the Graduate School of Hiroshima University
A Master Thesis of Advanced Science and Engineering

Improvement of Character Detection by Non-maximum Suppression Using Text Features

Yang Fei

The Degree of Master （Engineering）

（Informatics and Data Science Program）

Thesis Supervisor　Takio Kurita

February, 2024

Abstract

For text recognition, we have undergone the important shift from traditional Optical Character Recognition (OCR) to deep learning-based text recognition. Scene text detection is an important task in scene text recognition, which is constrained by conditions such as by the stylization of the text itself, background interference, and real-time performance of practical applications, making it difficult to achieve very high detection accuracy.

The main ways to implement Scene text detection include bounding box regression and pixel segmentation. This paper is mainly based on the first one, and text information is imported in the step of non-maximum suppression (NMS), which makes the traditional object detection model more conducive to the detection of characters in the text of the scene. We focus on modifying the network structure and NMS post-processing algorithms with the Single Shot MultiBox Detector (SSD) network, which is a mainstream in object detection.

As a multi-scale detection model, SSD does not optimize character detection. This paper draws on the relationship between words and made several adjustments：First, some feature layers containing fusion information are added to the network to improve the ability of detecting small targets such as text before NMS processing. Subsequently, adjustments to the intersection of union (IoU) and distance thresholds in NMS post-processing are carried out: in the NMS algorithm, the text information obtained by the receiving network is added to the class score by adding information such as text shape and character distance within the text, making the detector tend to detect text that meets the conditions.

By improving the network structure of SSD and NMS algorithm proposed by us, while increasing controllable training costs, NMS can more accurately filter out the bounding boxes of correct text characters, and its performance also has significant advantages compared to image segmentation methods.

Keywords: Scene text detection, Single Shot MultiBox Detector, NMS post-processing, Inter-character information

Contents

Chapter 1

Introduction

1.1 Background

1.1.1 Scene Text Detection

With the advent of the digital age, images and videos have become indispensable information carriers in people's daily lives. In these visual information, text, as a special symbol, conveys rich and accurate information. Before the outbreak of deep learning, most text recognition methods were based on OCR, which had a high ability to recognize clear and neat document text. Recently, the definition of Scene Text Recognition(STR) has been proposed to identify all kinds of difficult-to-recognizable real text in life scenes. Because this task is particularly challenging, it is necessary to accurately identify both the text area and the specific content of the text at the same time. However, the text in the actual scene has problems such as different shapes and sizes, background interference and blurred text itself, which is difficult to achieve very satisfactory results. In general, STR can first perform Scene Text Detection(STD) and subsequently recognize the acquired text regions, or it can directly perform end-to-end operations [25] to complete the task. Among them, STD has an important role, and its detection difficulty is less than the end-to-end STR model, with higher accuracy and better efficiency.At the same time, for the lightweight nature of the model of STD, training and reasoning are easier and can be extended to mobile terminals to realize real-time STD in mobile scenes.

For the complex task of STD, the current mainstream implementations include regression tasks for text-boxes [12, 13, 17, 19, 24] and image segmentation tasks based on pixels in the text area [1, 3, 23, 26–28], like Fig.1. In addition, some methods [8] have recently benefited from two branches of bounding box regression and image segmentation.

For the former, STD is treated as an ordinary object detection task and is basically implemented on a backbone network like SSD [16], Faster R-CNN [5] with appropriate bounding boxes selected for the text. Due to the limitations of the pre-generated bounding box itself,

Figure 1.1: Scene Text Detection

it is more suitable for regular shaped text.With respect to image segmentation, it can give a more accurate bounding box than bounding box regression and is suitable for STD of irregular shapes. Since it is a pixel-by-pixel or other prediction, anchors to match the bounding box are no longer needed.

1.1.2 Related Work

As mentioned in subsection 1.1.1, there are currently two main branches to achieve the mission goals of STD. In the following discussion, we focus on the related work within the branch of bounding box regression for it.

One of the pioneering works in this domain is the TextBoxes [24] method proposed by Tian et al. They introduced an end-to-end trainable neural network for text detection in natural images. By incorporating a multi-task loss function, the method simultaneously predicts the text presence score and the bounding box coordinates. This work laid the foundation for subsequent advancements in the field.

Building upon TextBoxes, Liao et al. introduced TextBoxes++ [12], which improved the overall performance by incorporating more complex techniques such as densely connected layers and utilizing feature maps at different scales. The approach demonstrated enhanced accuracy and robustness in detecting texts of varying sizes and orientations.

Another notable contribution is the work by Shi et al. with their method for detecting arbitrarily oriented text in the wild [19]. They proposed an approach that combines a fully convolutional network with a Markov random field model, enabling effective detection of text regions with diverse orientations and aspect ratios.

Moreover, Ma et al. introduced an approach known as the Arbitrary Shape Scene Text Detection method [17], which addressed the limitations of rectangular bounding boxes. Their method leveraged a shape robust encoding technique to detect text instances with ar-

bitrary shapes, thereby achieving significant improvements in handling irregular and curved text instances.

Furthermore, the research conducted by Liao et al. [13] also contributed significantly to the advancement of text detection techniques. Their Textboxes++ method incorporated a rotation-invariant detection algorithm, enabling the accurate detection of texts with various orientations and perspectives.

These advancements in bounding box regression methods have significantly improved the robustness and accuracy of Scene Text Detection, allowing for the detection of texts with varying shapes, sizes, and orientations in complex real-life scenes.

