Re-identification of Cars

Computer Vision - 24/25Z
VUT FIT

Bruna Marques, João Teixeira, Georgiana Antreou
{xmarqub00, xteixej00, xantreg00}@stud.fit.vutbr.cz

December 24, 2024

1 Task Definition

Cars can be identified on the basis of license plates, but it requires high-quality video recording and appropriate camera placement. This project aims to develop a system for identifying the most visually similar vehicle from a dataset of car images to the cars detected in video footage. The dataset, derived from PKU-VD, contains around 15000 diverse vehicle images that serve as a reference for matching. The system must employ computer vision techniques, including deep learning models for feature extraction, object detection, and similarity measurement, to identify similar vehicles.

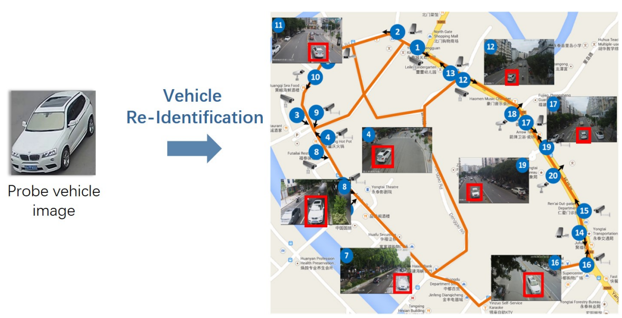

Figure 1: Example of vehicle re-identification

2 Related Work

Early attempts at vehicle re-identification focused on handcrafted features to capture the unique characteristics of vehicles. Techniques like Scale-Invariant Feature Transform (SIFT) and Histogram of Oriented Gradients (HOG) were commonly used. These methods relied on detecting specific patterns in vehicle parts, such as lights, license plates, or shapes. However, their performance was limited due to the lack of robustness against significant viewpoint changes and environmental variations.

Vehicle re-identification is a well-researched problem in computer vision, with several existing solutions employing deep learning and advanced matching techniques. Most frameworks are built upon the use of convolutional neural networks (CNNs), which are adept at extracting high-level visual features from vehicle images. Pre-trained models such as ResNet50 and VGG16 have been extensively utilized to generate embeddings that represent key attributes of vehicles, including shape, color, and texture [MX24].

For detecting vehicles in video sequences, object detection models like YOLO (You Only Look Once) and Faster R-CNN have shown state-of-the-art performance in terms of speed and accuracy. These models are particularly effective in identifying bounding boxes around vehicles, which can then be cropped and processed for feature extraction.

To improve multi-frame tracking in video streams, algorithms like DeepSORT (Simple Online and Realtime Tracking) are commonly integrated. These frameworks combine motion-based tracking with appearance-based re-identification, enabling consistent tracking of vehicles over time.

Despite substantial progress, challenges remain in achieving reliable car re-identification under extreme conditions, such as severe occlusions, identical car models, and adverse weather.

3 Description of solution

The solution integrates state-of-the-art models and techniques to try to achieve efficient and accurate vehicle similarity matching.

The PKU-VD dataset serves as the reference for matching vehicles detected in video footage. Each image in the dataset is resized to 224x224 pixels and preprocessed using standard normalization techniques. Features are extracted using a pre-trained ResNet50 model, configured to exclude the fully connected layers. The resulting embeddings are stored in a serialized cache for efficient retrieval during runtime.

YOLOv5, a real-time object detection model, is employed to detect vehicles in each frame of the video. The detections are filtered to retain only cars, based on their class labels. Detected bounding boxes are then cropped and processed for feature extraction.

For each detected car, visual features are extracted using the same ResNet50 model used during dataset preparation. This ensures compatibility between the video detections and the reference dataset embeddings. The extracted features are normalized and prepared for similarity computation.

The similarity between the detected vehicle features and the dataset embeddings is computed using cosine similarity. The dataset image with the highest similarity score is identified as the closest match to the detected vehicle.

Matched cars from the dataset are displayed alongside the corresponding detections in the video. A grid-based visualization is implemented to showcase the top matches, providing an intuitive way to evaluate the similarity results.

A feature cache mechanism is employed to store recent computations, reducing redundant processing and enhancing runtime performance. Additionally, periodic cache clearing ensures that memory usage remains within acceptable limits during extended operations.

4 Experiments and evaluation results

Since this application processes videos and checks similarity in real-time, we couldn't assess accuracy quantitatively. Instead, we performed a qualitative evaluation by testing it with various videos and assessing the quality of the results.

In the videos used for testing, in most cases the program is able to identify similar cars correctly by color, which makes sense given that it is one of the most visible features. Other characteristics that help the program to identify similar cars is height and width of the vehicle. Usually it is able to identify correctly if it is a bigger truck/SUV, a regular-sized sedan/coupe or a smaller hatchback.

Another feature that the program was good at identifying were sunroofs. It was usually able to match a car with sunroof in the videos to a car with sunroof in the database.

In Figure 2, we can see that the developed program is able to match the colors, design, and size of the vehicles. We can also see that, out of the 3 vehicles in the image, it was only able to find a similar car for 2 of them. This situation repeats frequently in the videos used for testing. This shows that the program is not error-free, as identifying vehicles, especially when many are not in our database, is a challenging task. An example of an error is shown in Figure 3, where a car parked on the left side of the image is incorrectly identified as a truck, despite the somewhat matching colors. Also, the problem of only 2 out of the 3 cars being recognized still persists.

There were also some problems regarding performance. Even with the optimizations done, the video was still running at very low FPS, around 10, which does not provide the user with a good experience and hinders the possibility of this program being used in a real world setting.

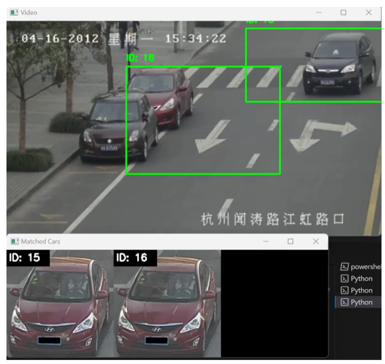

Figure 2: Vehicle identification by the program developed for this project

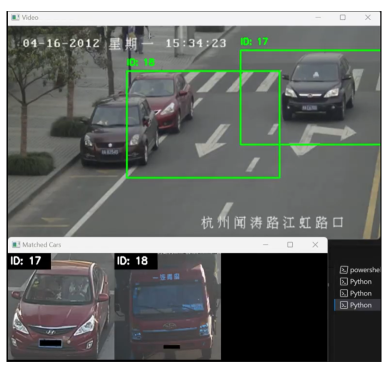

Figure 3: Error in vehicle identification (ID: 18)

References

[MX24] Changxi Ma and Fansong Xue. "A review of vehicle detection methods based on computer vision". In: *Journal of Intelligent and Connected Vehicles* 7.1 (2024), pp. 1–18. DOI: 10.26599/JICV.2023.9210019. URL: https://www.sciopen.com/article/10.26599/JICV.2023.9210019.