Article

Improving Data Augmentation for YOLOv5 Using Enhanced Segment Anything Model

Benyu Xu 1 and Su Yu 2,*

1 School of Mechanical and Automotive Engineering, Shanghai University of Engineering Science, Shanghai 201620, China; m310121250@sues.edu.cn
2 School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China
* Correspondence: yusu@sues.edu.cn; Tel.: +86-138-1732-3259

Abstract: As one of the state-of-the-art object detection algorithms, YOLOv5 relies heavily on the quality of the training dataset. In order to improve the detection accuracy and performance of YOLOv5 and to reduce its false positive and false negative rates, we propose to improve the Segment Anything Model (SAM) used for data augmentation. The feature maps and mask predictions generated by the SAM are used as auxiliary inputs for the Mask-to-Mask (M2M) module. The experimental results show that after processing the dataset with the improved Segment Anything Model, the detection performance of YOLOv5 is improved with 99.9% precision and 99.1% recall. The improved YOLOv5 model has a higher license plate recognition accuracy than the original detection model under strong snowfall conditions, and the incidence of false-negative and false-positive is greatly reduced. The enhanced model can meet the requirement of accurate real-time recognition of license plates under strong snowfall weather conditions.

Keywords: Segment Anything Model; YOLOv5; CCPD; object detection

1. Introduction

With the development of convolutional neural networks, deep learning-based object detection algorithms have gradually replaced traditional object detection algorithms. License plate detection is a subtask of object detection, and many generalized object detection algorithms can be directly applied to license plate detection. Classical two-stage target detection algorithms including R-CNN, Fast R-CNN, and Faster R-CNN have high detection accuracy, but the detection speed is not satisfactory, and these methods also have performance limitations when dealing with scenarios such as small-targeted license plates, occluded license plates, and dense license plates. Classical single-stage target detection algorithms include SSD, RetinaNet, YOLO, FCOS, CenterNet, etc. Their advantage lies in the ability to achieve very high detection speeds with partial loss of accuracy, but there are some limitations in dealing with target license plates of different scales and robustness to occluded license plates. Even though YOLOv5 is one of the most advanced target detection algorithms available, its performance depends heavily on the training dataset. Therefore, there is still room for improvement in detecting license plates in the face of complex scenarios such as abnormal weather and tilted license plates. With the continuous development in the field of target detection, the solution to these problems is also one of the focuses of future research.

The Segment Anything Model (SAM) [1] has garnered significant interest in recent times due to its outstanding performance and has become a foundational model in computer vision. Processed images by SAM provide a solid foundation for downstream tasks, showcasing its immense potential in computer vision. The powerful segmentation and zero-shot transfer capabilities of SAM can enhance the performance of visual tasks. In the
field of medical image segmentation, Lei et al. [2], cleverly integrated MedLAM (a limited annotation framework) with SAM. By annotating only six extreme points in three directions on several templates, the model autonomously recognizes target anatomical areas on all data designated for annotation, significantly reducing the annotation workload in medical image segmentation. For 3D object detection, Zhang et al. [3] proposed a SAM-powered BEV processing pipeline to leverage SAM’s substantial zero-shot capabilities for zero-shot 3D object detection, confirming SAM’s applicability in 3D object detection. Zhang et al. [4] introduced an end-to-end SpA-Former by incorporating SAM, recovering shadow-free images from a single shadow image. Unlike traditional methods that require two steps for shadow detection and removal, SpA-Former unifies these steps into one, eliminating the need for a separate shadow detection step and any post-processing refinement steps, effectively integrating shadow detection and removal into a single stage. Li et al. [5] utilized SAM as a guiding module, combined with the lightweight Mask-to-Matte (M2M) module, to precisely improve mask outputs into alpha masks for target instances, addressing various image matting tasks. Inspired by Matting Anything, deducting detected targets in advance facilitates YOLOv5 in target detection. This approach aims to reduce the false positive and false negative rates of YOLOv5, thereby enhancing detection accuracy and performance.