In the realm of scene text detection, considerable efforts have been dedicated to enhancing the performance of Non-Maximum Suppression (NMS), a critical post-processing step. Early research by Zhou et al. proposed EAST (Efficient and Accurate Scene Text Detector) [28], introducing a novel text box representation based on rotated rectangles. Their dynamic guided anchoring approach significantly improved NMS, enabling more accurate delineation of scene text boundaries. Following this, Ma et al. presented RRPN (Arbitrary-Oriented Scene Text Detection via Rotation Proposals) in 2018, which introduced Rotation Region Proposals as an advanced NMS strategy. RRPN demonstrated effectiveness in handling scene text of arbitrary orientations, contributing to improved robustness . Liao et al. furthered the innovation with TextField [26], introducing a novel NMS method based on directional fields. TextField was designed to adapt to irregular text scenarios, showcasing versatility in addressing non-uniform text structures . Subsequent research by Liao et al. in 2019 proposed DB (An Irregular Scene Text Detector with Differentiable Binarization). This work introduced differentiable binarization, mitigating the impact of thresholding on NMS and improving the detection performance, especially in scenarios with irregular text shapes. These collective contributions underscore the continuous evolution and refinement of NMS strategies within the domain of scene text detection, aiming to enhance accuracy and robustness across diverse text scenarios.

1.2 Research Significance

The significance of this research lies in the potential improvement it can bring to the post-processing phase, specifically the Non-Maximum Suppression (NMS) part, within the series of techniques based on bounding box regression for Scene Text Detection (STD). By focusing on enhancing the NMS algorithm's efficiency and accuracy, the proposed idea aims to address some of the existing limitations and challenges in the current state-of-the-art methods for STD.

The modification of the NMS component holds promise in optimizing the elimination of redundant bounding boxes, thereby leading to improved precision in text location and reduced false positives. Through a more refined and adaptive NMS strategy, the proposed approach can potentially enhance the overall performance of scene text detection systems,

particularly in scenarios where there is significant text overlap or cluttered backgrounds.

Furthermore, the proposed modifications to the NMS phase have the potential to improve the robustness of the system, making it more adept at handling complex text layouts, irregular shapes, and diverse orientations commonly encountered in real-life scenes. This adaptability is critical for enhancing the generalizability of the text detection system across various environments and text-rich images.

Additionally, the optimization of the NMS component can contribute to the acceleration of the overall text detection process, leading to faster inference times and improved efficiency. This acceleration is particularly crucial in real-time applications and systems deployed on resource-constrained platforms, such as mobile devices and embedded systems, where swift and accurate text detection is essential.These devices have limited inference performance and require less complex deep learning networks. Currently, the best algorithms and models are oriented towards pixel detection such as text edges, which consumes a huge amount of arithmetic power. For the task of recognizing simpler characters, such as trademark detection, street sign recognition and other applications, the amount of text present in this series of scenarios is relatively small, and there is still room for traditional recognition models to be improved.

Overall, the proposed enhancement to the NMS post-processing stage has the potential to significantly advance the capabilities of Scene Text Detection systems, leading to improved accuracy, robustness, and efficiency in text localization, and consequently, fostering broader applicability in a range of practical real-world scenarios.

1.3 Thesis Structure

The aim of this thesis is to discuss the current developments in scene text detection techniques and their important network models, and to address some of the challenges faced by them from the aspect of post-processing. Our research methodology adopted in this paper is mainly based on deep learning and NMS, combined with the introduction of textual information in order to improve the accuracy of character detection within scene text. The specific research methods are as follows:

1. Combing and analyzing the existing network models for object detection as well as scene text detection, summarizing their advantages and disadvantages as well as their scope of application; and focusing on explaining the basic network models used in this thesis.

2. Aiming at the inadequacy of the existing object detection models in terms of their detection ability under the conditions of complex backgrounds and variable texts, a character detection model based on improved post-processing algorithms is proposed to improve its detection accuracy and robustness. Specifically, the NMS algorithm is designed to be more suitable for text, making full use of the character association information within the text, and the feature pyramid network is introduced to make up for the low performance of the original network for small text detection.

3. Testing the model using the Street View House Numbers (SVHN) dataset as a simple form of the text dataset based on the high similarity of the numbers to other characters in the text;

4. Detailed experimental validation of the proposed algorithms and models, analysis of their performance under different conditions, comparative analysis with existing algorithms and models, and validation of their effectiveness through visualization of the post-processing process and actual image detection results.

Chapter 2

Related Knowledge

2.1 Deep Learning Model

2.1.1 Convolutional Neural Network

Convolutional Neural Networks (CNN), as one of the milestones of deep learning, has made remarkable achievements in the fields of image processing, computer vision, and natural language processing since its birth. Similarly, text detection and recognition, as an important task in image processing and computer vision, also has many important applications based on CNN. The development of convolutional neural networks can be traced back to the early 1980's. The Neocognitron [4] proposed by Kunihiko Fukushima is one of the pioneers of CNN, which mimics the structure of a biological visual system with hierarchical feature extraction.However, the work that really laid the foundation for CNN was proposed in 1998 by Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner [11]. They introduced the backpropagation algorithm for training CNN and applied it for the first time to the task of recognizing handwritten characters. Since then, CNN has gradually emerged as one of the representative models for deep learning in several fields.

With the proposal of CNN, its continuous evolution and improvement derived several classical models. The early LeNet-5 laid the foundation for convolutional neural networks, while AlexNet [10] won the 2012 ImageNet competition by introducing techniques such as ReLU activation function and Dropout. Subsequently, GoogLeNet (Inception) [21] pushed the development of deep models by adopting multi-scale convolutional kernels and parallel structure, and ResNet [7] solved the gradient vanishing problem by residual learning mechanism, respectively.MobileNet [9] provided a solution for lightweight deployments, while EfficientNet [22], by optimizing the depth, width and resolution of network achieved a balance between performance and computational overhead. These improvements not only achieve remarkable results in image processing, but also provide flexible solutions for different tasks and application scenarios, opening up new possibilities for the development of

deep learning.

The basic structure of CNN consists of convolutional layer, pooling layer, and fully connected layer, which enables it to effectively learn and represent data features.

1. **Convolutional layer**: The convolutional layer is the core component of CNN. In the convolutional layer, local features are extracted through convolutional operations where a convolutional kernel (also known as a filter) is slid over the input data and a convolutional operation is performed on each local region. The convolution operation has the feature of weight sharing, that is, the same convolution kernel shares parameters on the entire input, which helps to reduce the number of parameters of the model. Through multiple convolutional cores, the convolutional layer can learn different characteristics.

