WEIGHTED EDIT DISTANCE FOR COUNTRY CODE RECOGNITION IN LICENSE PLATES

Kateryna Chumachenko[⋆] *Alexandros Iosifidis*[†] *Moncef Gabbouj*[⋆]

[⋆] Faculty of Information Technology and Communication Sciences, Tampere University, Tampere, Finland
[†]Department of Electrical and Computer Engineering, Aarhus University, Aarhus, Denmark

ABSTRACT

This paper presents the problem of country code recognition from license plate images. We propose an approach based on character detection and subsequent clustering for country code localization. We further propose three weighted Edit Distance metrics for country of origin prediction from imperfect detections, namely based on character similarity, detection confidence, and relative operation importance. Experimental results show the benefit of proposed approaches on real-world data. The proposed method is lightweight and independent of the underlying object detector, facilitating its application on edge devices.

***Index Terms*—** License plate recognition, edit distance, Levenshtein distance, country code recognition

1. INTRODUCTION

Automatic License Plate Recognition (ALPR) from images is an essential problem within a wide area of applications, ranging from road traffic surveillance and security control in restricted areas to vehicle re-identification and verification in industrial and smart city, and security applications. Industrial needs along with rapid development of the computer vision field lead to fast emergence of novel ALPR methods, ranging from the ones relying on layout-specific heuristic rules to the ones based on object detection. Nevertheless, majority of the modern ALPR systems are limited to recognition of the license plate code itself, while omitting other potentially useful information present in the image, such as the country code. Information of this kind is particularly important for European license plates that have similar layouts and hence the origin country of the license plate can only be determined through country code recognition. This task is, however, more difficult, as country codes are represented by characters of small size that are challenging to recognize, resulting in imperfect predictions especially in real-world environments under the presence of various image degradation factors, such as motion blurring, poor lighting, or fog. Consequently, there is a need for development of techniques that are able to compensate for imperfect character recognition and predict the license plate country of origin based on inexact and potentially faulty codes.

In this paper, we introduce the problem of the license plate country of origin recognition from the country code. First, we show how country code localization can be achieved from detection-based character recognition. Second, we propose approximate string matching for country of origin recognition from imperfect country code detections, and several re-weighting schemes for Edit Distance (ED) [1]. Specifically, we propose three weighting schemes that take advantage of visual similarity of different characters, confidence scores predicted by detector, and individual importance of each operation in ED. The proposed weighted metrics can be applied to any other string matching tasks in detection-based Optical Character Recognition. Importantly, the proposed approach is lightweight and independent of the underlying object detector, allowing its applications on edge devices. Experimental evaluation on 1158 real-world images shows the superiority of the proposed weighted approaches as opposed to conventional unweighted variant.

This work was supported by Business Finland project 5G-VIIMA. A. Iosifidis acknowledges funding from the EU H2020 research and innovation program under grant agreement No. 957337 (MARVEL).

LP code	CC	ED	Proposed approach
MB5623R	SLO	SLOVENIA	SLOVENIA
54JVB	ES	ESTONIA	ESTONIA
GHF273	GY	BELARUS	CYPRUS
TC409	FIHI	BOSNIA & HERZEGOVINA	FINLAND
M497T89	HU	CROATIA	RUSSIA

Fig. 1. Examples of country recognition. LP code and CC correspond to detected license plate and country code characters. ED and Proposed approach correspond to predicted country of origin using unweighted Edit Distance and our proposed approach, respectively. Green and underlined corresponds to correct country prediction.

2. RELATED WORK

2.1. License Plate Recognition

The task of Automatic License Plate Recognition (ALPR) can be roughly divided into several subtasks: vehicle detection, license plate (LP) detection, and optical character recognition (OCR). In our work, we focus on the last step of the pipeline, namely, OCR. The majority of ALPR systems nowadays rely on Convolutional Neural Networks due to their outstanding performance in computer vision tasks, and majority of the OCR methods follow the detection-based approach. A generic object detector is trained to detect individual characters which subsequently comprise the license plate code. In [2] the authors perform character recognition using a modified YOLO detector [3, 4, 5], followed by certain heuristic rules designed specifically for Brasilian LPs. YOLO is a popular choice among object detectors employed in various OCR systems in the context of ALPR [3, 6, 7, 8]. In addition, several methods improve the OCR performance by means of auxiliary tasks. In [7] the authors jointly address denoising and rectification tasks in a multi-task manner, and in [3] the authors propose to employ adversarial super-resolution prior to YOLO-based character recognition. To the best of our knowledge, no attempts have been made for recognition of the country codes of license plates. The work in [9] takes a step towards this area by classifying the layout of the license plate and

further applying certain heuristic rules to refine the license plate prediction. The layout type is, however, limited to 5 regions.