2. Materials and Methods
2.1. YOLOv5

YOLOv5 [6] is a one-stage object detection model with four versions: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. The network structures of these versions are identical, with differences in network depth and width achieved through the parameters of depth_multiple and width_multiple. The overall framework structure of the YOLOv5 model is shown in Figure 1.

![Figure 1. YOLOv5 Overall Frame Structure.](image)

The structural characteristics of YOLOv5 include the adoption of Cross-Stage Partial (CSP) [7] and Spatial Pyramid Pooling Fast (SPPF) [6,8] methods in the backbone network, as well as the utilization of Feature Pyramid Network (FPN) [9] and Path Aggregation...
Network (PAN) [10] methods in the neck network. Additionally, techniques such as Mosaic data augmentation [11], adaptive anchor box calculation, and adaptive image scaling are employed to enhance training effectiveness.

Since its release, YOLOv5 has gained widespread recognition in both academia and industry. Similar to one-stage algorithms such as SSD [12], YOLO series [13–17], Centernet [18], Efficient Det [19], YOLOv5 treats object detection as a regression task, directly regressing bounding boxes and predicting the categories at multiple positions in the entire image for comprehensive information. As one of the most advanced target detection algorithms, the performance of YOLOv5 depends heavily on the dataset used for training, and there is still room for improvement in license plate detection in the face of abnormal weather scenarios.

The quality of the training dataset directly affects the model’s generalization ability and detection accuracy [20]. YOLOv5 requires a large amount and high quality of labeled data to learn the features and contextual information of the target. If the training dataset is of poor quality, the model may produce misleading learning, leading to performance degradation. Second, the diversity of training datasets is crucial to improve the robustness of the model. Factors such as different scenes, lighting conditions, weather conditions, and target poses can affect the performance of the detector. Therefore, the training dataset needs to cover a variety of possible real-world scenarios so that the model can perform effective detection in different environments. In addition, the size of the training dataset is also an important factor that affects the performance of the model. Larger datasets usually provide more samples and richer contextual information, which helps the model learn a more complex and accurate representation of the target. In contrast, if the size of the training dataset is small, the model may face the problems of overfitting and insufficient generalization ability. Therefore, in this paper, the improved Segment Anything Model is used for data augmentation as a way to build the dataset so that the detection target is more accurate and the detection performance of YOLOv5 is improved.

2.2. Segment Anything Model

The SAM is a breakthrough in image segmentation, carefully designed to help accurately delineate any object in an image. At its core, SAM utilizes state-of-the-art deep learning methods to meticulously analyze and segment a wide variety of objects, and initiates this complex process at the individual pixel level of granularity. This technology enables SAM to achieve unparalleled high-precision semantic segmentation, setting a new benchmark in the field. The model skillfully integrates the power of Deep Convolutional Neural Networks (DCNN) [21–25] with the advanced capabilities of semantic segmentation networks [26]. This strategic combination enables SAM to accurately interpret and segment images, effectively distinguishing between different objects and their respective contexts. In this way, SAM addresses a critical need in applications ranging from autonomous driving to medical imaging, where the ability to accurately recognize and segment objects can significantly improve performance and outcomes.

SAM adopts advanced attention mechanisms [21,23] to better capture contextual information of various regions in the image. It employs multi-scale feature fusion [25] to enhance segmentation accuracy. For a given image \( I \in \mathbb{R}^{H \times W} \), SAM first utilizes a ViT-based image encoder to obtain deep feature maps \( F \in \mathbb{R}^{C_{16} \times 16 \times 16} \). Then, it encodes \( N \) input prompts using a prompt encoder and sends them to the mask decoder along with the feature maps. The mask decoder returns a set of masks \( m_i \in \mathbb{R}^{H \times W} \) (\( i \in N \)) specified by the input prompts.

