

Vehicle Plate Matching using License Plate Recognition based on Modified Levenshtein Edit Distance

Shamaila Khan
Assist. Prof. (CSE Dept)
ASCE, Bhopal

Sarfraj Ali
(PG Scholer)
ASCE, Bhopal

ABSTRACT

Vehicle License plate recognition (LPR) method is a full-grown so far deficient approach used for computerized toll group and rapidity enforcement. Recently, an sophisticated matching approach that combines Bayesian likelihood and Levenshtein text-mining method was planned to improve the exactness of computerized vehicle license plate matching. The key module of this technique is what we known as Precision-Recall curve, which contains the conditional probabilities of observing one character at one node for a given observed character at an additional station. Therefore, the evaluation of the performance constraint relies on the by hand extracted position truth of a large number of plates, which is an unwieldy and deadly process. To beat this negative aspect, in this cram, we propose an inventive novel LPM-MLED (License Plate Matching - Modified Levenshtein Edit Distance) method that removes the need for extracting ground truth by hand. The propose method perform well in the exactness in plate matching, in comparison with those generated from the meticulous manual method. Furthermore, this method outperforms their manual counterparts in plummeting false matching rates. The computational LPM-MLED technique is also cheaper and easier to implement and continues to improve and correct itself over time.

Keywords

LPM-MLED (License Plate Matching- Modified Levenshtein Edit Distance), text mining, vehicle tracking.

1. INTRODUCTION

License-plate recognition technology was originally developed to read license plate character on moving vehicles. The process of capturing a plate image and recognizing the characters involves vehicle detection, image processing, and optical character recognition. By comparing the imperfectly read plate against another such plate, one may still be able to render reasonable judgment in terms of whether the two plates are actually a match. For instance, if two strings (sequence of characters) differ from each other by only one character, they may well have originated from the same plate. Therefore, a measure of similarity between two strings can be established to indicate the likelihood of a match. Measuring similarity of strings is a well-known problem in computer science which has applications in many fields such as computational biology, text processing, optical character recognition, image and signal processing, error correction, information retrieval, pattern recognition, and pattern matching in large databases. Consider two strings X and Y over a finite alphabet whose lengths are m and n respectively with $m \leq n$. We consider sequences of weighted *edit operations* (insertions, deletions, and substitutions of characters), by means of which X is transformed into Y [4]. If we call each such sequence an *edit sequence*, then the ordinary (conventional) *edit distance problem* (ED) seeks for an edit sequence with minimum total

weight over all sequences. The *edit distance* between X and Y is defined as the weight of such a sequence. Although the edit distance is a useful measure for similarity of two strings, for some applications the lengths of the strings compared need to be taken into account. There are some common features affecting on performance of existing LPR algorithms including:

- Lighting conditions such as cloudy weather, night working hours, reflecting sunlight or car's tail light or front light.
- Complex background which effects on speed of detecting the real region of plate.
- Damaged or dirty license plates that causes to gain fault identification numbers.
- Varying view angle and efficient distance between camera and moving car.

The problem identification of this work is as follows:

- The probability of true positive measure is low in traditional LPM.
- The probability of false positive measure is high in traditional LPM.
- The model accuracy of traditional LPM is low.
- The distance rate is high in traditional Edit Distance

In Following Chapter 2 gives literature reviews of the different method of license plate matching, this entire scheme entitled with their authors name and respective title. Chapter 3 describes the process of proposed method with algorithm and corresponding diagrams. Chapter 4 describes the implementation details and information about used data set. The implementation details are also explained out in terms of the algorithm process. Chapter 5 describes the result and analysis of the proposed work in terms of image quality metrics. Chapter 6 describes the conclusion of the proposed work and also describes the future work. Chapter 8 specifies the references of research paper, which use the corresponding details in our work.