2.2. Approximate string matching

Approximate string matching is a task of finding closest matches between pairs of strings that are not precisely equal. One of the common metrics utilized for this task is the Levenshtein distance (alternatively referred to as Edit Distance, while the latter can refer to a wider set of metrics) [1] that calculates the number of operations required for transforming one string to another. A set of operations constitutes insertion, deletion, and substitution. Alternatively, some extensions expand this set with the transposition operation [10, 11]. In a general formulation, Edit Distance between two strings $\mathbf{x}$ and $\mathbf{y}$ of arbitrary lengths is defined as:

$$\begin{aligned} d(\mathbf{x} \rightarrow \mathbf{y}) \; &= \; min(d(i-1, j-1) + \gamma(\mathbf{x}_i \rightarrow \mathbf{y}_j), \\ &\quad d(i-1, j) + \gamma(\mathbf{x}_i \rightarrow 0), \\ &\quad d(i, j-1) + \gamma(0 \rightarrow \mathbf{y}_j), \end{aligned} \tag{1}$$

where $d(i, j)$ is the Edit Distance between substrings $\mathbf{x}[1, ..., i]$ and $\mathbf{y}[1, ..., j]$, $\gamma(\mathbf{x}_i \rightarrow \mathbf{y}_j)$ is the cost of substitution of character $\mathbf{x}_i$ to $\mathbf{y}_j$, $\gamma(\mathbf{x}_i \rightarrow 0)$ is the cost of deletion of $\mathbf{x}_i$, and $\gamma(0 \rightarrow \mathbf{x}_i)$ is the cost of insertion of $\mathbf{x}_i$. In a simple formulation, cost of every operation is set to 1, i.e., $\gamma(\mathbf{x}_i \rightarrow \mathbf{y}_j) = 0$ if $\mathbf{x}_i = \mathbf{y}_j$, and $\gamma(\mathbf{x}_i \rightarrow \mathbf{y}_j) = 1$, otherwise [12].

The original definition of Levenshtein distance [1] is defined by the above-described unweighted formulation with unit cost of every operation. Nevertheless, the idea of re-weighting the costs of operations in Edit Distance is not new and multiple authors have proposed various weighting techniques that have shown to be useful in a variety of tasks [13, 14, 15]. In the context of license plate recognition, Edit Distance has been primarily used for vehicle re-identification based on the license plates. In [12] the authors propose a weighted Edit Distance based on probabilities of character misclassification in license plate recognition, for vehicle re-identification between two cameras. In [16] the authors evaluate various approximate string matching methods for vehicle re-identification, and propose to combine the weighted Edit Distance with n-Grams distance.

3. PROPOSED APPROACH

This section describes the proposed country of origin recognition approach[1]. We assume that the license plate is already detected from the image and operate on license plate images directly. A multitude of well-performing solutions have been proposed for vehicle and license plate detection [2, 5, 8], ranging from generic object detectors to license plate detectors capable of unwarping frontal images [8]. Any of them can be employed to complement our proposed approach in order to obtain an end-to-end system.

3.1. Character Detection

We follow the detection-driven approach for character recognition, and utilize CenterNet [17] as our character detector. The choice of CenterNet is dictated by its outstanding speed vs. performance ratio, as compared to YOLO that is often utilized in ALPR systems [2, 3, 6, 7, 8]. Nevertheless, note that development of a state-of-the-art OCR detector is not the goal of this paper. Instead, we use it as a baseline for evaluation of our subsequent methodology, and any other detection method from the wide range of methods proposed for OCR [2, 3, 6, 7, 8] can be equally utilized if trained on appropriate data, or the proposed CenterNet can be augmented with auxiliary tasks [3, 7] to improve the performance of the detector itself.

