The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM)
ISSN: 2602-3199

The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM), 2024

Volume 32, Pages 295-303

IConTES 2024: International Conference on Technology, Engineering and Science

Significant Improvement in License Plate Recognition through Image Deduplication

Djamel Berrabah
EEDIS, University Djillali Liabes

Yacine Gafour
LIM, University Ibn Khaldoun

Abstract The detection of license plates, or Automatic Number Plate Recognition (ANPR), is crucial for applications in parking management, vehicle tracking, and security. However, the efficiency of ANPR systems is often compromised by large datasets containing numerous similar or duplicate images, leading to increased storage costs and slowed processing times. This research proposes an innovative approach that combines perceptual hashing and locality-sensitive hashing (LSH) to enhance the detection of redundant images. Perceptual hashing generates unique visual fingerprints for images, facilitating efficient duplicate identification, while LSH groups similar images to reduce false positives. Additionally, Optical Character Recognition (OCR) is applied to the image pairs identified by LSH to extract license plates and verify vehicle identity. By integrating these techniques, the proposed method effectively mitigates redundancy, optimizing storage and improving the performance of ANPR systems for accurate real-world recognition.

Keywords: Automatic number plate recognition (ANPR), Perceptual hashing, Locality-sensitive hashing (LSH), Optical character recognition (OCR), Duplicate image detection.

Introduction

The application of ANPR (Automatic Number Plate Recognition) systems has gained significant traction in recent years, becoming integral to various domains such as traffic management, law enforcement, parking control, and toll collection. These systems rely on advanced image processing techniques to accurately recognize license plates from vehicles in real time. However, as the volume of data generated by ANPR systems increases, managing large datasets becomes more challenging. One critical issue is the presence of redundant images, which can result in increased storage requirements and slower image processing times, ultimately affecting both efficiency and accuracy.

Image deduplication techniques based on perceptual hashing (A-Hash, D-Hash, P-Hash, etc.) offer an effective solution. These algorithms generate a visual fingerprint for each image by simplifying its key characteristics, enabling rapid duplicate detection. In the ANPR context, these methods identify nearly identical license plate images, even under variations in lighting or viewing angle. For example, D-Hash is particularly effective for comparing differences between adjacent pixels, while P-Hash extracts global shapes and contrasts, detecting similar images even with slight visual modifications.

Recent research has focused on optimizing image databases in ANPR systems. For instance, hybrid techniques combining perceptual hashing and machine learning have been proposed to automatically detect and remove duplicates, improving dataset quality. Other studies explore specialized convolutional neural networks (CNNs) to classify images based on similarity, streamlining the filtering process. Additionally, wavelet-based methods have proven effective in extracting multi-resolution features, enabling detection of subtle variations in license

- Selection and peer-review under responsibility of the Organizing Committee of the Conference

plate images. These advanced techniques reduce redundancy, optimizing storage and computational resources while enhancing model performance.

In this context, our proposal involves applying perceptual hashing followed by locality-sensitive hashing (LSH) to improve duplicate filtering. LSH refines this process by grouping similar images more granularly, reducing false positives. Finally, an OCR (Optical Character Recognition) analysis is performed on image pairs identified by LSH to verify whether the vehicles' license plates correspond to the same car. By combining perceptual hashing, LSH, and OCR, this approach effectively ensures deduplication while maintaining accuracy, optimizing both storage and the performance of ANPR models for reliable real-world recognition.

State of the Art in Image Deduplication Techniques

Image deduplication plays a crucial role in various applications, particularly facilitating optimized storage management and efficient content retrieval. This section examines modern approaches to image deduplication and explores their potential to enhance the performance of Automatic Number Plate Recognition (ANPR).

Combination of Descriptors for Optimized Feature Extraction

Velmurugan and Baboo (2011) combined SURF features with color moments for content-based image retrieval (CBIR), demonstrating that using combined feature extraction techniques can increase the accuracy of image retrieval. Although this work targets general image retrieval, the approach could inspire a similar strategy for identifying nearly duplicate license plate images in an ANPR context. Li et al. (2014) proposes an enhancement of the classic SURF algorithm by combining it with the DAISY descriptor to strengthen its rotation invariance. This model, designed to overcome SURF's limitations in handling image rotations, offers a more robust matching. While Li does not directly address deduplication, his approach highlights the importance of solid feature extraction for reliable matches—a valuable asset for identifying similar license plate images under varying shooting conditions.