2. LITERATURE SURVEY

License plate recognition (LPR) technology is a mature yet imperfect technology used for automated toll collection and speed enforcement. Existing methods for improving the matching of plates recognized by LPR units rely on intensive manual data reduction, such that the misread plates are manually entered into the system [1]. To compensate for LPR misreading problem, we propose a new weight function based on a probability model to match the observed outcomes of a dual LPR setup [2]. Present LPR systems are not perfect and can fail to read 1/5 to 1/2 of the characters on license plates, depending on various factors. Fortunately, even when LPR

fails to read all of the characters on license plates, it is still able to read most of the characters with an appreciable degree of accuracy [3]. Levenshtein distance is a simple metric which can be an effective string approximation tool. After observing the effectiveness of this method, an improvement has been made to this method by grouping some similar looking alphabets and reducing the weighted difference among members of the same group [4]. Text Mining is an important step of knowledge discovery process. Text mining extracts hidden information from not-structured to semi-structured data. Text mining is the discovery by automatically extracting information from different written resources and also by computer for extracting new, previously unknown information[5]. Given two strings X and Y over a finite alphabet, the normalized edit distance between X and Y, $d(X, Y)$ is defined as the minimum of $W(P)/L(P)$, where P is an editing path between X and Y, $W(P)$ is the sum of the weights of the elementary edit operations of P, and $L(P)$ is the number of these operations (length of P). In this paper, it is shown that in general, $d(X, Y)$ cannot be computed by first obtaining the conventional (un-normalized) edit distance between X and Y and then normalizing this value by the length of the corresponding editing path [6].

3. PROPOSED METHODOLOGY

The algorithm for proposed methodology LPM-MLED (License Plate Matching with Modified Levenshtein Edit Distance) is as follows:

$[DM] = LPM_MLED(Source_Image, Target_Image)$

Step 1: Read license plate of vehicle at two different stations namely as Source_Image, Target_Image.

Step 2: These images has RGB form, it is convert into gray scale form.

Step 3: Apply Morphological operation and binarization step for scan the shape of character.

Step 4: Now apply vertical edge detection approach for evaluate the orientation of character image in license plate.

Step 5: Now Step 3 and Step 4 are performs iteratively for obtain actual character shape in alphanumeric form which is known as recursive learning approach.

Step 6: Find the actual character location in license plate.

Step 7: Perform image enhancement and de-noising for improve the quality of character image of license plate.

Step 8: Apply image segmentation for convert digital image in to multiple segment of character and form the character for calculation purpose.

Step 9: Now obtain alphanumeric string S and T from Source_Image and Target_Image, respectively.

Step 10: Now perform Modified Levenshtein Edit Distance for evaluating distance metric.

Step 11: Consider weight matrix for 38 alphanumeric characters with set

$\Omega^* = \{', - , 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z\}$

Step 12: The row of weight matrix is consider as S string and column of weight matrix is consider as T string. Weight value can be calculate as

$$W_{ij} := \log(S(i)+T(j))$$

The diagonal values are labeled as zero.

Step 13: Edit operations are consider as

$a \rightarrow b, \forall a, b \in \Omega^*$. There are three types of elementary edit operations namely as insertions, substitutions, and deletions, which take the forms $\lambda \rightarrow b$, $a \rightarrow b$, and $a \rightarrow \lambda$, respectively. λ is a null character.

Step 14: Initialize weight value (w) and number of transformation (count) as

$$w := 0.0, \quad \text{count} := 0$$

Step 15: Now perform following five conditions for evaluating modified levenshtein edit distance metric. The $\text{weightvalue}(\text{char})$ is function for obtain index of character from set Ω^* .

(a) If String S is null and String T has character set, then perform insertion operation as follows.