The CenterNet is trained to predict a bounding box along with a confidence score and class label from 36 classes, i.e., Latin alphabet letters and numbers. However, availability of appropriately annotated datasets for license plate recognition is limited to country-specific ones. Besides, to the best of our knowledge, none of these datasets provides annotations for the country codes. For this reason, we resort to training the detector on large amounts of artificially generated images that comprise of characters of different sizes. More details on training data can be found in Section 4. Following pre-training on artificial data, we manually annotate a small number of real-world license plate images and fine-tune the detector model on them. Following the detection, we additionally perform non-maximum suppression between classes with high Intersection over Union (IoU) threshold. Since we can safely assume that license plate characters are not overlapping, this allows to further reduce the ratio of false-positive detections.

3.2. Country code identification

Given the detected characters obtained by object detector, the goal is to first identify the detections corresponding to the license plate code, the ones corresponding to the country code, and the miscellaneous ones that correspond to other symbols potentially present on the license plate (e.g., an area code) or misdetections. Since the license plates of different countries often have drastically different layouts, the use of heuristics rules for country code localization is unreasonable. Instead, we follow a two-step clustering-based process. It is safe to assume that the detections corresponding to the license plate code are the ones with the largest height. Therefore, at a first step we apply K-Means clustering [18, 19] to cluster the detected bounding boxes into 2 clusters using the height of each bounding box as a feature descriptor. Out of the obtained clusters, we compare their average heights and merge the clusters if the heights are not distinct enough (specifically, if the ratio of average heights is less than 1.25). The bigger cluster is then selected as a license plate code.

Further, within the smaller height cluster, we seek to filter out the miscellaneous detections, i.e., those not corresponding to the country code. Note that these are not necessarily false-positive detections, but rather the redundant characters potentially present in the license plate, hence metrics such as confidence score cannot be utilized for their filtering. We use the width-height normalized top-left coordinate pairs $(\frac{x_j^{left}}{w_j}, \frac{y_j^{top}}{h_j})$ of each bounding box as feature descriptors and perform hierarchical single-linkage clustering [20] based on Euclidean distance. As a result, we obtain N distinct clusters of characters in the license plate. Further, we use the weighted Edit Distance metric (described further) to select the cluster with the lowest distance to any of the known country codes. In the case where several clusters have equal distance to certain country codes, we select the one with the highest average confidence as the prediction.

3.3. Weighted Edit Distance Metric

Clearly, detection of country code characters is a challenging task due to their small size. Especially in harsh real-world environments, where images are susceptible to illumination and contrast changes, skewing and blurring, imperfect country code character detection is highly probable. Therefore, we rely on approximate string matching

[1]The implementation along with the trained models and data used for training the detector will be made publicly available.

to account for these imperfections. Specifically, we propose to use the Edit Distance for matching the country codes with the dictionary of known existing country codes. The standard Levenshtein distance (i.e., unweighted ED) assumes equal costs between all the operations. Nonetheless, such assumption is not necessarily reasonable in the context of OCR. In this section, we describe several re-weighting strategies that can be employed for assigning costs to operations and, hence, improving the performance of country code matching.

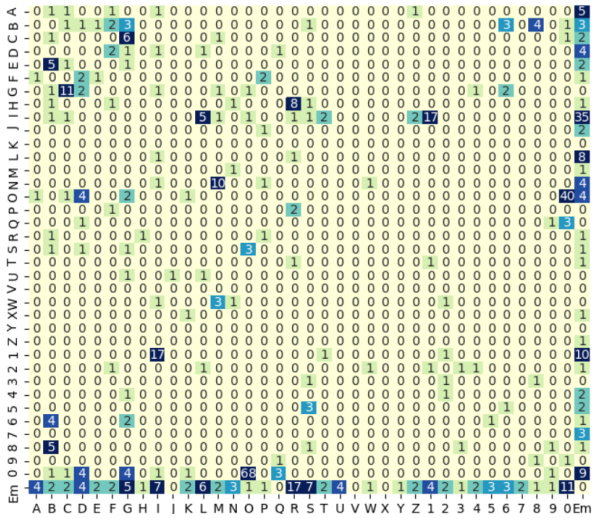

Fig. 2. Confusion matrix for characters in the license plates.

