Matthew Shreve, *Member, IEEE*

***Abstract*— Automated license plate recognition (ALPR) is essential in several roadway imaging applications. For ALPR systems deployed in the United States, variation between jurisdictions on character width, spacing, and the existence of noise sources (e.g., heavy shadows, non-uniform illumination, various optical geometries, poor contrast, and so on) present in LP images makes it challenging for the recognition accuracy and scalability of ALPR systems. Font and plate-layout variation across jurisdictions further adds to the difficulty of proper character segmentation and increases the level of manual annotation required for training classifiers for each state, which can result in excessive operational overhead and cost. In this paper, we propose a new ALPR workflow that includes novel methods for segmentation- and annotation-free ALPR, as well as improved plate localization and automation for failure identification. Our proposed workflow begins with localizing the LP region in the captured image using a two-stage approach that first extracts a set of candidate regions using a weak sparse network of winnows classifier and then filters them using a strong convolutional neural network (CNN) classifier in the second stage. Images that fail a primary confidence test for plate localization are further classified to identify localization failures, such as LP not present, LP too bright, LP too dark, or no vehicle found. In the localized plate region, we perform segmentation and optical character recognition (OCR) jointly by using a probabilistic inference method based on hidden Markov models (HMMs) where the most likely code sequence is determined by applying the Viterbi algorithm. In order to reduce manual annotation required for training classifiers for OCR, we propose the use of either artificially generated synthetic LP images or character samples acquired by trained ALPR systems already operating in other sites. The performance gap due to differences between training and target domain distributions is minimized using an unsupervised domain adaptation. We evaluated the performance of our proposed methods on LP images captured in several US jurisdictions under realistic conditions.**

***Index Terms*— Plate localization, deep learning, convolutional neural networks, segmentation, annotation, domain adaptation, image quality assessment, character recognition.**

Manuscript received May 9, 2016; revised September 30, 2016 and November 27, 2016; accepted December 3, 2016. This paper was presented at the IEEE Conference on Intelligent Transportation Systems 2015 [1] and at the IEEE Conference on Computer Vision and Pattern Recognition Workshop 2015 [2]. The Associate Editor for this paper was Z. Duric.

O. Bulan was with PARC, Webster, NY 14580 USA. He is now with General Motors, Warren, MI USA (e-mail: orhan.bulan@gm.com).

V. Kozitsky, P. Ramesh, and M. Shreve are with PARC, Webster, NY 14580 USA (e-mail: vladimir.kozitsky@parc.com; palghat.ramesh@parc.com; matthew.shreve@parc.com).

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/TITS.2016.2639020

I. Introduction

ALPR is an essential and commonly used technology in several roadway imaging applications [3]–[18]. ALPR systems typically consist of plate localization, character segmentation and recognition modules as shown in Fig. 1. For ALPR systems in USA, a module for identifying the state of the license plate is also needed as the same code can be issued by different jurisdictions. Though being a mature technology, the current challenges with ALPR systems are two-fold: a) scalability (i.e., deploying the same system to different sites) and, b) minimizing human intervention in the loop (i.e., reducing the number of images that require human review). Both of these challenges add to installation and operational costs which transportation solution providers desire to reduce.

One way to reduce human intervention in ALPR systems is to boost the performance of each ALPR module individually and hence, improve the overall system performance. While improving the recognition performance has an important role in reducing manual intervention, the images that do not contain a license plate, due to reasons such as image acquisition failures or partial/complete occlusion, still require human review for confirmation. By classifying and filtering out images where the license plate code does not exist or is not readable, the number of images that would require review can be significantly reduced.

Significant human effort is also required in scaling and/or re-applying an existing ALPR system to other jurisdictions and sites. Considering the wide variety of font samples, as well as the variations in plate design, layout, camera configurations, geometry, etc., manual annotation results excessive operational cost and overhead, and hence, poses an important challenge for the scalability of ALPR systems. This is especially in true in the character recognition module, which requires an off-line phase to train a classifier for each character in a one-vs-all fashion using a set of manually annotated character samples [1].

Another factor that often inhibits the scalability of ALPR systems is character segmentation, which can be a major source of ALPR errors [19]. A standard and commonly used segmentation approach in ALPR systems is projective segmentation, where a vertical projection of the image is used to determine the cut points around each character [3], [19]. Although this method is simple, fast, and effective in most

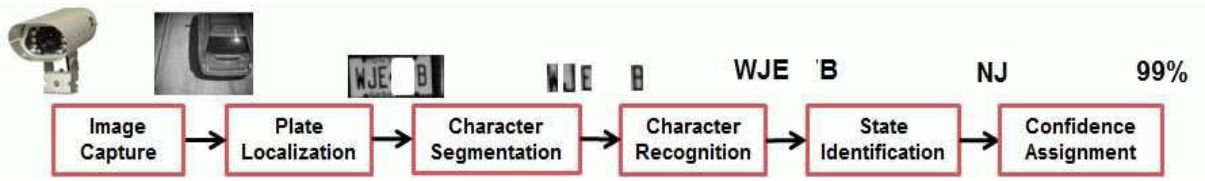


Fig. 1. A typical ALPR system modules.

license plate images, there are several cases that can prevent projective segmentation from being feasibly used in all US jurisdictions, especially in states that have complex plate backgrounds (e.g., Florida and Oregon), and plate logos that touch and overlap some characters in the image. In addition to plate logos, shadows crossing the license plate, borders and stickers are other factors that make standard techniques difficult to scale to different sites and jurisdictions.

In this paper, we propose a new work-flow and methods for segmentation and annotation free ALPR with improved plate localization and automatic failure identification. Our proposed work-flow first localizes the license plate region in the captured image using a two-stage approach where a set of candidate regions are first extracted using a weak SNoW classifier and then scrutinized by a strong CNN classifier in the second stage. Images that fail a primary confidence test for plate localization are further classified to identify reasons for failure, such as license plate (LP) not present, LP too bright, LP too dark or no vehicle found. In the localized plate region, we perform segmentation and OCR jointly by using a probabilistic inference method based on HMMs where the most likely code sequence is determined by applying the Viterbi algorithm. In order to reduce manual annotation required for training classifiers for OCR, we propose to use either artificially generated synthetic license plate images or character samples acquired by trained ALPR systems already operating in other sites. The performance gap due to differences between training and target domain distributions is minimized using an unsupervised domain adaptation. We evaluated the performance of our proposed methods on license plate images captured in several US jurisdictions under realistic conditions.

Our contribution in this paper is the following: First, we propose a new end-to-end system for segmentation and annotation free license plate recognition. Our system, specifically, addresses the scalability challenges in ALPR systems, which have not been considered in prior art, while still meeting the high accuracy and yield requirements, that ALPR systems require in tolling. A novel 2-stage plate localization and failure identification are presented using CNN features. In order to reduce the training time in deployment, the features extracted from synthetic images are transferred in a common representation domain using an unsupervised domain adaptation technique. We also included a language model in the HMM based plate decoding, which was not included in [8]. Finally, we conducted extensive experiments on LP images captured under realistic conditions and performed a comparative study of various features for character recognition and reported their performances on accuracy-yield curves.

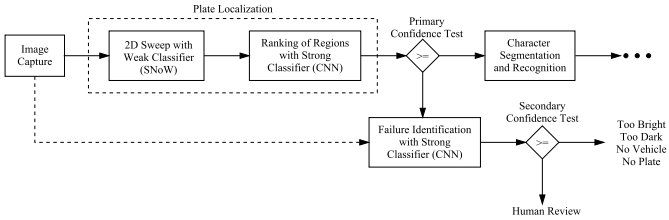


Fig. 2. Overview of plate localization and automation failure identification.

II. License Plate Localization and Automation Failure Identification

Accurate localization of the license plate in the captured image/video frame is the first step in an ALPR system. The majority of the existing techniques are based on image processing algorithms that leverage the color or geometric attributes of license plates [20]–[25]. Although fast and efficient, existing localization techniques typically are not machine learning based and do not include automation for failure identification, which can reduce the number of images requiring human review. Classifying the localization failures can also help in identifying the possible problems that can occur in an ALPR hardware system. If consecutive images are identified as "no vehicle," it could imply that there is a problem in the triggering system (e.g., induction loops) installed beneath the road. Similarly, identifying consecutive images as "too light" or "too dark" indicates a problem with the illuminator, for which the ALPR system can automatically send a notification to the operator.