The sophisticated design of SAM is underscored by its flexible prompt mechanism, which facilitates not only interactive usage [27,28], but also ensures that it excels in dealing with intricate scenes and visuals populated with multiple objects. This capability is especially critical in environments where detail and nuance are paramount. SAM’s architecture is engineered to be highly sensitive to edge information, a feature that plays a pivotal part in accurate object delineation.
role in accurately delineating the contours and boundaries of objects within an image. This level of precision is crucial for creating clear, distinct representations of each element in a scene, regardless of complexity. The inherent adaptability of SAM, stemming from its nuanced understanding of edge information and interactive prompt mechanism, makes it an ideal solution for a broad range of downstream tasks. These tasks could include, but are not limited to, image segmentation, object recognition, and even complex scene reconstruction, where the accurate capture and interpretation of object boundaries are essential for high-quality outcomes. This versatility and high performance in processing complex imagery positions SAM as a valuable tool in the field of computer vision and image processing, offering significant advantages for both research and practical applications.

2.3. Mask-to-Matte

Mask-to-Matte (M2M) is a highly practical image segmentation task aimed at accurately separating foreground objects in an image and generating a detailed transparency mask, referred to as “Matte.” This task provides crucial technological support for computer vision and graphics, especially in applications such as image editing, advertising design, virtual reality, and video production.

The key objective of M2M is to generate high-quality transparency masks from a simple foreground-background binary mask. To achieve this goal, M2M employs advanced techniques from deep learning, including Generative Adversarial Networks (GANs) [29]. Through training these networks, the generator of M2M learns how to generate accurate and detailed transparency masks from the input binary mask, while the discriminator evaluates the consistency between the generated Matte and real images, continuously improving the overall model’s performance.

In terms of technical principles, the M2M deep learning model analyzes images at the pixel level to learn the generation patterns of transparency masks. The generator is responsible for producing transparency masks, while the discriminator plays a role in assessing the quality of the generated Matte. This training approach enables the M2M model to better understand the relationship between foreground objects and the background in an image, producing more realistic and accurate Mattes, significantly enhancing segmentation accuracy.

M2M excels in its precise modeling of transparency, preserving the details of foreground objects by learning the transparency levels of each pixel in the image. The trained model exhibits strong generalization capabilities, adapting well to different scenes and image types, performing exceptionally well in a wide range of application scenarios.

In terms of applications, M2M finds widespread use in image editing, advertising design, virtual reality, and video production. Through its high-quality Matte generation, M2M offers users elevated control, allowing objects to interact transparently with other elements. Integrating M2M as part of the Matting Anything Model (MAM) aims to effectively and smoothly transform instance-aware mask predictions from SAM into instance-aware Alpha Matte predictions.

Given an input image \( I \in \mathbb{R}^{H \times W} \), the pre-trained SAM model produces a feature map \( F \in \mathbb{R}^{C \times H \times W} \) and mask prediction \( m \in \mathbb{R}^{1 \times H \times W} \) on the target instance with immediate guidance. The rescaled images, masks, and feature maps are connected to form the input \( F_{m2m} \in \mathbb{R}^{(C+4) \times H \times W \times \frac{W}{8}} \) to the M2M module. M2M employs several refinement blocks containing connected self-attention layers, batchnorm layers, and activation layers to generate alpha matte predictions, denoted as \( \alpha_{\text{os}} \in \mathbb{R}^{1 \times \frac{H}{8} \times \frac{W}{8}} \), at 1/8 resolution. The feature maps are then up-sampled to a higher resolution at 1/4 and full resolution alpha matte predictions are performed, denoted as \( \alpha_{\text{os}} \in \mathbb{R}^{C \times \frac{H}{4} \times \frac{W}{4}} \) and \( \alpha_{\text{os}} \in \mathbb{R}^{C \times H \times W} \), respectively. Multi-scale prediction with M2M combination enables MAM to handle objects at different scales and provide finer-grained alpha masking for detailed object extraction.
2.4. Matting Anything Model

Given an image \( I \), it can be considered as a combination of foreground image \( F \) and background image \( B \). As shown in formula (1), the coefficient is alpha matte \( \alpha \). Image matting is the estimation of \( \alpha \) when only \( I \) is provided as input. When the image \( I \) contains multiple instances, synthesis becomes formula (2).

\[
I = \alpha F + (1-\alpha)B \tag{1}
\]

\[
I = \sum_{i}^{N} \alpha_i F_i + (1 - \sum_{i}^{N} \alpha_i)B \tag{2}
\]

\( \alpha_i \) represents the alpha mask of instance \( i \). Typically, target and reference masks are used to guide the prediction of instance-aware alpha masks, or natural language descriptions are employed to estimate the alpha matte of the target instance. However, these methods are designed for specific scenarios with corresponding benchmarks, limiting their potential to simultaneously handle various image matting tasks and benchmarks.