2. **Pooling layer**: The pooling layer is used to reduce the dimensionality of data, reduce computational complexity, and preserve important features. Common pooling operations include maximum pooling and average pooling. Maximum pooling selects the maximum value in the local area, while average pooling selects the average value in the local area. The main purpose of pooling is to downsample features, reduce computational complexity, and retain important information.

3. **Fully connected layer**: The fully connected layer is usually located in the last layers of the network and is responsible for integrating the higher level features to the final output. In a fully connected layer, each neuron is connected to all neurons in the previous layer, forming a fully connected structure. The introduction of fully connected layers allows the network to adapt to different tasks, such as the classification task when the fully connected layer outputs probability distributions for different classes.

Convolutional neural networks build a multilayered deep structure by alternating between convolutional, pooling and fully connected layers. This deep structure helps the network learn features at different levels of abstraction, from low-level edges and textures to high-level objects and scenes. This hierarchical feature learning capability has enabled the success and continued development of CNNs in image processing and other fields.

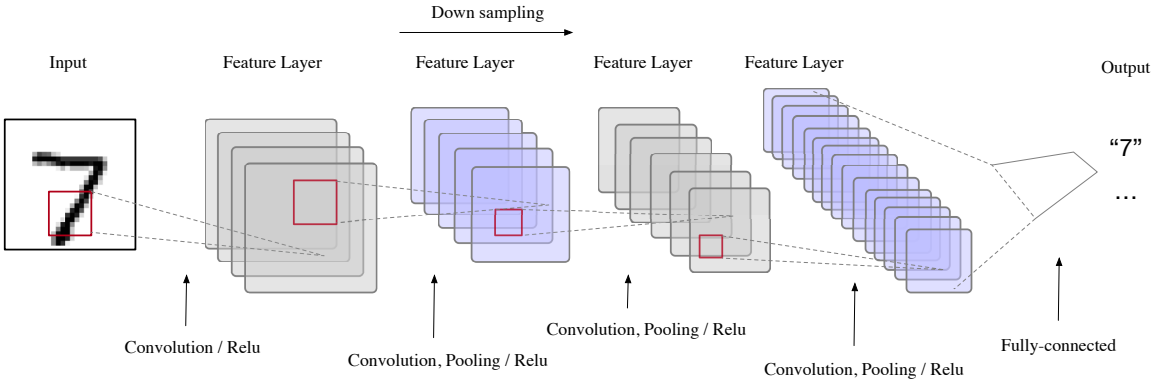

Figure 2.1: A simple convolutional neural network structure for handwritten digit recognition.

2.1.2 VGG16

In the development of Convolutional Neural Network (CNN), the VGG (Visual Geometry Group) model is of great significance, and its simple and powerful design has achieved great success in the field of image classification. VGG16, proposed by Karen Simonyan and Andrew Zisserman in 2014 [20], is another important contribution after LeNet and AlexNet.

Table 2.1: The family of Visual Geometry Group(VGG)

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
Input $(224 \times 224,$ **RGB Image** $)$					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
Maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
Maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
Maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
Maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
Maxpool					
FC - 4096					
FC - 4096					
FC - 1000					
Soft-max					

As a variant of CNN, VGG16 also has three parts: convolutional layer, pooling layer, and fully connected layer：

1. **Convolutional layer**: The convolutional layer of VGG16 is the core of its design. This network contains 13 convolutional layers, each of which uses a small-sized convolution kernel (usually 3x3). This design choice makes the network deeper and can better capture the features in the image. The ReLU activation function is adopted between the convolutional layers, and a nonlinear transformation is introduced. The first few convolutional layers use a single convolution kernel, and with the increase of network depth, the number of convolution kernels gradually increases to enhance the network's ability to learn about complex features.

2. **Pooling layer**: The pooling layer plays an important role in the design of VGG16. The maximum pooling operation is adopted to pool through a 2x2 window with a stride of 2, which gradually reduces the size of the feature map. This helps to reduce computational complexity, retain important features, and improve the model's ability to learn translation invariance. The pooling operation of VGG16 makes the network more capable of sensing and extracting abstract features.

3. **Fully connected layer**: The full connection layer part of VGG16 includes three full connection layers, which are used to integrate high-level features to the final output. The ReLU activation function is also used between the full connection layers, and the final output network's category probability distribution. The introduction of the full connection layer enables VGG16 to adapt to different tasks, such as image classification.

Afterwards, there were also some variants of VGG16 based on different image tasks. VGG-19 further deepens the network on the basis of VGG16, including 19 convolutional layers, with more parameters compared to VGG16. This approach of deepening the network performs better on some complex tasks, but also brings higher computational costs. VGG-19 has demonstrated excellent application performance in fields such as image recognition and scene understanding, also becoming an important model in deep learning research and practice.

2.1.3 MobileNet

The background of scene text detection is based on people's need to obtain text information in various life scenarios. The task requires the model to have efficient and lightweight characteristics, and the carrier for this task is mainly mobile devices. For mobile devices, the traditional convolutional network is too deep, and the computational volume and parameters generated by traditional convolution are huge, which makes it difficult for mobile devices with poor computing power to realize real-time text detection. MobileNet is a lightweight deep neural network proposed by Google for embedded devices such as mobile phones [9]. The core idea used is deep separable convolution.

Deep separable convolution is actually a decomposable convolution operation, which can also be decomposed into two smaller convolution operations: deep convolution and pointwise convolution, as shown in the figure. Deep convolution is different from standard

convolution, for standard convolution, its convolution kernel is used on all input channels, while deep convolution uses a different convolution kernel for each input channel, i.e., a convolution kernel corresponds to an input channel, so deep convolution is a channel-by-channel operation. And point-by-point convolution is actually ordinary 1×1 convolution. For deep separable convolution, the different input channels are first convolved separately using deep convolution, and then the outputs of the above are combined using point-by-point convolution. The overall effect of this is similar to a standard convolution, but it will greatly reduce the amount of computation and the number of model parameters.