We will now describe the two-stage license plate detection algorithm with automation failure identification (see Fig. 2 for a high level overview). In the first stage, we utilize a SNoW classifier trained with successive mean quantization transform (SMQT) features [26]. We extracted SMQT features at few scales given the camera geometry/configuration. Figure 3 shows an example of the regions identified by 2-D sweeping of the SNoW classifier trained with SMQT features across a license plate image.

After identifying the candidate license plate regions, the strong classifier is used to discriminate between readable and unreadable license plate images. The readable image set includes all regions that contain a license plate which can be recognized by a human and in turn capable of being successfully processed with an ALPR engine. The unreadable set includes all regions for which a human cannot recognize the license plate code and/or State. This set would include cases where the license plate is not present, is partially occluded, is too dark, too bright, or mangled, etc. Our goal

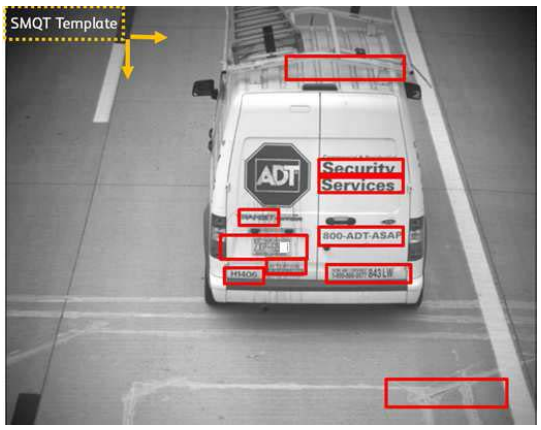


Fig. 3. Example of sweeping weak classifier and regions identified.

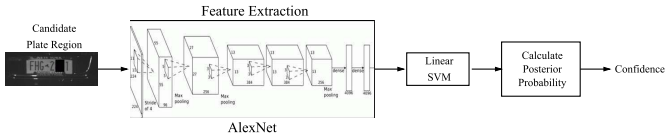


Fig. 4. Strong classifier (AlexNet) applied to each region of interest identified by the weak classifier.

is to automatically identify and exclude the un-readable (non-revenue) imagery from human review.

We choose AlexNet [27] as the feature extractor for discriminating between readable and unreadable LP images. AlexNet was one of the deep CNN architectures that showed state-of-the-art classification performance on the ImageNet dataset. The network contains five convolutional and three fully connected layers, where the last fully connected layer is fed to a softmax function to produce a distribution over the class labels. AlexNet has a total of 60 million parameters and 650,000 neurons. Figure 4 shows an illustration of plate/no-plate classification in the second stage. Each candidate region is resized to 224×224 before it is passed to AlexNet and the 4096 features are extracted from the layer before Softmax. We use these features as input to a linear SVM that is trained to differentiate between regions that either do or do not contain a plate. We use Platt's method to convert the SVM score to a posterior probability and use this probability as the confidence output of the second stage. In our experiments, we obtained slightly better results with a linear SVM classifier than the softmax classifier inherent in the neural network by tuning the hyper-parameters of the linear SVM. In our experiments, we selected quadratic programming in the optimization of finding the separating hyperplane. We also set the regularization parameter to 0.9 to carefully trade-off training error with the largest minimum margin hyperplane.

The output of the strong classifier is a set of candidate ROIs ranked by confidence. The highest confidence is compared against a threshold to determine if a plate exists. If the test is satisfied, the ROI is passed on for further processing as shown in Fig. 2. When the primary confidence test fails, it implies that there is no license plate in the image, possibly indicating a localization failure. In order to identify the failure reason,

Fig. 5. Example images for three classes where plate is not readable.

we train another strong classifier (AlexNet) with three classes: too bright, too dark, and no vehicle. A pre-trained AlexNet is fine-tuned for these three classes using the originally captured license plate images after resizing them to a standard template size of 224×224. When an image fails the primary test, the fine-tuned AlexNet calculates a probability for each of the three classes using softmax. Each class has a unique probability threshold that is tested to determine whether the original image should be classified as having a particular problem. The class with the highest margin above its threshold is selected as the final classification output. If no classes have confidence exceeding their threshold, then the image is sent to human review (refer to Fig. 5 for some example illustrations). The thresholds for each class are determined during the classifier validation phase which is offline and follows classifier training. During validation, we intend to pick a threshold such that the false positive rate (FPR) of the classifier is less then or equal to 0.5%. This low FPR (0.5%) is selected based on the acceptable limits of transportation agencies, as they desire to minimize the amount of lost revenue from incorrectly filtering out readable license plate images. Therefore, our goal is to maximize the true positive rate (TPR) at the FPR set point of 0.5%.

III. Training Classifiers for OCR

Training of an ALPR system for a new state or jurisdiction is a multi-month process that involves gathering vast amounts of in-situ training data (i.e. license plate images, corresponding plate characters and state of issue), followed by iterative tuning and optimization of the ALPR algorithms. Furthermore, since the current ALPR algorithms are heavily tuned to state-dependent plate characteristics such as font and layout, the training process must be repeated for each new state or the addition of a new plate template within a state covered by a current installation.

In this section, we describe a new method to minimize manual annotation required for training an OCR engine in an ALPR system. In the offline phase, our method utilizes either artificially generated synthetic license plate images or character samples acquired by OCR engines already trained in a system that is already operational. Training the OCR engine using character samples that are different from the images acquired from the actual camera capture site causes a mismatch between training and target data distributions, which causes deterioration in the OCR performance in the field. In order to improve the OCR performance and match training and target data distributions, we apply an unsupervised domain adaptation via subspace and dictionary learning [28]. In the domain adaptation, we used a set of labeled samples

Fig. 6. Synthetic license plate generation.

for each character from the training set and a set of unlabeled character samples acquired from the actual camera site. The unlabeled character samples can be extracted using the generic license plate localization and character segmentation modules that are typically independent of character font, license plate lay-out etc. The domain adaptation estimates the domain shift between training and target images and generates a shared feature representation across training and target samples. One-vs-all classifiers for the OCR engine are trained using the shared feature representation space.

A. Artificial License Plate Image Generation

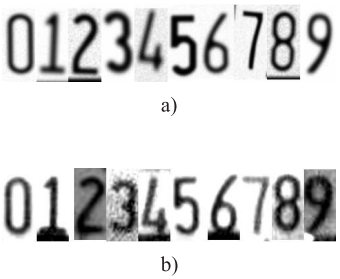
Generating synthetic images and using them to train classifiers have been studied in several object recognition tasks such as traffic sign recognition [29], handwritten word image retrieval [30], [31], and license plate recognition [32], [33]. Fig. 6 illustrates the overview of the license plate simulation work flow proposed in [33].

An end-to-end license plate simulation framework comprises two basic steps as shown in the figure. The first step is graphically rendering license plate characters onto blank license plate images. This rendering process considers effects such as font, spacing, layout, embossing, shadow, etc. to realistically mimic actual license plate images. The second step is mimicking imaging distortions typically encountered in the capture process. These distortions include color-to-infrared conversion, image blur, brightness and contrast variations, plate noise, and geometric distortions. The distortion parameters can be derived by solving an optimization problem [32] or by a direct analysis and measurement of a small number of real license plate images [33]. After generating synthetic license plate images, we apply a generic license plate localization algorithm on the artificially generated images to collect segmented characters.

B. Unsupervised Domain Adaptation For License Plate Recognition

Figure 7 shows example digit character images obtained from artificially generated license plates and from actual camera capture site. Even though synthetic character images look similar to actual images in general, the distortion can be quite broad in the real images which is hard to model and estimate in the synthetic image generation process. The slight difference between synthetic and real character images in terms of camera distortion causes a mismatch between distributions of synthetic and real images. When OCR classifiers are trained using synthetic images, a notable performance loss is observed in the recognition accuracy due to this mismatch.