The Matting Anything Model (MAM) is a versatile network designed to estimate the alpha matting of any target instance in an image using prompt-based user guidance, as illustrated in Figure 2.

![Figure 2. Matting Anything Model Processing Flow.](image)

MAM leverages the advanced SAM framework, which supports flexible prompts and outputs segmentation masks for any target instance for interactive use. Specifically, MAM takes feature maps and mask outputs from SAM as input, and incorporates a lightweight Mask-to-Matte module to predict the alpha mask for the target instance. This allows MAM to efficiently generate high-quality alpha mattes for diverse target instances based on SAM’s segmentation capabilities.

During the training process, the foreground instance \( F \in R^{H \times W} \) is combined with its corresponding ground truth \( \alpha_{\mu} \) \( \in R^{H \times W} \) and background image \( B \in R^{H \times W} \) to create synthetic images. The synthesis process is conducted using Formula (3), followed by the extraction of bounding boxes \( (x_0, y_0, x_1, y_1) \), encapsulating the interested instances in the synthetic images. Images \( I \) and bounding boxes are sent as cues to the pre-trained SAM, which returns the mask prediction for the instance. Finally, the images, masks, and feature maps are concatenated and sent to the M2M module. The M2M module further returns
multi-scale alpha mask predictions $\alpha_{\text{os}}$, $\alpha_{\text{os4}}$, $\alpha_{\text{os1}}$. The loss function $L$ is computed between multi-scale predictions and ground truth $\alpha_{\text{gt}}$, formulated as (4), where coefficients $\lambda_{\alpha}$ and $\lambda_{Lap}$ control the contribution of each loss term. Both loss terms are calculated on multi-scale prediction as formula (5) and formula (6).

$$I = \alpha_{\text{gt}} F + (1- \alpha_{\text{gt}}) B \quad (3)$$

$$L(\alpha_{\text{gt}}, \alpha_{\text{os}}, \alpha_{\text{os4}}, \alpha_{\text{os1}}) = \lambda_{\alpha} L_{\alpha} + \lambda_{Lap} L_{Lap} \quad (4)$$

$$L_{\alpha} = L_{\alpha}(\alpha_{\text{gt}}, \alpha_{\text{os}}, \alpha_{\text{os4}}) + L_{\alpha}(\alpha_{\text{gt}}, \alpha_{\text{os1}}) \quad (5)$$

$$L_{Lap} = L_{Lap}(\alpha_{\text{gt}}, \alpha_{\text{os}}, \alpha_{\text{os4}}) + L_{Lap}(\alpha_{\text{gt}}, \alpha_{\text{os1}}) \quad (6)$$

3. Experiments and Results

3.1. Dataset

The widely used open-source dataset for large-scale license plate detection and recognition in complex scenes in current domestic research is the CCPD [30] dataset created by the University of Science and Technology of China. The CCPD dataset captures license plate photos in various challenging environments, including blur, tilt, rainy, snowy, etc., comprising over 250,000 images of Chinese urban license plates. The authors have divided it into nine sub-datasets, as shown in Table 1.

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
<th>Quantities</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCPD-Base</td>
<td>Generic license plate</td>
<td>199,996</td>
</tr>
<tr>
<td>CCPD-FN</td>
<td>The license plate is relatively close or far from the camera’s capture location</td>
<td>20,967</td>
</tr>
<tr>
<td>CCPD-DB</td>
<td>License plate area is brighter, darker or unevenly lit</td>
<td>10,132</td>
</tr>
<tr>
<td>CCPD-Rotate</td>
<td>License plate horizontal tilt and vertical tilt friendly</td>
<td>10,053</td>
</tr>
<tr>
<td>CCPD-Tilt</td>
<td>License plates with greater horizontal tilt and vertical tilt</td>
<td>30,216</td>
</tr>
<tr>
<td>CCPD-Weather</td>
<td>License plate shot in rain, snow, and fog</td>
<td>9999</td>
</tr>
<tr>
<td>CCPD-Challenge</td>
<td>Challenging images in license plate detection and recognition</td>
<td>50,003</td>
</tr>
<tr>
<td>CCPD-Blur</td>
<td>Blurred license plate image due to camera shake</td>
<td>20,611</td>
</tr>
<tr>
<td>CCPD-NP</td>
<td>Pictures of new cars without license plates</td>
<td>20,967</td>
</tr>
</tbody>
</table>