Fast Indexing for Nearly Duplicate Image Detection

Lei et al. (2014) present an innovative indexing structure, Uniform Randomized Tree Groups (URT), which enables rapid detection of nearly duplicate images. This technique is based on two main concepts: uniformity (grouping images with similar features in the same scale subsets) and the use of feature subspaces and random projections to enhance the flexibility and robustness of the indexing structure. While this approach specifically targets near-duplicates, the search space reduction and robustness it provides are directly applicable to deduplication needs in ANPR.

Locality-Sensitive Hashing and Part-Based Representations

Ke et al. (2004) explore the detection of nearly duplicate images through a part-based representation combined with locality-sensitive hashing (LSH). Although their work focuses on copyright infringement detection, their strategy of local feature representation aligns with the needs for robust feature extraction in the ANPR context. By adapting LSH to index these license plate-specific features, it is possible to efficiently manage nearly duplicate images, thereby enhancing ANPR system performance.

PCA and Clusturing

Berrabah and Gafour (2023) explore the combined use of Principal Component Analysis (PCA) and K-means clustering for dataset reduction, enhancing efficiency while maintaining data quality. PCA effectively reduces dimensionality, preserving essential information, while K-means organizes the dataset, minimizing redundancy. The choice of parameters and feature selection is crucial for achieving optimal results. The experiments demonstrate the potential for eliminating redundant data, though success may vary based on dataset characteristics and algorithms used. Additionally, the challenge of class imbalance is highlighted, as it can lead to biased model predictions. The study suggests that while K-means is effective for identifying duplicate images, alternative methods like perceptual hashing may also be appropriate, emphasizing the importance of selecting the right feature representation and similarity metrics for optimal results.

Visual Bag of Words Models and Clustering

Li and Feng (2013) developed a method for detecting nearly duplicate images that combines the visual bag of words (BoW) model with locality-sensitive hashing (LSH) for fast and robust image comparison. The BoW model, which groups local features extracted via SIFT descriptors, facilitates image comparison based on visual vocabulary rather than pixel values. This model, coupled with the K-means algorithm for clustering similar features, provides robustness against variations in conditions such as lighting changes or occlusions, which are common in license plate images.

Efficient and Memory-Friendly Deduplication

Nian et al. (2016) designed an approach for deduplicating nearly duplicate images called Local Binary Representation (LBR). This binary coding scheme reduces memory requirements and computational costs by representing local regions as binary vectors of statistical texture histograms. While this method primarily targets near-duplicates in the context of video and online multimedia, it could potentially be adapted to identify nearly duplicate license plate images within ANPR systems.

Method

Automatic Number Plate Recognition (ANPR) systems are essential in areas such as traffic surveillance and security. Their effectiveness relies on accurately detecting and recognizing license plates under various conditions; however, achieving high precision is often challenging due to factors such as lighting, weather conditions, and the complexity of plate designs. To ensure optimal performance, it is crucial to train these systems on datasets that faithfully reflect local plate characteristics.

This study aims to enhance the accuracy of license plate detection through data deduplication. We propose an approach that integrates a deduplication technique to eliminate redundant images while preserving data diversity. Indeed, removing duplicates can strengthen the quality of training data, which is vital for the effectiveness of machine learning models.

Our methodology relies on three algorithms: perceptual hashing, Locality Sensitive Hashing (LSH), and Optical Character Recognition (OCR). We begin by hashing the images using a perceptual hashing function, then employ LSH to efficiently identify duplicates. By extracting and comparing the license plates from similar image pairs, we can determine which images are indeed duplicates.

Perceptual Hashing

Perceptual hashing refers to techniques that generate a unique fingerprint (hash) for images based on their visual content, rather than their binary data (Hamadouch et al., 2021). These hashes are designed to ensure that two images that are visually similar produce hashes that are highly correlated, while images that are distinctly different yield uncorrelated hashes. An effective perceptual hashing technique can recognize that one image has been derived from another—such as through resizing, compression, or minor alterations—while still maintaining a perceptual resemblance. This property is particularly useful in applications such as image retrieval, copyright enforcement, and duplicate detection, where identifying similar images despite variations is essential. Common algorithms used in perceptual hashing include Average Hash (A-Hash), Difference Hash (D-Hash), and Discrete Cosine Transform Hash (P-Hash), each with its own method of capturing and comparing visual features (Hamadouch et al., 2021).