In this section, our goal is to reduce this performance loss by applying an unsupervised domain adaptation via subspace



a)

b)

Fig. 7. A set of example digital characters obtained from a) artificially generated license plates, and b) actual camera capture site.

a) b) c)

Fig. 8. Examples of plates for which character segmentation is challenging.

and dictionary learning [28] to match distributions of synthetic (i.e., training data) and real images (i.e., target data). The dictionary based domain adaptation framework assumes a smooth transition path between subspaces of training and target data. A shared feature representation across domains is created by sampling several intermediate domains from the smooth transition path, where each sampled domain is represented by a dictionary D_n, $n = 1, 2, \ldots, N$ in the subspace. The algorithm starts with learning an over-complete dictionary D_0 to represent character images in the training data as a linear combination of atoms in the dictionary. D_0 represents the subspace for training images (e.g., synthetic images) and is used to incrementally learn dictionaries for the intermediate domains, where D_N represents the subspace for the target data. Since the path between training and target domains is assumed to be smooth, the dictionary D_{n+1} is calculated as $D_{n+1} = D_n + \Delta D_n$. Assuming $\mathbf{I_t} \in R^{k*M}$ is data instances (i.e., feature representation of M character images) from the target domain, ΔD_n is estimated by decomposing the target data using the dictionary D_n and calculating the reconstruction error E_n as follows:

$$\Gamma_n = \arg\min_{\Gamma} ||\mathbf{I_t} - D_n\Gamma||_F^2 \quad (1)$$

$$s.t.\ ||\alpha_i||_0 < S \quad (2)$$

$$E_n = ||\mathbf{I_t} - D_n\Gamma_n||_F^2 \quad (3)$$

where $\Gamma_n = [\alpha_1, \alpha_2, \ldots, \alpha_M]$ represents the decomposition of target data using the dictionary D_n, and S represents the sparsity level. After the reconstruction error E_n is calculated, ΔD_n is estimated by solving the following optimization problem.

$$\min_{\Delta D_n} ||E_n - \Delta D_n\Gamma_n||_F^2 + \lambda||\Delta D_n||_F^2 \quad (4)$$

where the first and second terms ensure better reconstruction error and smoothness of the path, respectively. λ is a regularization parameter to trade-off these two terms. The incremental learning continues until the second term is less than a pre-defined threshold. After learning dictionaries for intermediate domains, the feature vector is constructed as

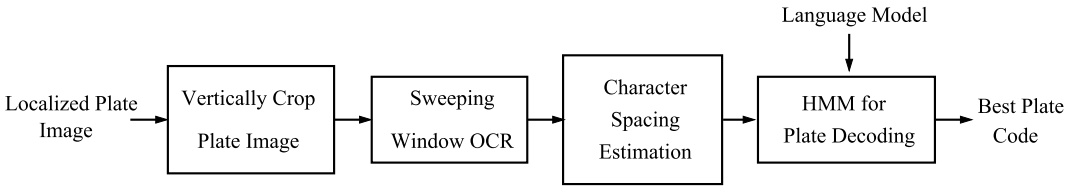


Fig. 9. Schematic of the segmentation-free license plate recognition.

$[(D_0\alpha)^T, (D_1\alpha)^T, \ldots, (D_N\alpha)^T]^T$ for a given data instance where α is the decomposition of the data instance onto training or target domain dictionaries. The calculated feature vectors for the training images are used to train a linear SVM classifier for each character. Note that the main goal in dictionary learning is to learn intermediate domains to find a common representation for the samples in the source and target domains and transform the features into this common representation domain through the use of dictionaries learned for the intermediate domains. OCR classifiers are trained after transforming the features into the common representation domain. Therefore, dictionary learning for feature transformation and training OCR classifiers are sequential processes where the latter follows the former.

IV. Sweeping OCR

Character segmentation is a major contributor to ALPR errors [19]. Figure 8 shows several cases where conventional (e.g., projective) segmentation techniques can result in poor performance. Figure 8 (a) illustrates the complex plate background in Florida State, where the plate logo touches characters, making it difficult for character segmentation. Similarly, Figs. 8 (b) and (c) illustrate other cases where segmentation is challenging due to shadow touching the characters or crossing the plate.

In this section, we present a segmentation-free approach to license plate recognition that replaces two steps that are often a part of traditional ALPR systems (i.e., character segmentation and character recognition) with a single step called sweeping OCR. This was motivated by related work that demonstrated that joint recognition and segmentation outperforms the independent implementation of these tasks [8]. Our method thus, avoids the character segmentation step, which is a critical but error-prone step in current license plate recognition technologies. The proposed method sweeps the OCR classifier across the plate and infers the characters and their locations using a probabilistic inference method based on HMMs. A language model for plates is developed using a corpus of codes with a distribution that is representative of the training and test sets. The plate code candidates from the HMM model is combined with the language model to infer the best plate code.

Figure 9 shows a schematic of our segmentation-free license plate recognition method. The localized license plate image is first vertically cropped and horizontally aligned using a low-level image processing techniques, similar to the operations used in conventional ALPR systems prior to segmentation [4]. After a vertical crop, a fixed window is swept across plate

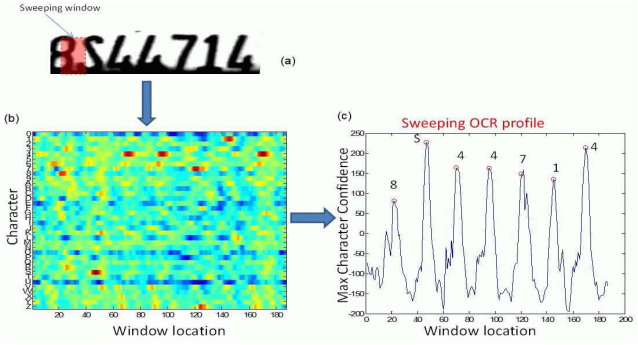


Fig. 10. Sweeping window OCR across a localized license plate image.

image and OCR is applied to the image at each window location. The result is a matrix of character confidences, for each symbol in the OCR dictionary, at each window location. The vertically cropped plate image is resized to a height of 40 pixels and a 40 pixel by 20 pixel window is swept across the image. This window size appears to be well suited for license plates from various states and across multiple jurisdictions. Figure 10 illustrates the process and the results. Figure 10 (a) shows a window (in red) being swept across the plate. Figure 10 (b) is a 2-D map of the OCR confidence matrix. In Figure 10 (c), we plot the maximum confidence at each window location where the peaks represent the set of candidate characters with high confidence.

For decoding the best license plate code of a given code length N, we formulate the problem as

$$\mathbf{c}_N^* = \underset{\mathbf{c}}{\operatorname{argmax}}\, p(c_1, c_2, \ldots, c_N, x_1, x_2, \ldots, x_N) \tag{5}$$

where $c_i \in (A, B, ..., Z, 0, 1, \ldots, 9)$'s represent possible characters from the character set and x_i's represent the corresponding character locations. We then infer the best code length L from

$$L = \underset{N}{\operatorname{argmax}}\, p(\mathbf{c}_N^*) P_{LM}(\mathbf{c}_N^*) \tag{6}$$

and the best license plate code as $\mathbf{c}_L^*$. Here $P_{LM}(\mathbf{c})$ represents the language model which assigns a probability to the plate code based on prior data of actual license plate codes. The language model is a Naive Bayes classifier based on two features: code length and the number of times the code template (LLLNNNN letter/number combination) with the detected starting character has been seen in the training data. In order to account for codes rarely seen in the training data, we allow a longer code with good code probability to override the language model. The intuition behind this is that we trust