Loop I := 1 to length(T)

Index := $\text{weightvalue}(T(I))$

w := w + W(1, Index)

increment count

End

(b) If String S has character set and String T has null, and then performs deletion operation as follows

Deletion:

Loop I := 1 to length(S)

Index := $\text{weightvalue}(S(I))$

w := w + W(Index, 1)

increment count

End

(c) If String S has character set and String T has character set. The size of String S is less than the size of String T and then performs insertion, deletion and substitutions operation as follows

Loop I := 1 to length(S)

Condition $S(I) == T(I)$

w := w + 0.0

If both string S and T are not null, then perform following operation

Index1 := $\text{weightvalue}(S(I))$

Index2 := $\text{weightvalue}(T(I))$

w := w + W(Index1, 1),

increment count

w := w + W(1, Index2)

increment count

If string S has null character and string T has not null character, then perform insertion operation in following way

Index=weightvalue(T(I))

w := w + W(1,Index)

increment count

End

(d) If String S has character set and String T has character set. The size of String S is greater than the size of String T and then performs insertion, deletion and substitutions operation as follows

Loop I := 1 to length(S)

Condition S(I) == T(I), w := w + 0.0

If both string S and T are not null, then perform following operation

Index1 := weightvalue(S(I))

Index2 := weightvalue(T(I))

w := w + W(Index1,1)

increment count

w := w + W(1,Index2)

increment count

If string S has not null character and string T has null character, then perform deletion operation in following way

Index := weightvalue(S(I))

w := w + W(Index,1)

increment count

End

(e) If String S has character set and String T has character set. The size of String S is equal to the size of String T and then performs substitutions operation as follows

Loop I := 1 to length(S)

Condition S(I) == T(I)

w := w + 0.0

If both string S and T are not null, then perform following operation

Index1 := weightvalue(S(I))

Index2 := weightvalue(T(I))

w := w + W(Index1,1)

increment count

w := w + W(1,Index2)

increment count

End

Step 14: Now obtain Modified Levenshtein Edit Distance path from S to T.

$$\bar{w} = \frac{w}{\text{count} + \log(\text{count})}$$

$d(S,T) = \min\{\bar{w}(P) \mid P \text{ is an editing path between } S \text{ and } T\}$.

Step 15: Finally obtain Modified Levenshtein Edit Distance $d(S,T)$.

4. EXPERIMENTAL STEP:

Experiments are performed on various vehicle license plates in different environment.

SN	License Plate Number	Images of License Plate
1	PA PL8S	
2	AAZ1297	
3	3HUA172	
4	GHG-5269	
5	KZF 827	
6	BFEJ-681	
7	756 JWB	
8	697 BBJ	

Figure 1: License Plate Data Set

The transport department of various countries like US, UK, Europe and Asia has different types of license plate[18]. We have to consider that 46 vehicle license plate from different country. These plates are observed in different environment and different location. We can capture license plate with different focus points and exposure settings. The testing license plates are numbered from 1 to 46. Each pair of images has different size and aspect ratio. The experimental process start with license plate recognition at two different station G and H. Station G is consider source station where plate is recognized first and station H is taken as target station where plate is recognized in next phase. There are two cases are encounter during LPR install at two different stations [8].

Case 1: If both station are correctly read license plate. At station H, the license plate “AFR-420” is recognize as same as “AFR-420”. When the vehicle is move to another toll station H then above license plate is consider same as “AFR-420”.



Figure 2: At station H, the license plate “AFR-420” is recognize as same as “AFR-420”

Case 2: If both station are incorrectly read license plate. At station H, the license plate “K0 3692 B” is recognizes as “K0 3692 B”. When the vehicle is move to another toll station H then above license plate is consider different as “K0 5692 B”.

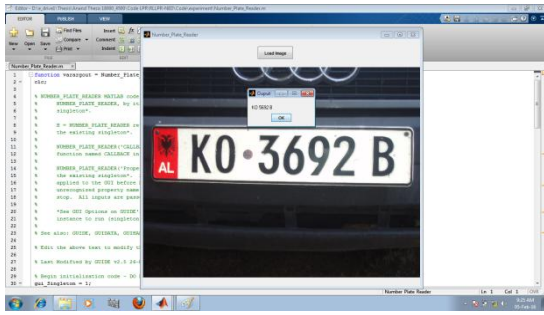


Figure 3: At station H, the license plate “K0 3692 B” is recognize different as “K0 5692 B”

When string is observed at two different stations, then strings are matching through Modified Levenshtein Edit Distance (MLED). There are two cases exists for calculate the distance between strings. The experimental process of LPM-MLED is as follows:

Case 1: If source and target string are same.

