Realtime License Plate Recognition

Timofey Brayko
t.brayko@innopolis.university
BS22 AAI-02

Artemii Miasoedov
a.miasoedov@innopolis.university
BS22, AAI-02

I. Introduction

License Plate Recognition (LPR) is a well-established task with numerous practical applications in the real world. These applications include controlling access through checkpoints, automatic gate opening, traffic monitoring, and issuing traffic violation tickets. While the task may seem straightforward, the complexity arises from the need for the system to accurately track and identify thousands of license plates daily, under various weather conditions. This is where the real challenges begin.

Traditional computer vision (CV) methods have shown limitations in accurately recognizing license plates in real-time, especially under challenging conditions such as varying lighting, occlusions, and different angles. To address these challenges, we propose the use of new D-Fine model [1], a deep learning approach that offers several advantages over classical CV methods. This model has been published nearly two month ago, but has already achieved in great results in Real-Time Object Detection [4].

II. Related Work

License Plate Recognition has been an active area of research for several decades. Early approaches relied on classical computer vision techniques such as edge detection, morphological operations, and template matching. While these methods were effective in controlled environments, they often failed in real-world scenarios due to their sensitivity to variations in lighting, occlusions, and plate orientations.

With the advent of deep learning, significant advancements have been made in LPR. Convolutional Neural Networks (CNNs) have shown promising results in feature extraction and classification tasks. Models like YOLO (You Only Look Once) and SSD (Single Shot MultiBox Detector) have been adapted for LPR, demonstrating improved performance over traditional methods.

III. Methodology

Our project focuses on the implementation and evaluation of the D-Fine model for real-time License Plate Recognition. The methodology involves several key steps, including data collection, preprocessing, model training, and evaluation.

A. Data Collection

For our project on real-time License Plate Recognition, we leveraged the PlatesMania dataset [3], which is renowned for its extensive collection of photographs and regular updates. However, a significant limitation of this dataset is that it only provides images and labels for the license plates, without bounding boxes. To overcome this challenge, we manually annotated approximately 2,000 photographs with bounding boxes.

B. Model Architecture

Since classical computer vision algorithms are considered to be unsatisfactory for real-world applications, we decided to adopt an ML-based approach. However, instead of utilizing YOLO, which has become a state-of-the-art solution, we opted for the novel D-Fine architecture, developed by Yansong Peng et al [2].

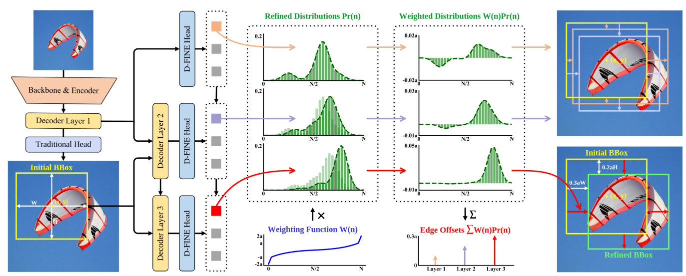

Fig. 1: Overview of D-Fine architecture with FDR. Source: Adapted from [2].

The authors [2] propose a newest method for detecting bounding box elements using Fine-grained Distribution Refinement (FDR). The first decoder generates an initial bounding box through the traditional bounding box regression approach, serving as a reference. Subsequent decoder layers then refine this bounding box by progressively analyzing the image at different zoom levels and adjusting its position, thereby enhancing its accuracy.

C. Workflow

The workflow will be based on a tandem of two trained D-Fine models, showing high accuracy and robustness in real-time applications. The first model will be responsible for detecting license plates within the frame. The extracted and cropped license plate regions will then be passed to the second, model, which will handle the extraction of the alphanumeric characters from the plate.

IV. Experiments and Evaluation

To thoroughly evaluate the performance of the D-Fine model in real-time LPR, we conducted a series of experiments. Our goal was to compare the D-Fine model both with classical CV algorithms. Assessing its effectiveness in various scenarios will demonstrate the project's real-world feasibility.

A. *Classical CV Algorihtms*

Initially, we explored the use of classical computer vision techniques, such as morphological analysis and Canny edge detection. We began by applying the Canny edge detection algorithm to identify the edges of license plates in the images. While this method effectively highlighted the boundaries of the plates in ideal conditions (Fig. 2), it struggled with images containing noise, low contrast, or complex backgrounds (Fig. 3). The edges of the license plates often blended with other edges in the scene, making it difficult to isolate the plates accurately. Each separate case required its own settings for hyperparameters, making this approach unsuitable.

Fig. 2: Positive samples of Canny Edge Algorithm

Fig. 3: Negative samples of Canny Edge Algorithm

B. *D-Fine Solution*

The first model, designed for license plate detection, was trained using manually annotated data. We utilized the source code of the architecture from the authors' repository [2] and trained the model from scratch, without using pre-trained checkpoints for COCO or Object365. To further improve the model's generalization, the dataset was subjected to augmentation techniques like rotation, scaling, and noise addition. This approach ensured reliable performance even on license plates with challenging perspectives or environmental distortions. This dataset proved to be sufficient for achieving strong performance on the validation set, demonstrating excellent metrics (AP=0.96%, AR=0.97%). For detailed results, see Figures 6, 7 and 8 in Appendix A.

Fig. 4: Successfully identified license plates.

When we trained the D-Fine model for license plate extraction, we faced a major obstacle: the dataset was insufficient to handle the complexity of this task. To overcome this, we implemented a semi-supervised learning strategy, using labels PlatesMania dataset provides. This involved leveraging the trained detection model to create synthetic labels from additional images. By incorporating this synthetic labels into the training process and expanding our annotated dataset, we significantly increased its size up to 68.000 images. This approach streamlined data preprocessing and resulted in a notable improvement in the model's generalization performance. In most cases, the model successfully extracts text from license plates, achieving high accuracy. However, it may still make errors in isolated instances 5.