3.3.1. *Similarity-based weighting*

The first re-weighting strategy lies in assignment of higher costs for substitution of similarly-looking characters. For example, in the detected country code "GZ", the distance to the country code from the dictionary "CZ" should be lower than that to "GB", as characters "C" and "G" are visually more similar to each other than characters "Z" and "B". Such substitution costs can be estimated by constructing a training set of image-level annotated license plates, estimating of the frequency of confusion of pairs of characters in groundtruth and predicted codes. Similarly, deletion and insertion costs can be determined based on frequency of false detections and missed detections of characters. To obtain the cost matrices, we calculate the Levenshtein distance between pairs of groundtruth and predicted license plate and country codes. By tracing back the operations in each pair, we calculate the frequency of substitutions between pairs of characters as well as frequencies of their deletions and insertions. Note that a minimum Levenshtein distance for certain pair can correspond to multiple optimal sets of operations, however, we find this approach to give reasonable performance levels, while requiring zero manual labor in finding transformation operations between prediction-groundtruth pairs. The similarity costs are defined as:

$$c_s^{sim}(\mathbf{x}_i, \mathbf{y}_j) = 1 - \frac{N_s^{(\mathbf{x}_i, \mathbf{y}_j)}}{\sum_{k=1}^{36} N_s^{(\mathbf{x}_i, \mathbf{y}_k)}}, \quad (2)$$

$$c_i^{sim}(\mathbf{x}_i) = 1 - \frac{N_i^{\mathbf{x}_i}}{\sum_{k=1}^{36} N_i^{\mathbf{x}_k}}, \quad (3)$$

$$c_d^{sim}(\mathbf{x}_i) = 1 - \frac{N_d^{\mathbf{x}_i}}{\sum_{k=1}^{36} N_d^{\mathbf{x}_k}}, \quad (4)$$

where $c_i^{sim}(\mathbf{x}_i)$ and $c_d^{sim}(\mathbf{x}_i)$ are the insertion and deletion costs of character $\mathbf{x}_i$, $c_s^{sim}(\mathbf{x}_i, \mathbf{y}_j)$ is the substitution cost of character $\mathbf{x}_i$ with $\mathbf{y}_j$, $N_s^{(\mathbf{x}_i, \mathbf{y}_j)}$ is the number of confusions of groundtruth character $\mathbf{y}_j$ with detected character $\mathbf{x}_i$, $N_i^{\mathbf{x}_i}$ and $N_d^{\mathbf{x}_i}$ correspond to number of insertions and deletions of character $\mathbf{x}_i$ in the train set.

For reference and to facilitate further work on this problem, Fig.2 provides the obtained confusion matrix that can be used for formation of various costs. Here, the last column and the last row correspond to deletion and insertion frequencies, respectively.

3.3.2. *Confidence-based weighting*

Since the majority of modern ALPR systems, as well as ours, rely on detection-based character recognition, each character is generally accompanied with a confidence score produced by detector, which can serve as a metric for determining its "validity" and, hence, its substitution and deletion costs. Specifically, low-confidence predictions are more likely to correspond to mislocalized or misclassified detections, thus their substitution and deletion costs should be lower than those of high-confidence detections. Thus, given a detected character $\mathbf{x}_i$ with confidence score s_i, its confidence-weighted substitution cost with $\mathbf{y}_j$ can be defined as $\gamma(\mathbf{x}_i \rightarrow \mathbf{y}_j) = s_i c_s(\mathbf{x}_i, \mathbf{y}_j)$, where $c_s(\mathbf{x}_i, \mathbf{y}_j)$ can be set to 1 as in Levenshtein distance, or to similarity-based substitution cost defined in Eq.2 (or to virtually any other cost). Similarly, the confidence-weighted deletion cost can be formulated as $\gamma(\mathbf{x}_i \rightarrow 0) = s_i c_d(\mathbf{x}_i)$. Since the substitution and deletion costs are now scaled, scaling needs to be performed on insertion cost as well, which does not have a corresponding detection. For the insertion cost, s_i^{ins} can be defined as $\frac{1}{l_x}\sum_{i=0}^{l_x} s_i$, where l_x is the length of string $\mathbf{x}$, i.e., confidence score for insertion of any character to string $\mathbf{x}$ is equal to the mean confidence score of the string. Similarly, insertion cost is then defined as $\gamma(0 \rightarrow \mathbf{x}_i) = s_i^{ins} c_i(\mathbf{x}_i)$.