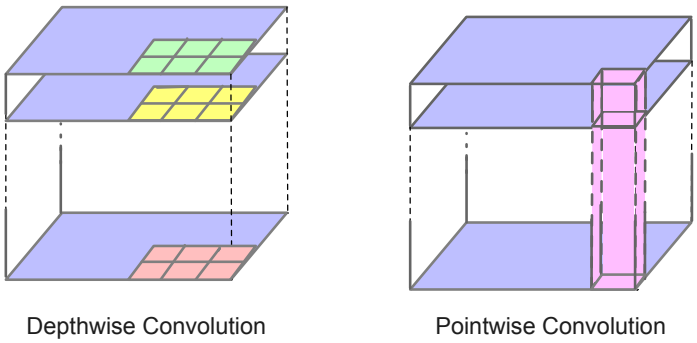

Figure 2.2: Depthwise convolution and pointwise convolution in MobileNet.

2.2 Intersection over Union

Object detection is a crucial task in computer vision that involves identifying and localizing objects within an image or a video. The accuracy of object detection algorithms is often assessed using various metrics, and one key metric that plays a significant role in evaluating the performance of these algorithms is Intersection over Union (IoU). In this section, it is necessary to interpret the concept of IoU, its significance, and its role in assessing the accuracy of object detection systems.

Object detection is a fundamental computer vision task that involves locating and classifying objects within an image or video. It has applications in diverse fields, including autonomous vehicles, surveillance, and medical imaging. Traditional methods for object detection relied on handcrafted features and machine learning algorithms, but recent advancements in deep learning, especially convolutional neural networks, have significantly improved the accuracy of object detection systems.

The performance of object detection algorithms is commonly measured using various evaluation metrics, including precision, recall, and F1 score. However, when dealing with bounding box predictions, where the goal is to precisely localize objects, IoU emerges as

a crucial metric for assessing the overlap between predicted and ground truth bounding boxes.

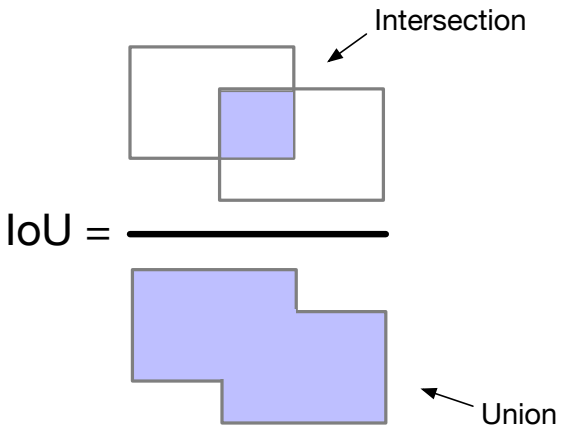

Figure 2.3: The illustration of Intersection over Union.

IoU is a metric used to evaluate the accuracy of object localization in computer vision tasks. It quantifies the overlap between the predicted bounding box and the ground truth bounding box. The IoU is calculated by dividing the area of intersection between the two bounding boxes by the area of their union. As described above, the Intersection over Union is defined as

$$\text{IoU} = \frac{\text{Area}_{\text{Intersection}}}{\text{Area}_{\text{Union}}} \tag{2.1}$$

IoU in Object Detection is usually used in Model Evaluation and Non-Maximum Suppression. IoU serves as a crucial metric for evaluating the performance of object detection models. A high IoU indicates accurate localization, while a low IoU suggests poor alignment between predicted and ground truth bounding boxes. Researchers and practitioners use IoU to compare and select the most suitable object detection algorithms for specific applications. IoU is also a key component in non-maximum suppression (NMS), a post-processing technique used to eliminate redundant and overlapping bounding box predictions. During NMS, bounding boxes with IoU values above a certain threshold are suppressed, retaining only the most accurate and non-overlapping predictions. In conclusion, IoU plays a pivotal role in the evaluation and improvement of object detection algorithms. Its ability to measure the overlap between predicted and ground truth bounding boxes makes it an essential metric for assessing the accuracy of object localization. Researchers continue to explore variations and extensions of IoU to address specific challenges, contributing to the ongoing advancements in the field of computer vision and object detection. As the demand for accurate and efficient object detection systems continues to

grow, IoU remains a cornerstone metric in the pursuit of developing robust and reliable computer vision solutions.

2.3 Non-maximum Suppression

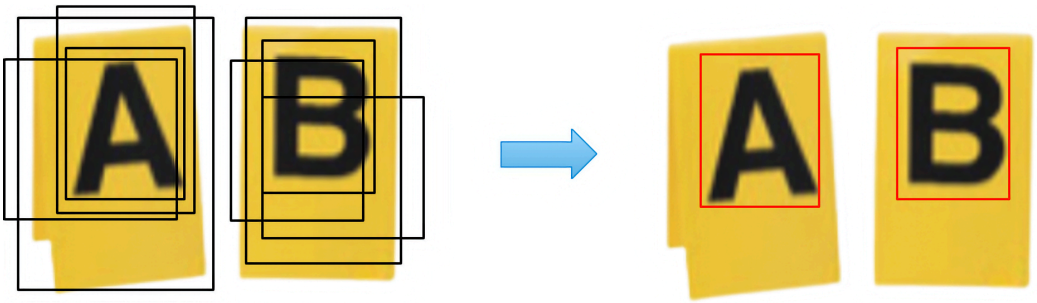

Figure 2.4: The illustration of Non-maximum Suppression.