This paper utilizes the CCPD-Weather sub-dataset, which does not inherently distinguish between training, validation, and test sets. Therefore, following the commonly used dataset splitting method in current domestic and international research [31], images are divided into training and test sets at a ratio of 4:1. Additionally, during training, the training set is further split into training and validation sets at a ratio of 4:1.

The dataset is annotated using the LabelImg annotation tool, and the annotation format is in txt. LabelImg is a widely used image annotation tool that supports the visual annotation of targets in images and allows users to save the annotation information for each image in txt format. We chose the txt format as the saving format for the annotation information because of its versatility and simplicity, which can be easily read and processed by a wide range of programming languages and computing frameworks. In our paper, the name of each image file corresponds to its data annotation, and such a naming convention is intended to simplify data management and subsequent processing procedures. For example, the filename "0478-4_1-136&448_441&579-
"441&558_141&579_136&469&448-0_0_8_32_29_24_32-87-45.txt" can be divided into seven fields:

1. "0478" represents the proportion of the license plate area to the entire image area (47.8%).
2. "4_1" represents the horizontal tilt and vertical tilt degrees (horizontal tilt 4 degrees, vertical tilt 1 degree).
3. "136&448_441&579" represents the coordinates of the top-left and bottom-right vertices (top-left vertex (136, 448) and bottom-right vertex (441, 579)).
4. "441&558_141&579_136&469&448" represents the precise coordinates of the four vertices of the license plate in the entire image, starting from the bottom-right vertex.
5. "0_0_8_32_29_24_32" represents the index of each character. The last character in each array is the letter ‘O’, not the digit ‘0’. ‘O’ is used as a symbol for “no character” because there is no ‘O’ in Chinese license plate characters. Therefore, the license plate assembled from the characters is “皖 AJ8508”.
6. “87” represents the brightness of the license plate region (87%).
7. “45” represents the blurriness of the license plate region (45%).
8. To ensure the accuracy and consistency of data annotation, we adopted strict annotation guidelines and conducted multiple rounds of quality checks. This process ensures the reliability and validity of our datasets for model training and evaluation.

3.2. Experimental Setup

The experimental platform and parameters used in this paper are presented in Table 2.

Table 2. Experimental platform and parameters.

<table>
<thead>
<tr>
<th>Title</th>
<th>Releases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Ubuntu18.04</td>
</tr>
<tr>
<td>PyTorch</td>
<td>1.9.0</td>
</tr>
<tr>
<td>CUDA</td>
<td>11.1</td>
</tr>
<tr>
<td>Python</td>
<td>3.8.10</td>
</tr>
<tr>
<td>GPU</td>
<td>GeForce RTX 3090(24G), NVIDIA, Santa Clara, CA, USA</td>
</tr>
<tr>
<td>CPU</td>
<td>Intel(R)Xeon(R)Platinum <a href="mailto:8338C@2.60GHz">8338C@2.60GHz</a>, Intel, Santa Clara, CA, USA</td>
</tr>
</tbody>
</table>

This paper evaluates the algorithm’s performance using Precision (P), Recall (R), and Average Precision (AP) as the main metrics. There are four different cases for the model’s detection results, namely True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Precision and Recall are calculated based on the counts of these four cases, leading to the derivation of other evaluation metrics. The formulas for Precision and Recall are given by Formula (7) and Formula (8) respectively.

\[
\text{Precision} = \frac{TP}{TP + FP} \quad (7)
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \quad (8)
\]

\[
AP = \frac{\sum P_i}{\sum r_i} \quad (9)
\]

Different precision and recall rates will be obtained by using different confidence levels, Recall and Precision are plotted as x and y axes, respectively, and the P-R curves are plotted, and the area covered under the curve is the AP value, which is calculated from
the discrete point of view as formula (9). Where $P$ denotes the corresponding $p_i$ value on the PR curve, and $\sum r = 1$.