3.3.3. *Operations weighting*

Besides re-weighting based on character similarity and detection confidence score, it is worth considering that due to the nature of the OCR task, the operations of deletion, substitution, and insertion should inherently be treated differently. In challenging detection tasks, such as detection of small country code characters, missed detections are more common than false-positive detections, especially that the ratio of false-positive detections can be adjusted by thresholding the confidence score of the detector. For example, for a country code detected as 'FI', 'FIN' should be a more probable true match than 'F' or 'FO', leading to the implication that the cost of insertion operation should be lower than that of deletion or substitution. At the same time, due to the nature of detection task, the detected bounding box is more likely to be misclassified than mislocalized, i.e., cost of substitution operation should be lower than that of deletion. This can be achieved by scaling the cost of each operation, i.e., $\gamma(\mathbf{x}_i \rightarrow \mathbf{y}_j) = \alpha c_s(\mathbf{x}_i, \mathbf{y}_j)$, $\gamma(\mathbf{x}_i \rightarrow 0) = \beta c_d(\mathbf{x}_i)$, $\gamma(0 \rightarrow \mathbf{x}_i) = \delta c_i(\mathbf{x}_i)$, s.t. $\delta < \alpha < \beta$, where $c_s(\mathbf{x}_i, \mathbf{y}_j), c_d(\mathbf{x}_i)$, and $c_i(\mathbf{x}_i)$ are the substitution, deletion, and insertion costs, set to 1 as in Levenshtein distance, similarity or confidence based score as in Sections 3.3.1 and 3.3.2, respectively, or any other cost of choice. In our experiments, we set $\delta = 0.5, \alpha = 0.75, \beta = 1$, while in practice we observe that other combinations of coefficients lead to improved performance as well, as long as the inequality is satisfied.

Note that the proposed weighted costs can be combined in various ways. We will further show the advantage of their combination, and see that the best result is achieved by using combination of all

		LP	CC	ED	Sim	Conf	Oper	Sim+Conf	Sim+Conf+Oper
64x256	**Country**	80.57	73.09	80.26	85.12	80.69	83.13	85.72	**86.72**
	LP+Country			65.67	68.97	65.94	67.43	69.41	**70.07**
128x416	**Country**	84.28	78.24	85.69	86.85	85.96	87.74	87.12	**88.15**
	LP+Country			72.94	73.76	73.21	74.82	74.03	**75.06**
256x1024	**Country**	86.47	85.05	88.67	91.20	88.51	89.86	91.46	**91.99**
	LP+Country			77.90	79.58	77.82	78.84	79.84	**80.36**

3 costs, i.e., setting $\gamma(\mathbf{x}_i \rightarrow \mathbf{y}_j) = \alpha s_i c_s^{sim}(\mathbf{x}_i, \mathbf{y}_j)$, $\gamma(\mathbf{x}_i \rightarrow 0) = \beta s_i c_d^{sim}(\mathbf{x}_i)$, $\gamma(0 \rightarrow \mathbf{x}_i) = \delta s_i^{ins} c_i^{sim}(\mathbf{x}_i)$.

4. EXPERIMENTS

For training the character recognition model based on CenterNet, we use an extensive amount of artificial images. Specifically, we create rectangular images of different aspect ratios with characters of 26 distinct fonts, random colors, and font sizes. We further apply random affine transformations, motion blurring, Gaussian noise, and alter the contrast level of the image. Several examples of the artificial images can be seen in Fig.3. We generate 40570 images for training and 10750 for validation, and train the model for 300 epochs until the validation loss stops improving. Further, we annotate with bounding boxes 69 real-world images and fine-tune the detector model: we use 44 images for training and 25 for validation, and similarly train until convergence. For our experiments, we train 3 separate models with 3 different image sizes: 64x256, 128x416, 256x1024.