2.3.1 Principle of NMS

For many target detection methods, the model generates too many detection boxes in the process of detecting targets. Numerous detection boxes make the detection results have a lot of redundancy or non-existent results, which reduces the quality of the prediction results.Non-Maximum Suppression(NMS) is a post-processing technique commonly used in target detection tasks to eliminate the detection boxes with more overlapping, thus improving the quality of the detection results. It mainly solves the following problems:

1. **Multiple Detection boxes Problem**: In a target detection task, it is common to generate many candidate detection boxes, some of which may overlap around the target. This may be due to different scales, different locations or different features. If left unprocessed, the same target may be represented by multiple overlapping detection boxes, leading to inaccurate results.
2. **Remove Redundant Information**: A large number of overlapping boxes introduces redundant information that reduces the interpret-ability and quality of the results, NMS can help to select the most representative boxes to improve the clarity of the results.
3. **Improved Localization Accuracy**: When there are multiple highly overlapping detection boxes, selecting the boxes with the highest confidence improves the accuracy of target location localization.
4. **Suppressing False Positives**: In some cases, the algorithm may produce low-confidence false detection of boxes that may overlap with the true target box. With NMS, these low-confidence false positives can be removed, improving the veracity of the detection results.

The specific process of NMS is as follows:

1. For each detected Bounding Box, calculate its Intersection Over Union (IoU) with all other detected Bounding Boxes.

2. The current Bounding Box is considered to overlap with all Bounding Boxes with which it has a high IoU value (usually greater than a threshold value, e.g. 0.5).

3. Sort all target boxes according to their overlap, usually in ascending order of Confidence Score.

4. Starting with the target box with the highest Confidence Score, do the following in order:

i. If the target box does not overlap with any of the previous target boxes, keep it and continue to process the next target box.

ii. If that target box overlaps with one of the preceding target boxes, calculate the IoU between them, and if the IoU is greater than a certain threshold (e.g., 0.5), set the confidence level of that target box to 0, do not keep it, and continue processing the next target box.

iii. If this target box overlaps with multiple previous target boxes, calculate a weight based on their IoU values and confidence levels, and then reduce the confidence level of this target box based on this weight. Specifically, the confidence of the target box with the largest IoU can be reduced the most, while the confidence of the target box with a smaller IoU can be reduced less.

5. Repeat step 4 until all target boxes have been processed.

2.3.2 Algorithm of NMS

Based on the description of the NMS process in the previous section, it can be simply assumed that NMS removes highly overlapping redundant boxes by comparing IoUs between candidate boxes, thereby obtaining the final detection result. The following is a representation of the most typical NMS algorithm:

The input includes the default box set B and the IoU threshold I_t, and the output includes the final prediction box set P and its corresponding score set S. Formal execution of the algorithm requires the completion of a loop that keeps executing the following steps until the default set of boxes B is empty:

Select the box with the highest score from the current default set of boxes B and set it to M. Add M to the final set of prediction boxes P and then remove M from the set of default boxes B. For each box b_i in the remaining default box set, if the IoU with the selected box M is greater than or equal to the threshold I_t, remove b_i from the default box set B and accordingly remove the corresponding score from the score set S. When the default box set B is empty, the algorithm ends and returns the final set of prediction boxes P and the corresponding set of scores S.

Algorithm 1 NMS Algorithm

Require: B: Default Box Set, I_t: IoU threshold,
Ensure: The final set of prediction boxes P and score S

$P \leftarrow \{\}$
while $B \neq empty$ **do**
  $m \leftarrow argmax(S)$
  $M \leftarrow b_m$
  $P \leftarrow P \cup M,\ B \leftarrow B - M$
  for $b_i\ in\ B$ **do**
    if $iou(M, b_i) \geq I_t$ **then**
      $B \leftarrow B - b_i,\ S \leftarrow S - s_i$
    end if
  end for
end while
return $P,\ S$

2.3.3 Soft-NMS

Soft-NMS is an improved NMS algorithm proposed by Google researcher Navaneeth Bodla et al. in 2017 [2]. Compared to traditional NMS, Soft-NMS adopts a softening strategy to better adapt to scenes with dense and overlapping targets by adjusting the scores of the detected frames instead of direct suppression.The core idea of Soft-NMS is to enable more likely frames to be retained in the final result by gradually decreasing the scores of the frames in the overlapping region during non-extremely large value suppression.

Soft-NMS introduces a decay function by which the scores of the boxes in the overlapping region are gradually reduced. Specifically, for two boxes with overlapping IoU, Soft-NMS calculates a weight through the decay function and multiplies this weight by the original score to obtain the final score. This process allows the box scores to be not directly suppressed, but gradually reduced by the adjustment of the weights. The decay function is a key part of Soft-NMS, which determines the rate at which the box score decreases. Common decay functions include linear functions, Gaussian functions, and so on. The linear function is simple and intuitive, while the Gaussian function is smoother.The flexibility of Soft-NMS allows different decay functions to be selected according to the needs of specific tasks to better adapt to different scenarios. In general, for target detection a Gaussian function is often used, which is:

$$s_i = s_i e^{-\frac{\mathrm{iou}(\mathcal{M}, b_i)^2}{\sigma}}, \forall b_i \notin \mathcal{D} \tag{2.2}$$

Therefore, Soft-NMS receives the intersection of the two target detection candidate boxes and is larger than IoU through the Gaussian function, and makes different degrees

of punishment. Combined with the strength of the punishment, it modifies the confidence of the target object: when the size of the intersection and ratio is zero, there is no punishment; when the intersection and ratio is higher, a greater penalty is given; and when the intersection is lower than the size, the punishment is gradually increased according to the size.

Algorithm 2 Soft-NMS Algorithm

Require: B: Default Box Set, I_t: IoU threshold,
Ensure: The final set of prediction boxes P and score S

$P \leftarrow \{\}$
while $B \neq empty$ **do**
  $m \leftarrow argmax(S)$
  $M \leftarrow b_m$
  $P \leftarrow P \cup M,\ B \leftarrow B - M$
  for $b_i\ in\ B$ **do**
    if $iou(M, b_i) \geq I_t$ **then**
      $B \leftarrow B - b_i,\ S \leftarrow S - s_i$
    end if
    $s_i \leftarrow s_i\ f(\text{iou}(M, b_i))$
  end for
end while
return $P,\ S$

The algorithm implemented in Soft-NMS pseudo-code is basically the same as the traditional NMS algorithm, with the difference of a new line of code. The algorithm combines the attenuation function to modify the score of each default box after each acquisition of candidate boxes, making the whole NMS process softer and easier to select the correct candidate box.