Algorithm 1 Viterbi Algorithm for Finding the License Plate Code

Let $\delta(i, j)$ be the probability of a plate character sequence of length i whose i^{th} character is c_j. Let $\gamma(i, j)$ be the probability of a plate character sequence of length i whose i^{th} character is c_j, defined by the lowest probability character in the sequence

procedure GETPLATECODE
  Initialize: $\delta(i, j) = 0$, $\gamma(i, j) = 0$, for $i, j = 1..N$
  for j = 1 to N **do**
    $\delta(1, j) = O(j)$
    $\gamma(1, j) = O(j)$
  end for
  for i = 2 to N **do**
    for j = 1 to N **do**
      $k^* = Argmax_{k=1..j-1}\delta(i-1, k)A(k, j)O(j)$
      $\delta(i, j) = \delta(i-1, k^*)A(k^*, j)O(j)$
      $\gamma(i, j) = Min(\gamma(i-1, k^*), A(k^*, j)O(j))$
      $back(i, j) = k^*$
    end for
  end for
  for i = 1 to N **do**
    $j^* = Argmax_{j=1..N}\delta(i, j)$ - last character for athe most probable plate sequence of length i
    $\gamma_{Max}(i) = \gamma(i, j^*)$
    $code(i)$ = backtrack$(i, j^*, back)$ - backtrack from j^* to get most probable code sequence of length i
  end for
  $L = Argmax_i \gamma_{max}(i) P_{LM}(code(i))$ - best plate length
  $code^* = code(L)$ - best plate code
end procedure

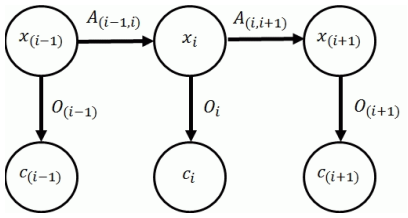


Fig. 11. Graphical representation of HMM model for LP decoding.

the visual evidence (i.e. code probability) more than prior probability when determining which plate code is more likely.

Modeling the multidimensional density function $p(c_1, c_2, \ldots, c_k, x_1, x_2, \ldots, x_N)$, especially in the absence of physically inspired model, however, constitutes a hard task. We therefore, use HMM to model the problem of finding the highest probability sequence and simplify the joint density function in Eq. 5. Figure 11 shows a schematic of the HMM where A represents the transition matrix and O the emission matrix. $A_{j,i}$ represents the transition probability to go from character c_j at x_j to character c_i at x_i. O_i is the OCR probability for character c_i at location x_i. In the HMM formulation, the optimization problem in Eq. 5 reduces to

$$\mathbf{c}^* = \underset{\mathbf{c}}{\operatorname{argmax}} \prod_{i=1}^{N} p(c_i|x_i)\, p(x_i|x_{i-1})\, P_{LM}(\mathbf{c}) \qquad (7)$$

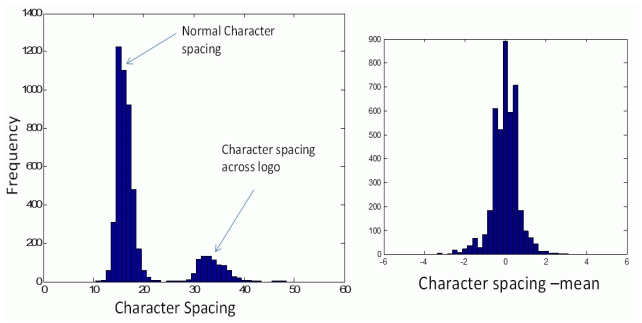


Fig. 12. Character to character spacings for 1000 license plates from different jurisdictions.

Fig. 13. Example of license plate images used in our experiments.

where $p(c_i|x_i)$ and $p(x_i|x_{i-1})$ represent the emission and transition probabilities, respectively. The emission probabilities $(p(c_i|x_i))$ are calculated from the classifiers trained for each character as described in the previous section. We modeled the transition probability as a function of the character spacing $(x_i - x_j)$. A Gaussian with a flat top ($A = 1$, $|x - s| \leq a$ and $A = \exp(-(x-s)^2/2\sigma^2)$, $|x - s| > a$) is used. Here s is the median character spacing from the spacing estimator, and a and σ are free parameters. We use $a = 1$ and $\sigma = 2$ in our implementation, which is roughly consistent with the observations from data as shown in Fig. 12. Note that large deviations from the median spacing is strongly penalized in the transition probability with one exception, the spacing across the logo. For each probable plate code, exactly one large spacing is allowed with transition probability 0.5.

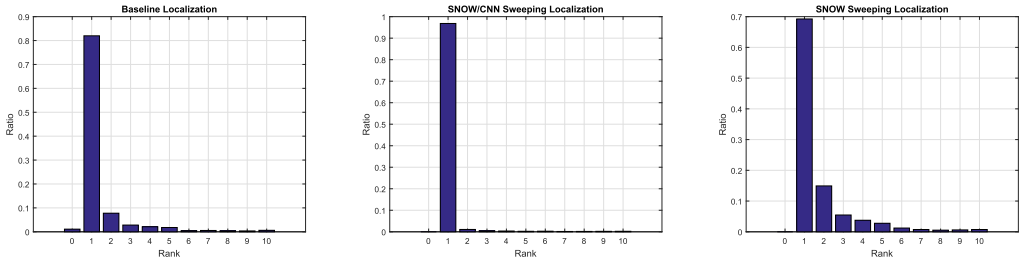
Given the emission and transition probabilities, we decoded the highest probability character sequence using Viterbi algorithm, which leverages dynamic programming to solve the optimization problem in Eq. 7.

V. EXPERIMENTS

In this section, we evaluated the performance of the proposed algorithms for each module of the ALPR system. The algorithms are tested on a set of images, acquired from several jurisdictions and captured under realistic conditions. The images were captured using an NIR camera with an IR illuminator to enable both day and night time vision. The resolution of the captured images were 2048×1628, and contained a license plate region larger than 50×140 pixels. The cameras are triggered by induction loops installed beneath the road as commonly practiced in ALPR systems. Figure 13 shows a set of example images used in our experiments.

A. Plate Localization and Failure Identification

We first evaluated the performance of the proposed localization method on $5,000$ NC images. We used $50,000$ positive



a) Rank of plate containing region using baseline localization [35]

b) Rank of plate containing region using the proposed 2-stage localization

c) Rank of plate containing region using sweeping SNoW classifier

Fig. 14. (a)-(b) Comparison of the proposed 2-stage plate localization algorithm with a baseline method [35] and (c) localization performance using SNoW classifier without applying the strong classifier in the second stage of our proposed method.

(i.e., license plate) and $100,000$ negative (i.e., no-plate) images for training the SNoW classifier. We selected AlexNet as the strong classifier, with weights that were pre-trained on the ImageNet dataset. The AlexNet was then fine-tuned using $16,000$ LP and $63,000$ no-LP images with $10,000$ iterations. The same training set was also used to train a linear-SVM classifier using the AlexNet features. Training the classifiers, we set the SMQT template size to 50×140 (rows, cols), which is the smallest size of license plates encountered in our image set. The SNoW classifier is swept across the input image horizontally in steps of 15 pixels and vertically in steps of 10 pixels. The sweep is conducted at two scales with magnitudes of 1.0 and 0.7. At each location and scale, the resulting classifier score (i.e., confidence) is recorded. Once the sweep is complete, we perform a non-maximum suppression [34] and extract highest score regions from the input image.

We compare the performance of the proposed method with a baseline method described in [35]. Each plate localization method was configured to return 10 candidate ROIs for an input image. Each captured image contained only one plate and the rank of the plate containing ROI was recorded for each test image. The frequency of the ranking is plotted in Figs. 14 (a)-(b) for the baseline and the proposed methods, respectively. Figure 14 (c) shows the performance of the first stage of our proposed method alone, where the ranking is calculated using the confidence of the SNoW classifier.

For the baseline approach in Fig. 14, we see that 83% of the time the top ROI contains a license plate and 8% of the time, the second ROI contains the plate. For the proposed method, however, top ROI contains a license plate over 96% of the time, significantly improving the baseline method. When the strong classifier is omitted from the pipeline, the percentage of the top ROI containing a license plate reduces to 68%, which shows the effectiveness of the strong classifier in the second stage. The average rank for the baseline and the proposed methods is 1.43 and 1.11, respectively. Note that enabling a low rank for plate localization is important in ALPR systems as it ensures faster real time processing of captured imagery. If the plate region is not ranked highly, many non-plate regions need to be processed by steps 3-6 in Fig. 1 before a result is returned. If a plate is found with a high enough confidence, the processing of subsequent regions can be safely aborted.