For country code recognition, none of the existing public datasets provides appropriate annotations. In fact, to the best of our knowledge, the only European LP dataset is Open ALPR [21] consisting of 104 images, where a large portion of license plates have missing country codes (or they are indistinguishable to the extent of impossibility of manual annotation), while the vast majority of those with country codes correspond to the same country. Therefore, this dataset is unsuitable for adequate evaluation. Instead, we collected and created image-level annotations for 1158 images from the FRANCOPLAQUE collection [22] representing 59 distinct regions: A: Austria, AL: Albania, AM: Armenia, AND: Andorra, AZ: Azerbaijan, B: Belgium, BG: Bulgaria, BIH: Bosnia and Herzegovina, BY: Belarus, CY: Cyprus, CZ: CzechRepublic, HR: Croatia, D: Germany, DK: Denmark, E: Spain, EST: Estonia, F: France, FIN: Finland, FO: Faroe Islands (Denmark), GB: United Kingdom, GBM: Isle of Man (UK), GBJ: Jersey (UK), GBG: Guernsey (UK), GBZ: Gibraltar, GE: Georgia, GR: Greece, H: Hungary, I: Italy, IRL: Ireland, IS: Iceland, KZ: Kazakhstan, KGZ: Kyrgyzstan, L: Luxembourg, LT: Lithuania, LV: Latvia, M: Malta, MD: Moldova, MNE: Montenegro, N: Norway, NL: Netherlands, MK: Macedonia, MC: Monaco, P: Portugal, PL: Poland, RO: Romania, RUS: Russia, S: Sweden, SK: Slovakia, SLO: Slovenia, SRB: Serbia, TJ: Tajikistan, TR: Turkey, UA: Ukraine, UZ: Uzbekistan. Each region is represented by at least 5 images. Additionally, from the same collection we annotated with bounding boxes 69 images which are used for fine-tuning the character detection model. After detection, we discard detections with a confidence score lower than 0.3, and additionally perform non-maximum suppression between classes with IoU of 0.6. For evaluation of our weighted edit distance metrics, we perform stratified 5-fold cross-validation on the collected dataset, selecting 80% for calculation of similarity scores and 20% for testing of our proposed metrics. The mean accuracy over all folds on the test set is reported.

Table 1 shows the obtained results for three distinct image sizes, with "LP" and "CC" corresponding to the accuracy of LP and country code prediction, respectively, and "Country" - the accuracy of the country of origin prediction using different approximate string matching metrics. "LP+Country" denotes the accuracy of the full license plate prediction, i.e, where both License Plate and country of origin were correctly established. Note that in all experiments we only considered a LP code or country code prediction to be correct when all of its characters are matching with the corresponding groundtruth. We evaluated our proposed weighted metrics along

Fig. 3. Examples of artificial images used for training the detector.

with a conventional unweighted Edit Distance. We evaluated the performance achieved by each weighting scheme independently, as well as their combinations. In Table 1, ED corresponds to conventional unweighted Edit Distance, Sim, Conf, and Oper correspond to similarity-, confidence-, and operation-based weighted ED described in Sections 3.3.1., 3.3.2, and 3.3.3, Sim+Conf corresponds to similarity cost scaled with confidence, and Sim+Conf+Oper corresponds to the combination of all three proposed metrics, i.e., similarity cost scaled with confidence and operation coefficient. As can be seen, the use of approximate string matching significantly improves the country recognition even in the unweighted edit distance case, as without approximate string matching the country recognition accuracy is essentially the country code recognition accuracy. Moreover, all proposed re-weighting schemes improve the country recognition further, even if they are used independently. Additionally, scaling the similarity cost with confidence weights further improves the performance, and the best performance on all three image sizes is achieved by a combined metric that uses all three proposed re-weighting schemes, leading to an improvement of 7 to 13 % compared to exact string matching, and an improvement of 3 to 6.5 % compared to unweighted Edit Distance.

Fig.1 shows some examples of license plates with country of origin recognition. Note that for copyright reasons the figure includes synthetic images on which various degradations were applied to simulate real environment. In that figure, "Proposed approach" corresponds to the weighted Edit Distance combining the three proposed weighting schemes. As can be seen, for the first and second license plates the models are able to recognize the countries correctly, despite the missed character in the second license plate country code. The benefit of using the proposed method can be seen from the last three license plates. The detected code "GY" has equal Edit Distance of 1 to the country codes "BY", "CY", "GB", and "GR"; and "FIHI" has equal ED of 2 to both "BIH" and "FIN", while weighting breaks this equality, hence leading to correct country identification. Notably, the detected country code "HU" has ED of 1 to "HR", but 2 to "RUS", i.e., using an unweighted metric, the true country of origin is a further match than the incorrect one. Nevertheless, the proposed method is able to account for these misdetections and the countries

of origins of given LPs are predicted correctly.