2.4 Summary

This chapter introduces the concepts related to the research of this thesis, mainly including the introduction of Convolutional Neural Networks and their important networks, the concepts and implementation of IoU and NMS. Convolutional Neural Networks (CNNs) are foundational in computer vision, with VGG16 and MobileNet standing out. VGG16, introduced in 2014, is known for its deep architecture, particularly in image classification. MobileNet, presented in 2017, focuses on lightweight design for mobile devices, utilizing depthwise separable convolutions and width multipliers.

Intersection over Union (IoU) is a key metric, assessing bounding box overlap in object detection. Non-Maximum Suppression (NMS) is crucial post-processing, eliminating

redundant bounding boxes. Soft-NMS, an enhancement, introduces a softened weighting mechanism for improved adaptability in scenarios with dense or overlapping objects.

In summary, VGG16 and MobileNet contribute to CNN architecture diversity, while IoU, NMS, and Soft-NMS play pivotal roles in object detection evaluation and refinement.

Chapter 3

Single Shot MultiBox Detector

3.1 Introduction

As discussed in Section 1.1, SSD is a widely used deep learning network for object detection, representing a significant advancement over traditional methods such as R-CNN (Region-based Convolutional Neural Network) and its variants, Fast R-CNN and Faster R-CNN. Prior to the advent of the SSD network, conventional approaches to object detection typically involved two stages: initially generating proposal regions for objects, and then classifying these regions. Specifically, the R-CNN [6], proposed by Girshick et al. in 2014, combined region proposal methods with a CNN, extracting about 2000 candidate regions from an image and then independently processing each region through the CNN for feature extraction, with the final features used for boundary regression. This approach was computationally expensive. In 2015, Girshick introduced Fast R-CNN [5], which incorporated a ROI (Region of Interest) pooling layer, allowing features from region proposals of varying sizes to be transformed into a fixed size. Moreover, it performed CNN processing on the entire image just once, significantly reducing redundant computations. Building on this, Ren introduced the Region Proposal Network [18], which could directly generate region proposals from image feature maps and share convolutional features with the detection network, enabling end-to-end completion of the detection task by the neural network and further enhancing detection efficiency.

These network improvements primarily focused on enhancing the speed and quality of region proposal generation, yet they fundamentally remained two-stage network processes. SSD, on the other hand, handles both region proposal and classification simultaneously through a single network, greatly increasing speed while maintaining accuracy. Furthermore, the multi-scale feature maps used by SSD facilitate the detection of objects of varying sizes, a substantial improvement over the single-scale feature maps of the R-CNN series networks.

In addition, it is worth mentioning the YOLO (You Only Look Once) model, which has

made significant improvements in detection speed. It can quickly predict the object and its position in the image by viewing the image once. Unlike the RCNN series and SSD, YOLO regards the whole image as a whole and detects objects by dividing the grid and predicting multiple boundary boxes and probability scores in each grid. However, the disadvantages of this are also obvious. Based on the image as a whole, it is bound to be not conducive to the detection of small targets, especially text. At the same time, the text has the characteristics of close connection, which will also cause YOLO to be less accurate than the region-based method when dealing with such areas that are very close or even slightly overlapping. SSD can provide more balanced accuracy while ensuring faster speed, which is more suitable for character or text detection.