Steps in Generating Perceptual Hashes

Perceptual hashing algorithms leverage the visual characteristics of images to create unique hashes. The primary goal is to produce hashes that remain unchanged or only slightly altered when modifications preserving the content are made to the image. For two images X and Y, their corresponding perceptual hashes are defined as $h_X=H(X)$ and $h_Y=H(Y)$. A similarity metric $D(h_X,h_Y)$ is then employed, with a threshold t determined empirically. If $D(h_X,h_Y) < t$, this indicates that X and Y are copies of the same image, altered only minimally. The three main steps involved in perceptual hashing are: 1) Image Preprocessing, which prepares the image for analysis; 2) Extraction of Perceptual Features, where relevant visual characteristics are identified; and 3) Quantification or Compression, which generates the final hash string. Various algorithms, such as Average Hash (A-Hash), Difference Hash (D-Hash), and Discrete Cosine Transform Hash (P-Hash), differ in how they extract perceptual features from the image (Samanta, & Jain, 2021).

Image Preprocessing

The preprocessing phase in perceptual hashing algorithms prepares the image to facilitate feature extraction. This step reduces the amount of data to be processed, thereby speeding up the entire process. Various operations are typically carried out at this stage, such as resizing, color transformation, normalization, filtering, and histogram equalization. For instance, an image may be converted to a specific color model, such as YCbCr (luminance and chrominance), HSV (hue, saturation, value), or grayscale. Noise, often introduced by capture conditions, can be reduced through the application of Gaussian filters. Additionally, histogram equalization is sometimes used to enhance contrast by redistributing intensity values. Finally, images may be resized to a standard size, ensuring consistency in subsequent processing.(Samanta & Jain, 2021).

Extraction of Perceptual Features

Feature extraction is a crucial step in the perceptual hashing process, as it determines how effectively the algorithm can capture the essential visual characteristics of an image. The objective is to identify robust features that remain invariant to common image transformations such as resizing, rotation, and compression. Effective feature extraction can be achieved using two main types of techniques: (a) frequency domain transformations, such as Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), and Fourier-Mellin Transform, and (b) dimensionality reduction techniques, including Principal Component Analysis (PCA), Non-negative Matrix Factorization (NMF), and Singular Value Decomposition (SVD). (Samanta & Jain, 2021).

These features are then used to create a compact representation of the image, which serves as its perceptual hash. By focusing on the most significant aspects of the image, the hashing algorithm can produce hashes that are similar for visually similar images while remaining distinct for those that are different. This allows the system to recognize modified versions of an image effectively, even when the underlying content has been altered.

Quantification or Compression

The numerical values representing the features of an image can be quantified to generate a fixed-size hash, providing a compact and relatively unique representation of the image. This quantification process often relies on the statistical properties of the extracted features. For example, in algorithms utilizing the Discrete Cosine Transform (DCT), the DCT coefficients can be quantified by comparing each coefficient to references such as the median or mean of the coefficients. It is crucial for this quantification to be carefully calibrated to preserve essential visual information, ensuring that the hashes remain significant for similar images while remaining distinct for those that differ. This plays a central role in the effectiveness of perceptual hashing, facilitating the detection and recognition of modified images.

The perceptual hashing algorithm has consistently demonstrated its effectiveness in various applications, such as image retrieval and duplicate detection. However, it involves a sequential comparison of image features, which can become computationally expensive when dealing with large datasets. This increased processing time poses challenges in efficiently managing and analyzing extensive image collections. To address this issue, we propose applying the Locality Sensitive Hashing (LSH) algorithm instead of relying on sequential comparisons of image hashes. By leveraging LSH, we can efficiently group similar images together, thereby reducing the overall number of comparisons required. This approach not only mitigates the time cost associated with large datasets but also enhances the scalability of the perceptual hashing process.

Locality-Sensitive Hashing

After perceptual hashing is performed, locality-sensitive hashing (LSH) is applied to efficiently and scalably identify similar or nearly similar images. LSH techniques utilize specialized hashing functions that group similar images into the same compartments, increasing the likelihood that similar items will be found together. While LSH does not guarantee that all data in a compartment are similar, it significantly improves the chances of clustering similar items, making it a valuable approach for image retrieval, duplicate detection, and similar tasks.