In order to validate our localization results further, we conducted another experiment on CA images. In this case, we constructed our image set by selecting images that an ALPR system failed to recognize. In this image set, the plates, for instance, are often occluded (bent or occluded by part of the vehicle), have strong shadows cast on them (rendering some or all of the plate unreadable), or overexposed (too bright). Each of the images contained human labeled ground truth that specified if a LP was present. Similarly to plate localization, we fine-tuned a separate AlexNet and used the soft-max classifier for generating confidence scores for each class. As an alternative approach (and as a baseline), a random forest classifier (RF) was trained on Local Binary Pattern (LBP) [36] and Histogram of oriented Gradients [37] (HoG) features that were extracted from each plate region. The dataset consisted of approximately $20,000$ no-LP images and $9,000$ LP images. Training was performed on 75% of the data, while testing was performed on the remainder. The results of both classifiers are given in Figure 15. The performance of our plate localization on this dataset is reasonably high, especially when considering the often poor quality of the plates in these images. Overall, the CNN is able to correctly identify approx. 97% of human readable plates at an expense of 0.5% FPR, while also capable of identifying over 86% of regions that do not have plates at the same FPR. Another result from this figure is the extent by which the CNN outperforms the RF classifier (65% increase in performance for the LP class, and a 52% improvement for the no-LP class). It is worth noting the feature parameters (for LBP and HoG) were selected using a grid search that maximized TPR performance for the no-LP class at 0.5% FPR.

To test the performance of the failure identification portion of our proposal, we used CA images since these images contained human labeled ground truth categorized by the classes of interest (i.e., LP not present, LP too bright, LP too dark or no vehicle). We used 8000 LP not present (NOPL), 500 LP too bright (BRIT), 350 LP too dark (DARK), and 400 no vehicle (RJNV) images to train and test the AlexNet (i.e., 75% used for training, 25% used for testing). Soft-max is used to calculate a probability for each of the classes. We selected probability thresholds for each class separately to determine whether the original image should be classified as having a particular problem. The class with the highest

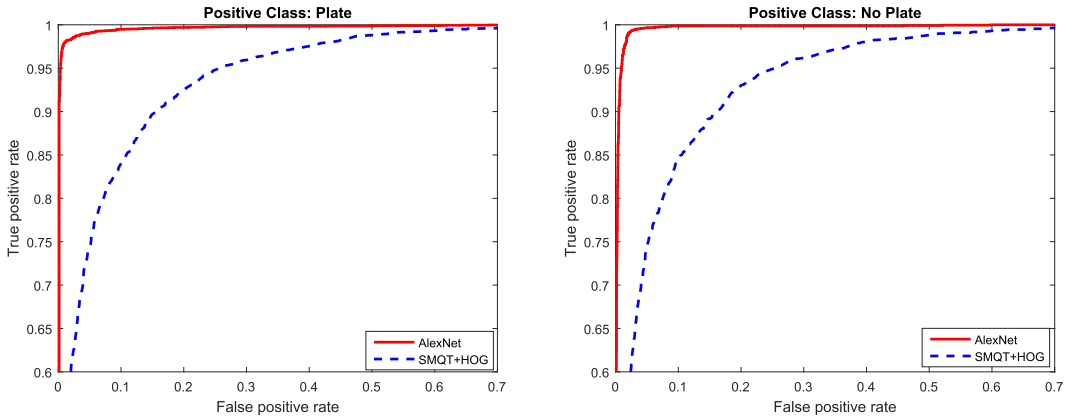


Fig. 15. ROC curves for LP/no-LP classification on images with difficult to read plate that failed a third party ALPR system. (a) ROC curves for the case where positive class is selected as LP class. (b) ROC curves for the case where positive class is selected as no-LP class.

TABLE I
CONFUSION MATRIX SHOWING THE PERFORMANCE OF IMAGE CLASSIFICATION FOR FAILURE IDENTIFICATION

	BRIT	DARK	NOPL	RJNV
BRIT	0.77	0.01	0.22	0
DARK	0	0.33	0.49	0.19
NOPL	0.02	0.01	0.97	0
RJNV	0	0	0	0.99

margin above its threshold is selected as the final classification output. Table I shows the classification performance at FPR of 0.5%. Overall, the performance is $> 90\%$ on the no plate and no vehicle classes, while there is some confusion among the too bright and too dark classes (77% and 32%, respectively). Also note that in the latter case, both of these classes are often confused for the no plate class. One explanation for this confusion is that there is a lack of high frequency information being recovered at the plate area in all three classes (i.e., strong shadows and/or over-exposure results in a flat, uniform texture at the plate region, as can be seen in Figure 5).

B. Character Recognition and Domain Adaptation

We have evaluated the performance of the proposed training methodology across character images acquired from CA plates. In our experiment, we collected 2500 real samples for each character. We also generated 2000 synthetic images per character using the methodology described in [33]. 1500 out of 2500 real sample per character were used to train the classifier and the rest 1000 samples were used for testing. The images are scaled to 48×24 before feature extraction.

1) Comparative Study for Character Recognition: We performed a comparative OCR study on segmented characters across various features (i.e., SMQT, histogram of oriented gradients (HOG), and, LeNet-5) and classifiers (i.e., SNoW and SVM). We trained one-vs-all classifiers for each character on a dataset generated from CA plates, where we collected 2500 real samples for each character. We also generated 2000 synthetic images per character. 1500 out of 2500 real sample per character were used to train the classifiers and the rest 1000 samples were used for testing. We trained a SNoW classifier with the SMQT features as HOG and LeNet features were coupled with a linear-SVM classifier. The images were scaled to 48×24 before feature extraction. The cell size in feature extraction stage for HOG was set to 8×8 which results in a feature vector of size 360 per character image while the resulting feature vector length was 409600 for SMQT features.

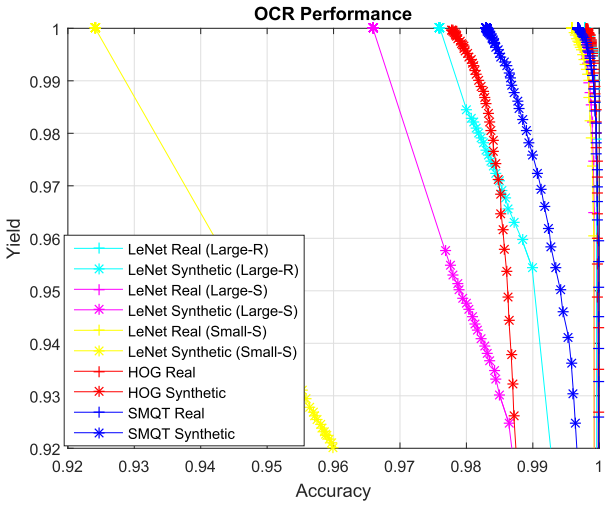


Fig. 16. Accuracy-yield curves for character recognition across different features and classifiers.

Fig. 16 shows the accuracy-yield curves across different features and classifiers. The yield is calculated as the fraction of conclusions returned by the OCR engine versus the number of queries made. The OCR engine is provided a confidence requirement and has the option of not returning an answer for cases where this requirement is not satisfied. The accuracy is defined as the fraction of correct results from the total number of conclusions returned. By varying the confidence requirement, we can effectively trade off yield versus accuracy as can be seen in Fig. 16. For each of the HOG and SMQT features, we plotted 2-curves called HOG Real (plus red curve), HOG Synthetic (star red curve), SMQT Real (plus blue curve), and SMQT Synthetic (star blue curve),

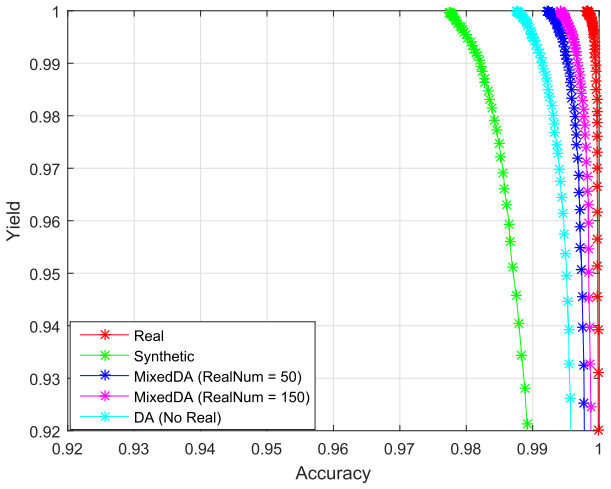


Fig. 17. Accuracy-yield curves for character images, acquired from CA plates.