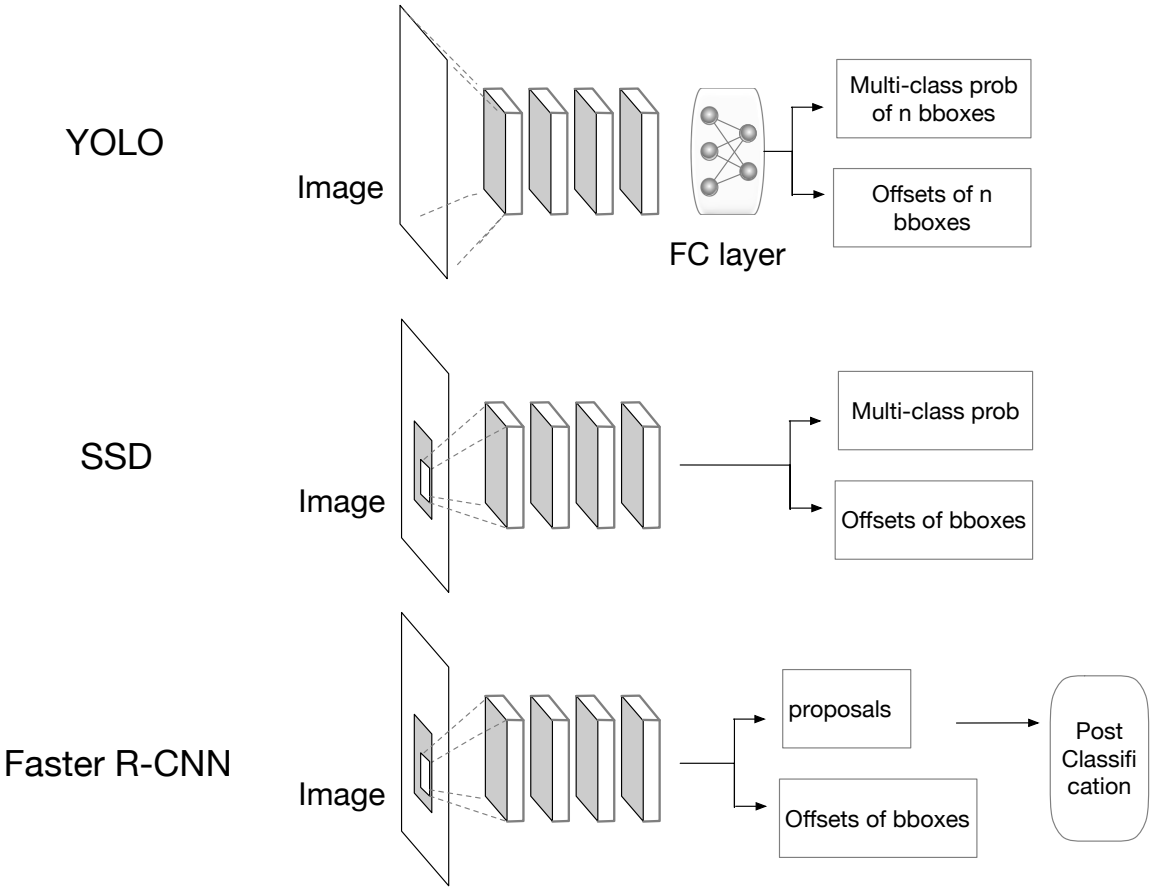

Figure 3.1: Structural difference between SSD and YOLO, Faster R-CNN.

All in all, SSD is a neural network architecture for target detection that can simultaneously predict the location and class of multiple targets in a single forward propagation. Compared to other detection algorithms, the R-CNN series is slow and YOLO's mAP performance is poor, while SSD eliminates the intermediate process of bounding boxes, pixel

or feature resampling, and at the same time, uses a small convolutional kernel on the feature map to predict the box offsets of a series of At the same time, a small convolutional kernel is used on the feature map to predict a series of box offsets of bounding boxes, which improves the speed and guarantees the accuracy at the same time.

3.2 Network Structure

The SSD network consists of several components, including a backbone network, extra feature layers, prediction layer, and a post-processing. As shown in Fig.3.2, the SSD is modified from VGG16 by adding six additional convolutional layers after the $Conv4\_3$ convolutional layer at the end: $Conv6$, $Conv7$, $Conv8\_2$, $Conv9\_2$, $Conv10\_2$, and $Conv11\_2$. These convolutional layers are used to extract higher-level features. These features are used to predict the position and category of the target object. Each convolutional layer predicts candidate boxes (Anchors in R-CNN, YOLO) of different sizes and proportions, thereby achieving the detection of target objects of different sizes and shapes.

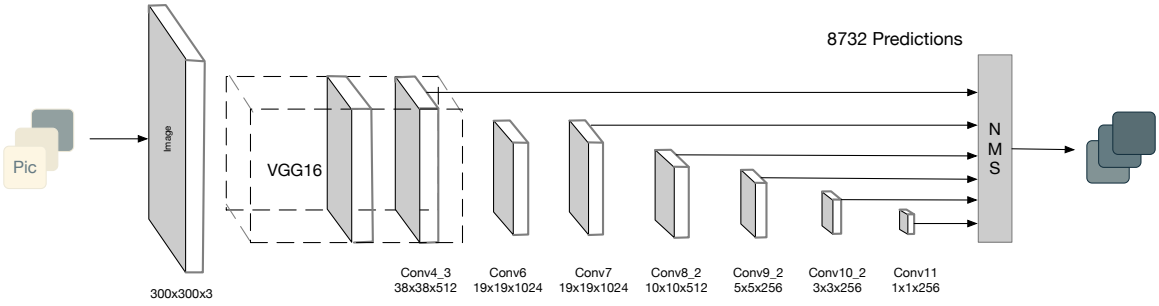

Figure 3.2: The structure of Single Shot MultiBox Detector.

3.2.1 Backbone

The backbone network of the SSD, which is also the base feature extraction network of the SSD, uses the classical VGG16 network, but with adaptive modifications. Among them, the original fully connected layer of VGG16 is converted into a convolutional layer, which conveniently maintains the spatial information and on the other hand reduces the number of parameters. And for the input image dimension is fixed value, on the default setting of 300*300. At this time, the dimension of the feature graph corresponding to $Conv4\_3$ becomes 38*38.

Additionally in SSD, the backbone network uses the VGG16 model This model was used as the base feature extractor while some modifications were made to adapt it to the object detection task. conv6 layer uses the Dilated Convolutions, which has the following benefits:

1. **Increase receptive field**: By introducing additional spatial intervals (i.e. expansion rate) into the standard convolution, dilated convolution can increase the receptor field of the convolution layer without reducing the resolution of the feature map. This means that the network can observe a wider image context, which is very useful for understanding a wider range of objects.

$$F = 2 * (Rate_{\text{dilation}} - 1) * (Size_{\text{kernel}} - 1) + Size_{\text{kernel}} \tag{3.1}$$

It can be seen from the formula that the number of parameters of the convolution kernel remains unchanged, and the size of the receptive field increases exponentially with the increase of the "Dilation Rate" parameter.

2. **Improve computing efficiency**: The use of dilated convolution can expand the receptive field without adding additional parameters, which can improve the performance of the model without significantly increasing the computational burden.

3. **Keep the resolution of the features**: For object detection, especially for small objects such as text characters, it is necessary to maintain sufficient feature resolution. Dilated convolution allows the network to maintain a high feature map resolution, which is very important for locating small objects.

4. **Avoid over-pooling**: The design of VGG16 includes multiple pooling layers, which will gradually reduce the spatial resolution of the feature map. In object detection tasks, especially when multi-size objects need to be detected, excessive pooling will lead to the loss of details. By using dilated convolution, deeper feature extraction can be carried out on the network without further reducing the resolution.

The use of dilated convolution in SSD improves the SSD network's ability to understand large-scale contexts without losing feature map resolution, while maintaining model parameters and computational efficiency. This enables the model to effectively detect objects of different sizes, thereby maintaining high detection accuracy while maintaining high speed.

3.2.2 Extra feature layers

After the basic network, the SSD introduces multiple additional convolution layers. These layers produce a series of feature maps with decreasing sizes, each of which is used to detect objects of different sizes. For the SSD300 model (which means the input image's size has been set 300x300 pixels), the output of the additional feature layer is usually as follows:

- The first layer ($conv8\_2$) outputs feature maps with a size of $512 \times 10 \times 10$.
- The second layer ($conv9\_2$) outputs feature maps with a size of $256 \times 5 \times 5$.
- The third layer ($conv10\_2$) outputs feature maps with a size of $256 \times 3 \times 3$.