Fig. 5: Errors in License Plate Number Extraction.

V. Analysis and Observation

Detection Model Performance: The detection model, using the D-Fine architecture, demonstrated exceptional performance. With high average precision (AP) and average recall (AR) metrics on the validation set, the model successfully identified and localized license plates with remarkable accuracy. The detection model's robustness was evident in its ability to handle various challenging conditions, including different lighting, angles, and occlusions. This performance underscores the effectiveness of the D-Fine model's advanced neural network architecture and training techniques.

Recognition Model Performance: The recognition model, also based on the D-Fine architecture, showed promising results but requires further training to reach its full potential. While the model achieved good accuracy in recognizing license plate characters, it struggled with more difficult and distinguishable samples. However, we observed significant progress in the model's learning curve. As we introduced more challenging samples during training, the model quickly adapted and improved its recognition capabilities. This rapid learning indicates the model's potential to handle complex scenarios with additional training data and fine-tuning.

Semi-Supervised Learning Impact: The adoption of a semi-supervised learning approach played a crucial role in enhancing the recognition model's performance. By generating synthetic labels and iteratively training the model, we significantly increased the size and diversity of our dataset. This approach not only reduced the time required for data preprocessing but also improved the model's generalization capabilities.

VI. Conclusion

Our project on real-time License Plate Recognition using the D-Fine model has shown significant improvements over classical computer vision methods. The D-Fine model excelled in both detection and recognition tasks, demonstrating high accuracy and robustness in diverse and challenging conditions.

The detection model performed exceptionally well, accurately identifying and localizing license plates. The recognition model showed promising results but requires further training to handle more complex samples. The semi-supervised learning approach enhanced the model's generalization capabilities.

Comparisons with classical CV algorithms implemented using OpenCV highlighted the limitations of traditional methods. The D-Fine model's deep learning approach provided a more robust and adaptable solution, outperforming methods based on classical edge detection in all evaluated metrics.

Future improvements can be achieved by collecting more diverse training data, fine-tuning the model, exploring hybrid approaches, and optimizing for real-time processing. Overall, our project demonstrates the advantages of the D-Fine model in real-time LPR tasks, providing a solid foundation for future enhancements and practical deployments.

References

[1] Artemii Miasoedov and Timofey Brayko, "Github repository of the project". [Online]. Available: https://github.com/ATMI/Plates-Recognition

[2] Yansong Peng, Hebei Li, Peixi Wu, Yueyi Zhang, Xiaoyan Sun, and Feng Wu, "D-FINE: Redefine Regression Task in DETRs as Fine-grained Distribution Refinement", unpublished, arXiv, 2410.13842, Oct. 2024. [Online]. Available: https://arxiv.org/abs/2410.13842. GitHub Repo: https://github.com/Peterande/D-FINE?tab=readme-ov-file.

[3] "PlatesMania: Source of the dataset". [Online]. Available: https://platesmania.com/ru/

[4] "Object Detection on MS COCO Leaderboard". [Online]. Available: https://paperswithcode.com/sota/real-time-object-detection-on-coco?p=d-fine-redefine-regression-task-in-detrs-as

Appendix A
Appendix A: Metric Plots

Here we provide the additional plots of the metrics of our models.

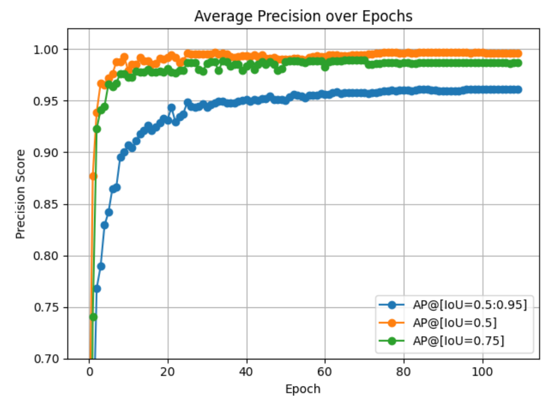

Fig. 6: Average Precision over Epochs.

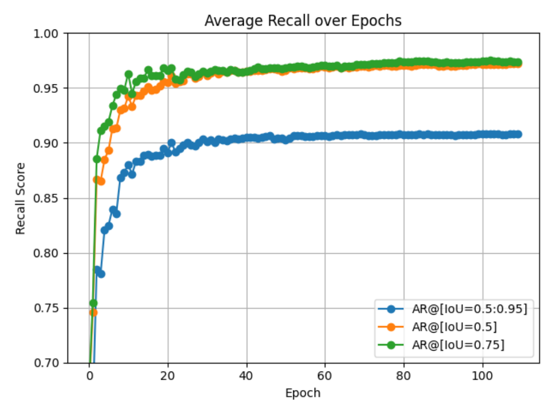

Fig. 7: Average Recall over Epochs.

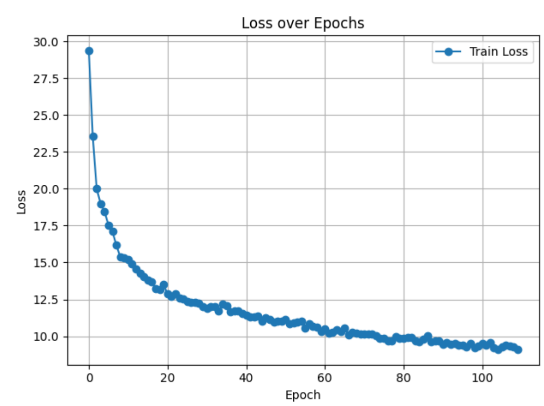

Fig. 8: Loss over Epochs.