Locality-sensitive hash functions are defined as follows: A family of functions H = {h : S → U} is an LSH family for any two points p, q ∈ S, if for any function h in H, the following conditions are satisfied:

If d(p, q) ≤ r1, then PrH (h(p) = h(q)) ≥ P1.

If d(p, q) ≥ r1, then PrH (h(p) = h(q)) ≤ P1.

Here d(p, q) denotes the distance between p and q, PrH () indicates the probability, r1 and r2 are constants for distribution. (r1 < r2), and P1 and P2 are constants for the probabilities (P1 > P2). A family H of functions satisfying the above conditions is called locality-sensitive with respect to (r1, r2, P1, P2).(Lee, 2012; Slaney & Casey, 2004)

The hashing functions by Indyk et al. are defined as follows: First, data points are encoded in binary codes within Hamming space. To define a hashing function $h()$, a fixed number of positions are randomly sampled with replacement from the set of positions, and the function value for a data point is constructed by concatenating the binary values at the selected positions. When the number of selected positions is d, the bucket size for a hashing function becomes 2^{d}. Several hashing functions H are constructed in the same way. Each data point in the database is hashed by hashing functions into buckets. When a query d_q is given, it is hashed by each hashing function h into the corresponding bucket. The data points from the buckets to which the hashing functions map the query become the candidates with which the query is compared to determine if they are neighbors.

How LSH Works in Practice

Preprocessing: From Image to Signature

In this preprocessing step, the goal is to compress images into signatures, making the task of finding duplicates easier. If two images are similar, their signatures should also be similar. For a given hashing function and for a given k, the generated signature is of length k2, For k=16, the generated signature looks like this:

1	1	1	0	1	1	1	0	1	1	0	0	0	0	0	0
1	1	1	0	1	1	0	0	1	1	1	0	0	0	0	0
1	1	1	1	1	0	0	0	0	1	1	0	0	0	0	0
1	1	1	1	1	0	0	1	0	1	1	0	0	0	0	0
1	1	1	1	1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	1	1	0	1	1	0	0	1	1	0	0	0	0
1	1	0	1	1	0	1	1	0	1	1	1	0	0	0	0
1	1	0	1	1	0	0	1	0	1	1	1	0	0	0	0
1	0	1	1	1	0	1	1	0	0	0	1	1	0	0	0
1	0	1	1	0	1	1	1	1	0	0	0	1	1	0	0
1	0	1	0	0	1	1	1	1	0	0	1	1	1	1	1
0	1	1	1	0	1	1	1	0	0	0	1	1	0	1	1
1	0	1	0	0	1	1	1	0	0	0	1	0	1	0	1
1	0	1	1	0	0	1	1	1	0	0	0	0	0	1	0
0	0	1	1	0	1	1	0	1	1	0	1	0	0	1	0
1	1	1	0	0	1	0	0	1	1	0	0	0	1	0	0

Generating Candidate Pairs

Now, instead of comparing the signatures with each other, each signature is divided into b bands of r rows (bits). This means, of course, that b × r = k2 with k2 being the length of the signature. We will then apply b hashing functions 1 to each band to all substrings of signatures (pieces of r bits) from band b.

If any of these signature substrings end up in the same bucket for any of the hashing functions, we will consider these signatures and their associated files as candidate pairs (quasi-duplicates). We can influence the sensitivity of candidate pair detection by adjusting the parameters b and r. If we take fewer but wider bands, it becomes less "easy" for the signature substrings (bands) to hash into the same bucket, and vice versa (Lee, 2012).

Since this process is likely to create false positives, a post-processing step is necessary where we actually compare the signatures of our candidate pairs with each other. This allows us to check whether they are indeed quasi-duplicates and to what extent. We do this by calculating the Hamming distance between two signatures, that is, the number of bits where the two signatures differ. If this number divided by the signature length (Hamming Distance / k2) is less than a self-defined threshold t, we can consider them as quasi-duplicates. For example, one might choose t equal to 0.8 to consider images that are 80% similar as quasi-duplicates.