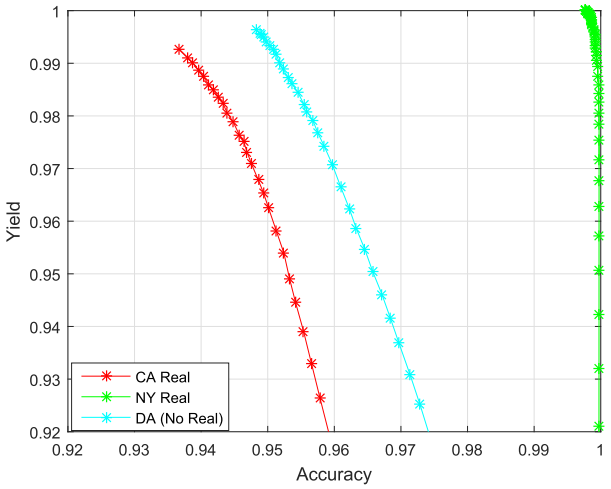


Fig. 18. Accuracy-yield curves for character images, acquired from NY plates.

where synthetic or real indicate the images used in training. As expected, when the classifiers were trained using synthetic images, the performance is lower than the case when the classifiers were trained using real images for both HOG and SMQT features. Both features show comparable performance when the classifiers were trained with real images but SMQT features showed slightly better performance than HOG when the classifiers were trained using synthetic images.

We also evaluate the performance of LeNet on the same dataset as LeNet was originally designed for character recognition [38], [39]. In this case, we scaled the images to 32×32 to make them compatible with the first layer of the LeNet architecture. The features were extracted from the last fully-connected layer, which were than used to train a linear-SVM classifier. For LeNet, we performed 3 different experiments for pre-training the network and repeat each experiment for training classifiers using synthetic and real images. In the first experiment, we pre-trained the LeNet using CA synthetic images (i.e., 2000 images per character) and used this pre-trained network as the feature extractor for training

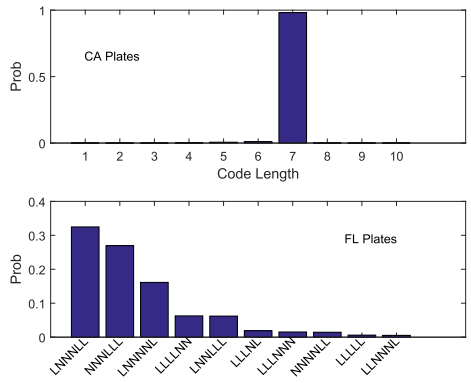


Fig. 19. Language models for CA (top) and FL plates (bottom).

linear-SVM classifiers using CA synthetic and CA real images. The resulting classifiers were tested on the CA real images similar to HOG and SMQT features. The performances for these classifiers are shown as LeNet Synthetic (Small-S) and LeNet Real (Small-S) in the figure. In the second experiment, we pre-trained the LeNet using 550,000 synthetic character images uniformly generated for 11 States (i.e., TX, FL, NJ, CA, IL, MA, CT, IN, NY, PA, MD) in USA and used the pre-trained network as the feature extractor for training SVM classifiers using CA synthetic and CA real images. The results for this case were shown as LeNet Synthetic (Large-S) and LeNet Real (Large-S) in the figure. In the last experiment, we pre-trained the LeNet using 450,000 real character images collected from 8 US States (i.e., TX, NC, NJ, CT, NH, MA, PA, NY) and used the pre-trained network as the feature extractor for training SVM classifiers using CA synthetic and CA real images. LeNet Synthetic (Large-R) and LeNet Real (Large-R) show the performances for this case in the figure. Note that the performances for LeNet Real (Small-S), LeNet Real (Large-S), LeNet Real (Large-R) are comparable and similar to HOG and SMQT performance when the classifiers were trained with CA real images. However, LeNet Synthetic (Small-S), LeNet Synthetic (Large-S), and LeNet Synthetic (Large-R) were outperformed by SMQT and HOG features when the classifiers were trained with CA synthetic images.

2) Evaluation of Domain Adaptation: In this section, we have evaluated the performance of the proposed training methodology across character images acquired from CA plates. We used the same dataset generated from CA plates as described in Sec. V-B.1.

Due to its strong OCR performance in the previous section,[1] we extracted HOG features from the character images. Using the extracted features we trained linear SVM classifiers in a one-vs-all fashion for each character for both synthetic and real images. After feature extraction we applied domain adaptation on training and target domains when synthetic images are used for training. In the domain adaptation, we used 720 synthetic and 360 real images for learning dictionaries that represent the intermediate domains as described in Sec. III-B. We set

[1]Note that HOG and SMQT performances are comparable but significantly better than LeNet performance especially when the classifiers are trained with synthetic images

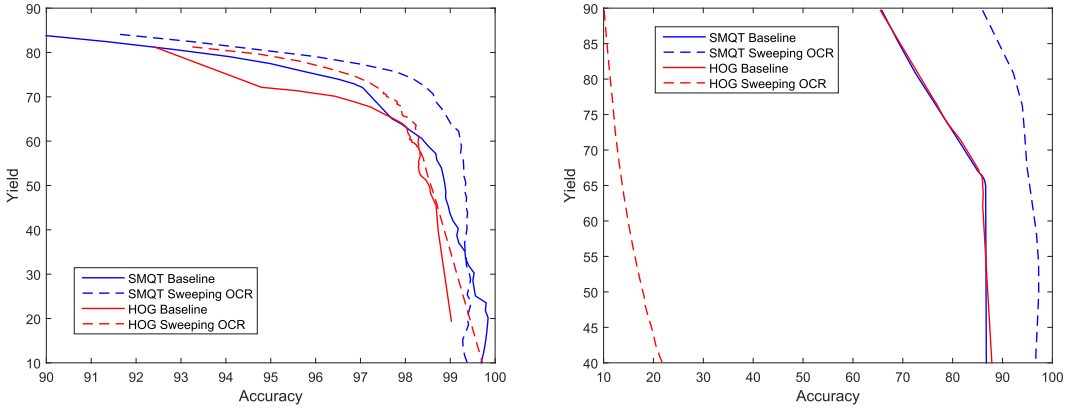


Fig. 20. ROC for CA dataset (left) and FL dataset (right) with baseline approach and sweeping OCR approach. (a) ROC for CA images. (b) ROC for FL images.

the dictionary size to 240, regularization parameter λ to 500 and sparsity level S to 40. The stopping criteria σ for the number of sampling is set to 0.4 to stop the algorithm when $||\Delta D_n||_F^2$ is less than σ. This stopping criteria yielded to sample 5 intermediate domains and resulted in a feature vector of size 2160 after the domain adaptation. Figure 17 shows the accuracy-yield curves for character images, acquired from CA plates, using synthetic (green curve) and real images (red curve) for training. Note that applying domain adaptation on source (synthetic) and target (real) domains improves the performance (cyan curve) and including some target domain samples in training along with domain adaptation further improves the performance (blue and magenta curves).

In order to validate our results further, we performed another experiment on NY plates. In this experiment, instead of using synthetic images, we used CA images as the source and NY images as the target and repeat the first experiment. Figure 18 shows the accuracy-yield curves for character images, acquired from NY plates, using CA (red curve) and NY images (green curve). Similar to the first experiment, applying domain adaptation across domains improves the OCR performance (i.e., compare cyan and red curves). Note that the gap between domain adaptation (cyan curve) and the best case (green curve) in Fig 18 is much larger than the gap in Fig. 17. The reason for the larger gap is due to the font difference in CA and NY plates, which is not an issue for artificially generated license plate images. In a practical deployment, given many alternative source domains (i.e., synthetic images, images acquired from other jurisdictions or countries), the best domain can be selected based on the domain shift calculated between target and source domains.