Case 2: If source and target string are different.

5. RESULT ANALYSIS

The analysis of the existing work (LPM-LED) and the proposed work (LPM-MLED) on different parameters are given in following tables. The analysis can be performs in two cases.

Case 1: If both station are correctly read license plate.

Case 2: If station G is correctly read license plate but station H is incorrectly read license plate.

Model Accuracy of proposed method (LPM-MLED) is greater than previous method (LPM-LED). It means that proposed method is more consistent approach for matching license plate than existing method (LPM-LED). The F1-Score of proposed method (LPM-MLED) is greater than previous method (LPM-LED). It means that proposed method (LPM-MLED) is suitable for retrieve more relevant character from license plate as compare than existing method (LPM-LED) which are selected from license plate character. The Distance Metric is same for (LPM-MLED) and (LPM-LED) because characters of license plate are same at both stations. Pie chart analysis comparisons for Model Accuracy and F1-Score can be demonstrated as follows.

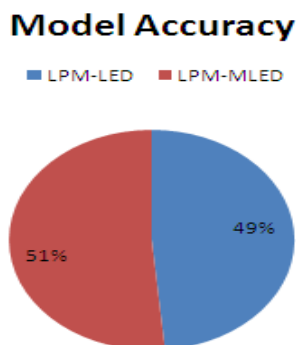


Figure 5: Compare Model Accuracy between LPM-LED and LPM-MLED for Correct Matching

Model Accuracy of proposed method (LPM-MLED) is greater than previous method (LPM-LED). It means that proposed method is more consistent approach for matching license plate than existing method (LPM-LED). Graph chart analysis

comparisons for Model Accuracy can be demonstrated as follows.

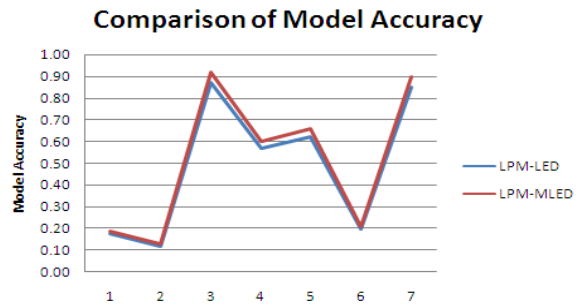


Figure 4: Model Accuracy between LPM-LED and LPM-MLED for Incorrect Matching

The F1-Score of proposed method (LPM-MLED) is greater than previous method (LPM-LED). It means that proposed method (LPM-MLED) is suitable for retrieve more relevant character from license plate as compare than existing method (LPM-LED) which are selected from license plate character. Graph chart analysis comparisons for F1-Score can be demonstrated as follows.

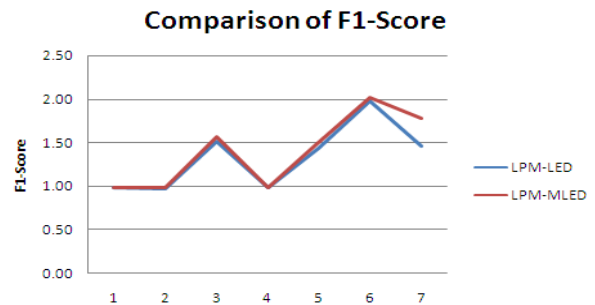


Figure 6: F1-Score between LPM-LED and LPM-MLED for Incorrect Matching

The Distance Metric value is same for (LPM-MLED) and (LPM-LED) because characters of license plate are same at both stations.