5. CONCLUSION

We introduced the problem of country code recognition from license plate images and proposed a method to solve it. We proposed several weighted Edit Distance metrics for improving country code recognition from imperfect detections, and their superiority compared to unweighted metrics was shown experimentally. The proposed weighted metrics are not limited to country code recognition, as they can be used for other detection-based OCR tasks too.

6. REFERENCES

[1] G. Navarro, "A guided tour to approximate string matching," *ACM computing surveys (CSUR)*, vol. 33, no. 1, pp. 31–88, 2001.

[2] S. Silva and C. Jung, "Real-time brazilian license plate detection and recognition using deep convolutional neural networks," in *SIBGRAPI Conference on Graphics, Patterns and Images*. IEEE, 2017, pp. 55–62.

[3] Y. Lee, J. Jun, Y. Hong, and M. Jeon, "Practical license plate recognition in unconstrained surveillance systems with adversarial super-resolution," *arXiv preprint arXiv:1910.04324*, 2019.

[4] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7263–7271.

[5] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[6] R. Laroca, E. Severo, L. Zanlorensi, L. Oliveira, G. Gonçalves, W. Schwartz, and D. Menotti, "A robust real-time automatic license plate recognition based on the yolo detector," in *International Joint Conference On Neural Networks*. IEEE, 2018, pp. 1–10.

[7] Y. Lee, J. Lee, H. Ahn, and M. Jeon, "Snider: Single noisy image denoising and rectification for improving license plate recognition," in *IEEE International Conference on Computer Vision Workshops*, 2019.

[8] S. M. Silva and R. C. Jung, "License plate detection and recognition in unconstrained scenarios," in *European Conference on Computer Vision*, 2018, pp. 580–596.

[9] R. Laroca, L. Zanlorensi, G. Gonçalves, E. Todt, W. Schwartz, and D. Menotti, "An efficient and layout-independent automatic license plate recognition system based on the yolo detector," *arXiv preprint arXiv:1909.01754*, 2019.

[10] V. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet Physics Doklady*, 1966, vol. 10, pp. 707–710.

[11] F. Damerau, "A technique for computer detection and correction of spelling errors," *Communications of the ACM*, vol. 7, no. 3, pp. 171–176, 1964.

[12] F. Oliveira-Neto, L. Han, and M. Jeong, "Online license plate matching procedures using license-plate recognition machines and new weighted edit distance," *Transportation Research Part C: Emerging Technologies*, vol. 21, no. 1, pp. 306–320, 2012.

[13] A. Samuelsson, "Weighting edit distance to improve spelling correction in music entity search," M.S. thesis, KTH Royal Institute of Technology, 2017.

[14] L. Fontan, I. Ferrané, J. Farinas, J. Pinquier, and X. Aumont, "Using phonologically weighted levenshtein distances for the prediction of microscopic intelligibility," *Interspeech*, pp. 650–654, 2016.

[15] K. Majorek, S. Dunin-Horkawicz, K. Steczkiewicz, A. Muszewska, M. Nowotny, K.f Ginalski, and J. Bujnicki, "The rnase h-like superfamily: New members, comparative structural analysis and evolutionary classification," *Nucleic Acids Research*, vol. 42, no. 7, pp. 4160–4179, 2014.

[16] N. Watcharapinchai and S. Rujikietgumjorn, "Approximate license plate string matching for vehicle re-identification," in *IEEE International Conference on Advanced Video and Signal Based Surveillance*. IEEE, 2017, pp. 1–6.

[17] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.

[18] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," Tech. Rep., Stanford, 2006.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[20] D. Müllner, "Modern hierarchical, agglomerative clustering algorithms," *arXiv preprint arXiv:1109.2378*, 2011.

[21] "Open alpr dataset," https://github.com/openalpr/benchmarks, Accessed: 2020-11-12.

[22] "Francoplaque license plates collection," http://plaque.free.fr/index-english.html, Accessed: 2020-11-12.