C. Performance on Sweeping OCR

In this section we describe the experiments for evaluating the performance of the sweeping OCR. We considered 2 datasets: (a) a CA dataset consisting of 6346 images, and (b) FL dataset consisting of 805 images. Each image was preprocessed to localize the plate. For baseline performance, the localized plate image was segmented around the characters using projective segmentation, followed by an OCR of the segmented character images. The baseline OCR classifier was trained on a set of segmented character image samples from the respective states (i.e., CA and FL). For the sweeping OCR method, the OCR has to be trained to reject partial characters, in-between regions between two adjacent characters, and logo regions (in the case of FL). This is done by adding examples of partial characters, in-between characters and logo regions to the negative set of each character classifier.

Motivated by their superior performance, we considered both HOG and SMQT features to train the OCR classifiers. For HOG features, we trained an SVM classifier, while for SMQT, we trained a SNoW classifier. We used SMQT on a 3×3 cell, which for a 40×20 image gives a feature length of $512 \times 40 \times 20$ compared to feature length of just 360 for HOG.

The inference step in the sweeping OCR method (Eq. 7) makes use of a language model. The language model is derived from a corpus of previously observed plate codes, and can be customized for each state. For CA, we used a simple language model based on the likely-hood of the code length (Fig 19 top), and observed that a code length of 7 is extremely likely. For FL, we used a language model based on the likely hood of the code's letter (L) number (N) sequence (Fig 19 bottom). For instance, LLNNNN would correspond to a code of length 6, with the first two being letters, the rest being numbers.

Fig 20 shows the ROCs for the CA and FL dataset. The CA plates do not have any background, while the FL plates have a fairly complex background image. For the CA dataset, while the baseline approach (with projective segmentation) performed well, we observed that the sweeping OCR method still outperformed the baseline. The SMQT features were only marginally better than HOG, especially considering the large disparity in feature length. For the FL dataset, the performance of the baseline approach was considerably poorer, sweeping OCR with SMQT showed a significant improvement in performance over baseline. However, the sweeping OCR with HOG features performed poorly, suggesting the inability of the HOG features to sufficiently contrast the character

regions with non-character regions, which is essential for the sweeping OCR to work.

Finally, we tested our end-to-end system performance on a 10, 000 LP dataset and achieved a performance higher than 99% accuracy at an automation (yield) rate 80%. In our system implementation, NIR cameras were mounted on gantries at toll booths, where the cameras were triggered by in-ground sensors buried under the road. The captured images were transferred and processed on a processing unit that included a Xeon x5650 processor, 24GB RAM, and GTX 570 GPU. The processing time for each image was on average 2 seconds, which was fast enough to keep up with the traffic flow. For each image, on average, 80% of the execution time was spent on the segmentation free OCR module. Plate localization and failure identification modules took only 10% of the execution time and the remaining 10% was spent in the rest of the operations. Extracting features and calculating classification scores in a sweeping window fashion takes most of the processing time in the segmentation free OCR module. Please note that the average computational times for the several state-of-the-art ALPR systems have been reported in [22], which were tested on a computer with a 4GB memory and a 2.8GHz processor. The algorithms were implemented in C++ and the average processing times were reported as 0.8, 5, 0.6, 3, 1, and 1.5 seconds per image for the algorithms in [6], [7], [22], [25], [40], and [41], respectively. Our proposed algorithm is comparable with these ALPR algorithms in terms of processing time while addressing several other challenges in ALPR systems.

VI. Conclusion and Discussion

In this paper, we propose a new end-to-end ALPR system that can be scaled to several US jurisdictions and countries with minimal manual annotation and human intervention. This paper presents several key findings.

The cascaded approach, that first uses a weak SNoW classifier followed by a strong CNN classifier, shows significant improvement for localizing the plate region in LP images. It also enables a second stage analysis for the images that fail the primary confidence test to identify the reason for failure, which can significantly reduce the number of images that are sent to human review by eliminating unreadable/no-revenue LP images from the set. Classifying the localization failures can also help identifying the possible problems in an ALPR hardware system and hence, enables quicker response-time to fix the problem. We note that our 2-stage architecture for license plate detection resembles the-state-of-the-art object detection algorithm called R-CNN [42], which also first identifies a set of candidate regions and applies CNN on each candidate region to detect objects in an image. R-CNN uses a generic approach as a measure of objectness to identify candidate regions in an image. When this generic objectness measure is applied to license plate images [43], it results in a very high miss rate as license plate structure is significantly different from typical objects with closed boundaries and relatively smooth interior.

Minimizing manual annotation and data collection required for training OCR classifiers in new jurisdictions/countries plays a key role in reducing the installation time for ALPR systems. When an OCR engine is trained using either artificially generated synthetic images or character samples acquired from other sites, the OCR performance in the field is degraded due to the mismatch between training and target data distributions. An unsupervised domain adaptation estimates the domain shift between training and target domains and improves the OCR performance without requiring manual annotation in new deployments. The estimated domain shifts between target and multiple source domains enable to select the domain that yields the best OCR performance. We show that synthetic license plate image generation algorithms along with the recent advances in domain adaptation literature paves the way through annotation free license plate recognition systems.

Character segmentation and recognition modules, typically independent in ALPR systems, can be merged to efficiently decode the plate code using a sweeping OCR engine across the localized LP images. The calculated scores by the sweeping OCR can then be used in a probabilistic inference based on HMM and Viterbi decoding to find the highest probability code sequence using a language model. This segmentation-free approach overcomes the limitations of character segmentation in jurisdictions (e.g., FL and NC) with difficult plate layout, where the plate logo touches the characters.

References

[1] O. Bulan, V. Kozitsky, and A. Burry, "Towards annotation free license plate recognition," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2015, pp. 1495–1499.

[2] O. Bulan, S. Wshah, P. Ramesh, V. Kozitsky, and A. Burry, "Usdot number localization and recognition from vehicle side-view nir images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jan. 2015, pp. 91–96.

[3] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic license plate recognition (ALPR): A state-of-the-art review," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 2, pp. 311–325, Feb. 2013.

[4] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, I. D. Psoroulas, V. Loumos, and E. Kayafas, "License plate recognition from still images and video sequences: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 3, pp. 377–391, Sep. 2008.

[5] V. Abolghasemi and A. Ahmadyfard, "An edge-based color-aided method for license plate detection," *Image Vis. Comput.*, vol. 27, no. 8, pp. 1134–1142, 2009.

[6] C. N. E. Anagnostopoulos, I. E. Anagnostopoulos, V. Loumos, and E. Kayafas, "A license plate-recognition algorithm for intelligent transportation system applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 3, pp. 377–392, Sep. 2006.

[7] Y. Wen, Y. Lu, J. Yan, Z. Zhou, K. M. von Deneen, and P. Shi, "An algorithm for license plate recognition applied to intelligent transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 3, pp. 830–845, Mar. 2011.

[8] X. Fan and G. Fan, "Graphical models for joint segmentation and recognition of license plate characters," *IEEE Signal Process. Lett.*, vol. 16, no. 1, pp. 10–13, Jan. 2009.

[9] C.-N. E. Anagnostopoulos, "License plate recognition: A brief tutorial," *IEEE Intell. Transp. Syst. Mag.*, vol. 6, no. 1, pp. 59–67, Jan. 2014.

[10] C. A. Rahman *et al.*, "A real time vehicle's license plate recognition system," in *Proc. AVSS*, 2003, p. 163.

[11] S.-Z. Wang and H.-J. Lee, "Detection and recognition of license plate characters with different appearances," in *Proc. IEEE Intell. Transp. Syst.*, vol. 2. Oct. 2003, pp. 979–984.

[12] H.-J. Lee, S.-Y. Chen, and S.-Z. Wang, "Extraction and recognition of license plates of motorcycles and vehicles on highways," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, vol. 4. Aug. 2004, pp. 356–359.

[13] J. Xu, S. Li, and Z. Chen, "Color analysis for chinese car plate recognition," in *Proc. IEEE Int. Conf. Robot., Intell. Syst., Signal*, vol. 2. Oct. 2003, pp. 1312–1316.

[14] J. Matas and K. Zimmermann, "Unconstrained licence plate and text localization and recognition," in *Proc. IEEE Intell. Transp. Syst.*, Sep. 2005, pp. 225–230.