The problem lies in defining the threshold, which sometimes leads to the removal of images that have unique existence in the dataset, resulting in a loss of information regarding the vehicle's license plate in question. To remedy this issue, we thought about using OCR to ensure whether the pair of images represents the same vehicle or two different vehicles.

OCR (Optical Character Recognition)

Optical Character Recognition (OCR) is a technology that converts text images (such as scanned documents, photos of text, or PDFs) into editable digital text. It allows a computer to "read" printed, handwritten, or typed characters and convert them into digital data. The OCR process begins by capturing an image containing text, often using a scanner or camera. The image is converted into pixels, with each pixel containing information about color and shape. Next, the OCR software analyzes the image to detect characters and distinguish them from the background. This step may involve "cleaning" the image to remove visual noise like spots or shadows. The software identifies each character and word by using methods based on letter shapes. Modern OCR systems utilize neural networks or other machine learning algorithms, enhancing their accuracy, even for handwritten text or challenging fonts. Once characters have been identified, they are converted into digital text, which users can then edit, search, or use within other software.

Final Check; License Plate Comparison

In this phase, we review the near-duplicates found in the previous step, first extracting the license plate from each image and then comparing them. This step ensures that no essential data is lost, as the use of hashing may introduce some loss of precision.

Experimental Results

In this section, we outline the evaluation plan to demonstrate the effectiveness of our image deduplication technique in enhancing license plate recognition (LPR) systems. Our evaluation aims to determine whether removing nearly duplicate images from the dataset using image deduplication improves the performance of a YOLOv8 object detection model trained on a customized dataset for LPR.

Computational Environment Used for Experimentation

- Operating System: Ubuntu 22.04.4 LTS x86_64
- Hôte : 80XH Lenovo ideapad 320-15ISK
- Kernel : 6.5.0-35-generic
- Shell : bash 5.1.16
- CPU : Intel i3-6006U (4) @ 2.000GHz

- GPU : Intel HD Graphics 520
- Memoire : 11735MB
- Storage : SSD, 120 GB

Dataset

In the experimental phase, we utilized the CCPD dataset, which contains images of Chinese vehicles. This dataset is extensive, diverse, and meticulously annotated. The images in CCPD were collected from an urban parking management company in a provincial capital in China. For specific street locations, the company records parking fees by logging details such as the license plate number, cost, and parking duration, along with a front or rear photo of the vehicle as proof. Consequently, CCPD includes images captured under a variety of lighting conditions and weather environments. Since the only requirement for capturing these photos is to include the license plate, images vary in angles and positions and occasionally include slight motion blur. This results in a rich dataset with images taken from different perspectives and angles, sometimes with a bit of blur.

CCPD contain:

- train: 5769 images for the train
- test: 5006 images for the test
- Valid: 1001 images for the validation

YOLOv8 Object Detection Model

YOLOv8 is the latest version of YOLO by Ultralytics. As a state-of-the-art (SOTA) model, YOLOv8 builds on the success of previous versions by introducing new features and improvements for enhanced performance, flexibility, and efficiency. YOLOv8 supports a comprehensive range of AI vision tasks, including detection, segmentation, pose estimation, tracking, and classification. This versatility allows users to leverage the capabilities of YOLOv8 across various applications and domains.

Evaluation Metrics

We will use standard metrics (mAP50, mAP50-95, Precision, and Recall) commonly employed to evaluate object detection models to assess the impact of image deduplication on YOLOv8's performance in license plate recognition. These metrics consider the trade-off between the model's ability to correctly identify license plates and its capacity to find all license plates present in the images.

Evaluation Procedure

To evaluate the effectiveness of image deduplication on YOLOv8's performance in license plate recognition (LPR), we will follow a structured evaluation procedure. We will use a multi-step training approach to assess the impact of image deduplication on YOLOv8's performance in license plate recognition:

Training the Model without Deduplication:

Initially, we will train a baseline YOLOv8 model using the original dataset without applying any deduplication. This baseline model serves as a reference point for comparison with the results obtained using our deduplication technique (Table 1).