3.3. Experimental Process and Results Analysis

In our research, we initially utilize the raw CCPD-Weather dataset as the foundational input for training the YOLOv5 model. This particular dataset, which has not undergone any preprocessing steps, is directly employed to facilitate the model’s learning process. The objective is to assess the model’s baseline performance under varying weather conditions. Following the completion of the training phase, the outcomes, including the various parameters and metrics achieved, are comprehensively detailed and visually represented in Figure 3 of our study. This illustration serves to provide a clear and concise depiction of the model’s performance metrics, derived directly from the training conducted with the unprocessed CCPD-Weather dataset.

Figure 3. Parameter results for training before improvement.

When we subjected YOLOv5 to training using the dataset in its raw, unprocessed form, the outcomes were remarkably high in terms of the model’s accuracy metrics. Specifically, the precision metric reached 98.2%, indicating a high rate of true positive predictions out of all positive calls made by the model. Similarly, the recall rate stood at 98.0%, showcasing the model’s efficiency in identifying true positives from the total actual positives present within the dataset. Additionally, the model demonstrated an average precision of 98.9%, which underscores its overall accuracy across different thresholds of detection.

Despite these impressive metrics, our subsequent evaluation of the model, particularly when applied to the designated test set, revealed some limitations. It became apparent that, although the model performs with high accuracy, there are instances where it generates noticeable errors in the form of false positives and false negatives. This observation is clearly illustrated in Figure 4a, where specific instances of these inaccuracies are highlighted. False positives refer to instances where the model incorrectly identifies an object as a target when it is not, whereas false negatives occur when the model fails to detect an actual target. These discrepancies, despite the high precision and recall rates, indicate areas where the model’s performance can still be optimized for even greater accuracy and reliability.
Figure 4. Comparison of tests before and after improvement. (a) Pre-improvement. (b) Post-improvement.

Figure 5 illustrates the process of processing a license plate image, showing how each image is processed individually by MAM to ensure clarity and accuracy of detection.

The processed license plate images are mixed into the CCPD-Weather dataset, and the allocated dataset is then fed into the YOLOv5 model for training. The parameter results of the training are shown in Figure 6. This figure provides a comprehensive overview of the training results, highlighting the YOLOv5 model’s improved capability to detect license plates accurately within the challenging conditions presented by new CCPD-Weather dataset.
After training on the dataset augmented by MAM, the YOLOv5 model achieves a precision of 99.9%, an increase of 1.7%; recall of 99.1%, an increase of 1.1%; and an average precision of 99.5%, an increase of 0.6%. Testing the model on the same test set reveals significant improvements in false positives, false negatives, and confidence levels for correctly predicted instances, as depicted in Figure 4b.

In order to compare the superiority of the improved algorithmic model proposed in this paper to improve the effectiveness of license plate detection under strong snowfall, comparative experiments are conducted using YOLOv5, SSD, and Faster-RCNN target detection algorithms. As can be seen from Table 3, relative to the original dataset, the dataset processed by the improved SAM makes YOLOv5 have a certain improvement in the precision rate, recall rate, and average precision rate, and the precision rate is 7.8% and 1.5% higher than the Faster-RCNN and SSD models, respectively, and the recall rate is 12% and 1.1% higher than that of SSD and the original YOLOv5 model, respectively. From the testing, the model trained from the improved SAM-processed dataset avoids misdetection and omission, and has a higher confidence level of detection.

Table 3. Comparison of indicators.