[15] V. S. L. Nathan, J. Ramkumar, and S. K. Priya, "New approaches for license plate recognition system," in *Proc. Int. Conf. Intell. Sens. Inf.*, Jan. 2004, pp. 149–152.

[16] H.-K. Xu, F.-H. Yu, J.-H. Jiao, and H.-S. Song, "A new approach of the vehicle license plate location," in *Proc. 6th Int. Conf. Parallel Distrib. Comput. Appl. Technol. (PDCAT)*, Dec. 2005, pp. 1055–1057.

[17] H. Caner, H. S. Gecim, and A. Z. Alkar, "Efficient embedded neural-network-based license plate recognition system," *IEEE Trans. Veh. Technol.*, vol. 57, no. 5, pp. 2675–2683, Sep. 2008.

[18] H. Zhang, W. Jia, X. He, and Q. Wu, "Learning-based license plate detection using global and local features," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, vol. 2. Aug. 2006, pp. 1102–1105.

[19] F. Abtahi, Z. Zhu, and A. M. Burry, "A deep reinforcement learning approach to character segmentation of license plate images," in *Proc. 14th IAPR Int. Conf. Mach. Vis. Appl. (MVA)*, May 2015, pp. 539–542.

[20] A. M. Al-Ghaili, S. Mashohor, A. R. Ramli, and A. Ismail, "Vertical-edge-based car-license-plate detection method," *IEEE Trans. Veh. Technol.*, vol. 62, no. 1, pp. 26–38, Jan. 2013.

[21] Q. Li, "A geometric framework for rectangular shape detection," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 4139–4149, Sep. 2014.

[22] J. Dun, S. Zhang, X. Ye, and Y. Zhang, "Chinese license plate localization in multi-lane with complex background based on concomitant colors," *IEEE Intell. Transp. Syst. Mag.*, vol. 7, no. 3, pp. 51–61, Fall 2015.

[23] W. Zhou, H. Li, Y. Lu, and Q. Tian, "Principal visual word discovery for automatic license plate detection," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 4269–4279, Sep. 2012.

[24] J. M. Guo and Y. F. Liu, "License plate localization and character segmentation with feedback self-learning and hybrid binarization techniques," *IEEE Trans. Veh. Technol.*, vol. 57, no. 3, pp. 1417–1424, May 2008.

[25] A. H. Ashtari, M. J. Nordin, and M. Fathy, "An Iranian license plate recognition system based on color features," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1690–1705, Aug. 2014.

[26] M. Nilsson, J. Nordberg, and I. Claesson, "Face detection using local SMQT features and split up snow classifier," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, vol. 2. Apr. 2007, p. 589.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[28] J. Ni, Q. Qiu, and R. Chellappa, "Subspace interpolation via dictionary learning for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 692–699.

[29] H. Hoessler, C. Wöhler, F. Lindner, and U. Kreßel, "Classifier training based on synthetically generated samples," in *Proc. 5th Int. Conf. Comput. Vis. Syst.*, Bielefeld, Germany, 2007.

[30] J. A. Rodríguez-Serrano and F. Perronnin, "Handwritten word image retrieval with synthesized typed queries," in *Proc. 10th Int. Conf. Document Anal. Recognit. (ICDAR)*, Jul. 2009, pp. 351–355.

[31] J. A. Rodríguez-Serrano, F. Perronnin, J. Lladós, and G. Sánchez, "A similarity measure between vector sequences with application to handwritten word image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jan. 2009, pp. 1722–1729.

[32] A. Mecocci and C. Tommaso, "Generative models for license plate recognition by using a limited number of training samples," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2006, pp. 2769–2772.

[33] R. Bala *et al.*, "Image simulation for automatic license plate recognition," *Proc. SPIE*, vol. 8305, p. 83050Z, Feb. 2012.

[34] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, vol. 3. 2006, pp. 850–855.

[35] Z. Fan, V. Kozitsky, and A. M. Burry, "Method and system for identifying a license plate," U.S. Patent 8 792 682, Jul. 29, 2014.

[36] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[37] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893.

[38] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[39] F. Lauer, C. Y. Suen, and G. Bloch, "A trainable feature extractor for handwritten digit recognition," *Pattern Recognit.*, vol. 40, no. 6, pp. 1816–1824, 2007.

[40] S.-L. Chang, L.-S. Chen, Y.-C. Chung, and S.-W. Chen, "Automatic license plate recognition," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 1, pp. 42–53, Mar. 2004.

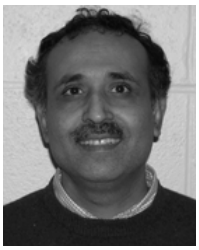
[41] B. Li, B. Tian, Q. Yao, and K. Wang, "A vehicle license plate recognition system based on analysis of maximally stable extremal regions," in *Proc. 9th IEEE Int. Conf. Netw., Sens. Control (ICNSC)*, Apr. 2012, pp. 399–404.

[42] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 580–587.

[43] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Apr. 2013.

Orhan Bulan (SM'16) received the B.S. degree in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in 2006, and the M.S. and Ph.D. degrees in electrical and computer engineering from University of Rochester, Rochester, NY, USA, in 2007 and 2012, respectively. He is currently a Senior Artificial Intelligence Computer Scientist at General Motors. His current research interests include machine learning and computer vision for autonomous vehicles. He holds over 15 patents with over 20 patent applications pending. He has co-authored over 35 papers in peer-reviewed journals and conference proceedings. He is a member of the Society of Photo-optical Instrumentation Engineers. He is also a Frequent Reviewer of IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY.

Vladimir Kozitsky (M'–) received the bachelor's and master's degrees in electrical and computer engineering from Cornell University in 2004 and a second master's degree in computer science from University of Rochester in 2014. He is a Researcher with the Video and Image Analytics Laboratory, PARC, where he involves in the automatic extraction of actionable intelligence from images and videos. A key element of the work involves designing robust and scalable computer vision algorithms that accurately and infrequently solicit input from humans and thus optimally leverage the complement of human and machine intelligence. He leads one of several research projects exploring vision-based applications in the field of transportation. He holds 15 U.S. patents with 20 pending applications. His areas of expertise include machine learning, pattern recognition, intelligent sensing, and image and video processing. He is a Peer Reviewer for *Journal of Electronic Imaging*.



Palghat Ramesh (M'–) received the bachelor's degree in mechanical engineering from IIT Madras, Chennai, India, the Ph.D. degree from Cornell University, and the master's degree in computer science and data analytics from University of Rochester. He also served as an Adjunct Professor with University of Rochester's Mechanical Engineering Department from 1999 to 2004, where he taught courses in heat transfer. He is currently a Principal Scientist with PARC, background in computational physics and data analytics. He has authored or co-authored over 50 publications in peer-reviewed journals and conference proceedings. He holds 53 U.S. patents. He is interested in combining machine learning algorithms with ideas borrowed from behaviors of physical systems for modeling and simulation applications, and involves in computer vision projects related to license plate recognition, transportation analytics on traffic flow modeling, and customer care analytics on user interest mining. He received the Xerox Excellence in Science and Technology Award in 1999 for his contributions to the iGen3 printer's development system.

Matthew Shreve (M'–) received the M.S. degree in mathematics from Youngstown State University and the Ph.D. degree in Computer Science from University of South Florida. Since joining Xerox in 2013, he has been involved in several projects including recognizing the body gestures and facial expressions of customers in a retail environment, and outdoor vehicle monitoring, which includes tracking and recognizing criminal activity, such as vandalism. He is also an Adjunct Professor with the Department of Electrical and Computer Engineering, University of Rochester. His research interests include computer vision, image processing, pattern recognition, machine learning, and artificial intelligence. He has served on the organizing committee of top IEEE conferences, such as the International Conference on *Pattern Recognition*. He has received several professional awards, including the IEEE International Transactions on Transportation Systems Best Journal Award from 2010 to 2011 and the USF Engineer Best Poster Award in 2009. He also serves as a Reviewer/Referee for several journals and conferences, such as *Pattern Recognition Letters*, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B, *Transportation Research—Part C*, *Signal Image and Video Processing*, *The Journal on Electronic Imaging*, and *Intelligent Transportation Systems*.