Table 1. Results without deduplication

Variables	Values
Precision	0.89902
Rappel	0.94274
mAP50	0.91463
mAP50-95	0.47924

Model Training with Deduplication

We will then explore the impact of different image deduplication parameters on model performance. This will involve multiple training runs, each with a distinct set of deduplication parameters. These parameters include: similarity threshold, hash size, and hash bands (Table 2)

Table 2. Results with deduplication

Variables	values			
Hash Signature Size	1024	4096	16384	65536
Hash Signature Bands	64	32	16	64
Threshold	0.9	0.8	0.9	0.9
Precision	0.94238	0.94238	0.94246	0.94238
Recall	0.92408	0.92408	0.90002	0.92408
mAP50	0.95009	0.95009	0.94672	0.95009
mAP50-95	0.48136	0.48136	0.48993	0.48136

Conclusion

In this paper, we explored the enhancement of Automatic Number Plate Recognition (ANPR) systems through the integration of image deduplication techniques. Our research found that established deduplication methods significantly contribute to the accuracy, efficiency, and robustness of ANPR systems by improving dataset quality.

Through evaluating various deduplication algorithms, we identified effective strategies for removing redundant images, refining the data used by ANPR models. Our experiments, conducted on the CCPD dataset in China, showed measurable improvements in recognition accuracy and reduced computational costs, validating the hypothesis that deduplication strengthens ANPR system performance.

This work fills a gap in current research, which has largely focused on recognition algorithms without considering the benefits of preprocessing with deduplication. We demonstrate the potential of deduplication for ANPR systems and open avenues for further optimization of preprocessing steps to advance smart city initiatives and traffic management capabilities.

Scientific Ethics Declaration

The authors declare that the scientific ethical and legal responsibility of this article published in EPSTEM Journal belongs to the authors.

Acknowledgements or Notes

* This article was presented as an poster presentation at the International Conference on Technology, Engineering and Science (www.icontes.net) held in Antalya/Turkey on November 14-17, 2024.

References

Berrabah, D., & Gafour, Y. (2023). Improve image classification using data optimization. *The Eurasia Proceedings of Science Technology Engineering and Mathematics,26*, 262-271.

Hamadouche, M., Zebbiche, K., Guerroumi, M., Tebbi, H., & Zafoune, Y. (2021). A comparative study of perceptual hashing algorithms: Application on fingerprint images. In *The 2nd International Conference on Computer Science's Complex Systems and their Applications*.

Ke, Y., Sukthankar, R., & Huston, L. (2004). An efficient parts-based nearduplicate and sub-image retrieval system. *Proceedings of the 12th Annual ACM International Conference on Multimedia,* 869–876.

Lee, K, M, (2012). Locality-sensitive hashing techniques for nearest neighbor search. *International Journal of Fuzzy Logic and Intelligent Systems*, *12*(4), 300– 307.

Lei, Y., Qiu, G., Zheng, L., & Huang, F. (2014). Nearduplicate image detection using uniform randomized trees. *ACM Transactions on Multimedia Computing, Communications, and Applications, 10*, 1–15.

Li, L., & Zic, J. (2014). Image matching algorithm based on feature-point and DAISY descriptor. *Journal of Multim.*, *9*(6), 829-834.

Li, Z., & Feng, X. (2013). Near duplicate image detecting algorithm based on bag of visual word model. *Journal of Multimedia, 8,* 557–564.

Nian, F., Li, T., Wu, X., Gao, Q., & Li, F. (2016). Efficient near-duplicate image detection with a local-based binary representation. *Multimedia Tools Applications, 75*(5), 2435–2452.

Samanta, P., & Jain, S. (2021). Analysis of perceptual hashing algorithms in image manipulation detection. *Procedia Computer Science*, *185*, 203-212.

Slaney, M., & Casey, M. (2008). Locality-sensitive hashing for finding nearest neighbors (lecture notes). *IEEE Signal Processing Magazine*, *25*(2), 128-131.

Velmurugan, K., & Baboo, L. D. S. S. (2011). Content-based image retrieval using SURF and colour moments. *Global Journal of Computer Science and Technology*, *11*(10), 1-4.

Author Information

Djamel Berrabah
EEDIS, University of Djillali Liabes
Sidi Bel Abbes, Algeria
Contact e-mail: *berrabdjamel@gmail.com*

Yacine Gafour
University Ibn Khaldoun
Tiaret, Algeria

To cite this article:

Berrabah, D., & Gafour, Y. (2024). Significant improvement in license plate recognition through image deduplication. *The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM), 32,* 295-303.