<table>
<thead>
<tr>
<th>Evaluation Indicators</th>
<th>$P$</th>
<th>$R$</th>
<th>$mAP$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faster-RCNN</td>
<td>92.1%</td>
<td>100.0%</td>
<td>99.9%</td>
</tr>
<tr>
<td>SSD</td>
<td>98.4%</td>
<td>87.1%</td>
<td>99.2%</td>
</tr>
<tr>
<td>Pre-improvement</td>
<td>98.2%</td>
<td>98.0%</td>
<td>98.9%</td>
</tr>
<tr>
<td>Post-improvement</td>
<td>99.9%</td>
<td>99.1%</td>
<td>99.5%</td>
</tr>
</tbody>
</table>

4. Conclusions

In the field of license plate detection under severe weather conditions, this study pioneers the enhancement of SAM with a focus on improving data augmentation techniques to address and mitigate the dependence of YOLOv5’s performance on its training dataset. By introducing an innovative approach to augmenting the CCPD, we greatly balanced the dataset, which, as demonstrated by our experiments, corrects the performance limitations inherent in YOLOv5 when trained on traditional datasets. This enhanced dataset processed through improved SAM significantly improves the model’s adaptability and ability to detect license plates, even under the most challenging and severe weather conditions such as heavy snowfall.
The key to our study is the careful improvement of the SAM, which generates a well-balanced CCPD dataset. This strategic improvement played a crucial role in the training process of YOLOv5, enabling it to exhibit superior performance under severe weather conditions including, but not limited to, snow, rain, and fog. The adaptability shown by the model, especially in the case of extreme snowfall, proves the effectiveness of our data enhancement strategy.

The significance of this research cannot be overemphasized, as it not only addresses the existing limitations faced by YOLOv5 in harsh weather conditions, but also paves the way for wider applications in real-world scenarios where environmental factors play a key role in license plate detection. The expansion of the CCPD dataset through SAM processing makes a significant contribution to improving the adaptability and overall performance of the license plate detection model, highlighting the importance of continued efforts in dataset expansion. By focusing on expanding the scope of the dataset to cover extreme severe weather conditions, we can ensure the robustness and generalization capabilities of these models, making them more capable of dealing with the variety of challenging conditions prevalent in the real world.

In essence, this research not only reveals the potential of SAM in enhancing data augmentation techniques, but also emphasizes the need for more inclusive and comprehensive datasets. Such advances are critical to advancing the field of license plate detection, especially in the face of unforeseen and extreme environmental conditions, thus ensuring the reliability and validity of these models in a large number of real-world applications.

However, we must also recognize the limitations of the current dataset, i.e., the dataset is mainly dominated by intense snowfall conditions, ignoring other extreme environmental scenarios such as heavy rain, dense fog, and dust storms. In order to take the model to the next level of detection capability and practical applicability, future research must prioritize the expansion of the dataset to include license plates taken under more extreme weather conditions. Incorporating images from sandstorms, tornadoes, and other severe weather conditions will not only improve the model’s detection accuracy, but also enhance its usefulness in real-world applications. This comprehensive approach to dataset collection is essential to ensure the robustness and reliability of the model, allowing it to function optimally across a wide range of environmental challenges.

**Author Contributions:** Conceptualization, B.X. and S.Y.; methodology, B.X. and S.Y.; software, B.X. and S.Y.; validation, B.X. and S.Y.; formal analysis, B.X. and S.Y.; investigation, B.X. and S.Y.; resources, B.X. and S.Y.; data curation, B.X. and S.Y.; writing—original draft preparation, B.X.; writing—review and editing, S.Y.; supervision, S.Y.; project administration, B.X.; funding acquisition, S.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research work was supported by China National Science and Technology Support Program Project “Development and Demonstration of Automatic Production Line for Precision Assembly of Electronic Products” (Project No:2015BAF10B00), Scientific Research Program Project of Shanghai Science and Technology Commission “Dyeing Robot Management Software System” (Project No:17511110204).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Publicly available datasets were analyzed in this study. These data can be downloaded from https://github.com/detectRecog/CCPD (accessed on 16 December 2023).

**Acknowledgments:** The authors thank the editor and anonymous reviewers for providing helpful suggestions for improving the quality of this manuscript.

**Conflicts of Interest:** The authors declare no conflicts of interest.
References


Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.