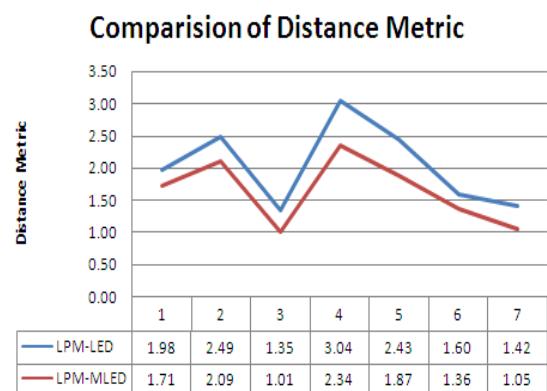


Figure 7: Distance Metric between LPM-LED and LPM-MLED for Incorrect Matching

6. CONCLUSIONS AND FUTURE WORK

Without tinkering with the LPR hardware and the embedded video-image-processing algorithm, our proposed procedure LPM-MLED improves the license plate matching rate significantly. The precision-recall parameter explained that proposed method shows the effective result as compare than

existing method. License plate recognition can be learned automatically via the recursive learning procedure, which used to be prohibitively expensive to acquire and maintain for a large network with many LPR units installed, can be attained in a matter of days. As the algorithm continues to run in the background, the association matrices are maintained, updated, and improved over time. When new license plate designs are introduced to the system, which can be often as there are over 3000 different license plate designs in the U.S., the algorithm automatically learns to match them quickly. The algorithm can be applied in a system with LPR units from mixed vendors. Since the algorithm is entirely post processing based, as long as an LPR unit provides a plate string for comparison purposes, it can be used as input. Even when different LPR hardware producers have various levels of recognition performance, as long as they can recognize a reasonable amount of characters, the matching performance will be satisfactory.

The proposed method highly relies on the extension of Levenshtein ED that accepts the symbol-based edit operations. We can use SURF, Fast SIFT, CHoG method for find the feature descriptor of an image after extract image part from license plate. We adapted the similarity measure for LPR application by estimating value into the weight functions in order to find the most likely alignment between the pair of strings. To the best of our knowledge, there is no similar application in the literature. Future research can also explore other formulations for the weight functions and other models to associate the LPR-reported strings (e.g., transduction model), as well as classification methods.

7. REFERENCES

- [1] Francisco Moraes Oliveira-Neto, Lee D. Han, and Myong Kee Jeong, "An Online Self-Learning Algorithm for License Plate Matching", *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*, VOL. 14, NO. 4, DECEMBER 2013.
- [2] Rashmi Agrawal, Mridula Batra, "A Detailed Study on Text Mining Techniques", *International Journal of Soft Computing and Engineering (IJSCE)*, ISSN: 2231-2307, Volume-2, Issue-6, January 2013.
- [3] Sonali Vijay Gaikwad, Archana Chaugule and Pramod Patil, "Text Mining Methods and Techniques", *International Journal of Computer Applications (0975 – 8887)* Volume 85 – No 17, January 2014.
- [4] Rishin Haldar and Debajyoti Mukhopadhyay, "Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach", *Journal of Web Intelligence & Distributed Computing Research Lab*, jan 2012.
- [5] Andrcs Marzal and Enrique Vidal, "Computation of Normalized Edit Distance and Applications", *IEEE TRANSACTIONS ON PA'ITERN ANALYSIS AND MACHINE INTELLIGENCE*, VOL. 15, NO. 9, SEPTEMBER 1993.
- [6] Abhik Das, Praneeth Netrapalli, Sujay Sanghavi and Sriram Vishwanath, "Learning Markov Graphs Up To Edit Distance", *Journal of Department of ECE, The University of Texas at Austin, USA*.
- [7] Eric Sven Ristad and Peter N. Yianilos, "Learning String Edit Distance", *Research Report of Dept of Computer Science, Princeton University*, Oct 1997.
- [8] Abdullah N. Arslan and Omer Egecioglu, "An Efficient Uniform-Cost Normalized Edit Distance Algorithm", *Journal of Department of Computer Science University of California, Santa Barbara*.
- [9] Francisco Moraes Oliveira-Neta, Lee D. Han, "Online License Plate Matching Procedure using License-Plate Recognition Machines and New Weighted Edit Distance", *Elsevier Journal*, Nov 2011.