- The last layer ($conv11\_2$) outputs a feature map with a size of $256 \times 1 \times 1$.

Each layer not only reduces in size, but also increases in depth. As the network continues to deepen, feature maps will increasingly focus on global information rather than local details.

This 4 feature layers obtained from this extra feature layer, together with the feature maps obtained from $Conv4\_3$ and $Conv7$ in the backbone part, extracted a total of 6 feature maps with sizes of (38, 38), (19, 19), (10, 10), (5, 5), (3, 3), (1, 1), which the network feeds into the the prediction section for target detection.

3.2.3 Prediction

The network is based on the 6 feature maps extracted by backbone and Extras in the extraction of location and classification information, and its structure is shown in Figure 3.3:

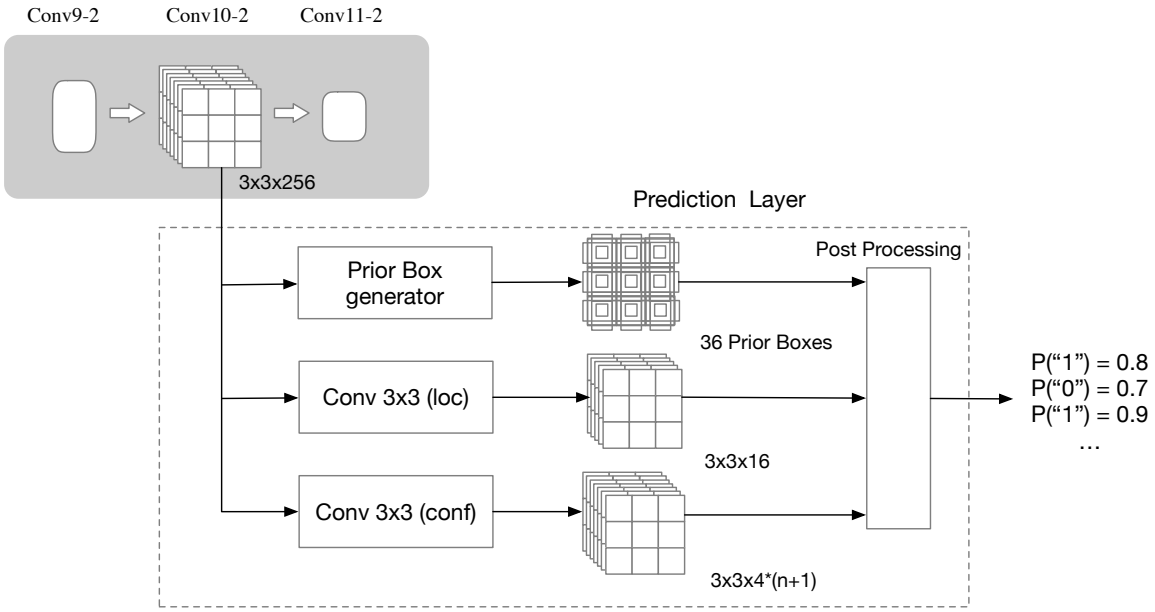

Figure 3.3: Prediction process in SSD.

This part is mainly composed of three branches:The PriorBox layer, Location and Classification confidence Layer. The PriorBox layer is used to generate prior boxes, also known as anchor boxes in faster R-CNN. Taking the feature layer $conv10\_2$ contained in the extras section of the SSD as an example, assuming that the number of prior boxes per unit is 4, the feature layer has a total of 36 prior boxes ($4 \times 3 \times 3$).

For Location Layer, it is completed using a 3×3 convolution. Each a prior box has four coordinates and a total of 144 prediction results ($3 \times 3 \times 4 \times 4$).

Finally, for Classification confidence Layer, also done using 3x3 convolution, assuming that the dataset has n classes of objects, there are $n + 1$ predictions corresponding to each prior box, for a total of $36n + 36$ predictions ($3 \times 3 \times 4 \times (n + 1)$).

3.2.4 Post-Processing

In the SSD model, a standard NMS algorithm is used in the post-processing part to screen out redundant detection boxes and accomplish the object detection task. Significantly, as an anchor-based single-stage detection model like SSD is itself densely prediction-based, the NMS does not need to be participated in the process of prediction, which means that it is excluded from the training of the model. With this post-processing process, the amount of computation can be significantly reduced and the performance of the model can be improved.

In practical task, basically top_k is used to indicate the number of high confidence score boxes which are limited by the number and low confidence socre boxes are discarded as they have no processing value.The NMS algorithm utilizes these high confidence score boxes to compute the IoU and compares them carefully with the threshold overlap value to remove the duplicate boxes. After the above multiple operations, the final $keep$ list is obtained to store the boxes which are the bounding boxes shown in the image.

3.3 Mechanism

3.3.1 Box generation in PriorBox layer

SSD has a total of 6 feature maps of different scales, each feature map is set on a different number of a prior boxes (the same feature map is set on each unit of the same a prior boxes, and the number refers to the number of a prior boxes of a unit). Assuming that m feature maps are predicted, the default box scale for each feature map is calculated as follows:

$$s_k = s_{min} + \frac{s_{max} - s_{min}}{m - 1}(k - 1), k \in [1, m] \tag{3.2}$$

In this equation, the scale of the lowest layer s_{min} is 0.2, the scale of the highest layer s_{max} is 0.9, and the scale of the intermediate layers is obtained by equally spaced sampling. m is the number of feature maps, and s denotes the ratio of the default border size to the image size.

Each convolution layer outputs a certain number of prior boxes (Default Boxes), which have different sizes and aspect ratios. For each prior box, the SSD predicts its IoU with respect to the real target object, as well as the category and coordinate offset of the target object. These predictions are used to generate the final target detection results. This process requires the use of the NMS to process the resulting prediction boxes.

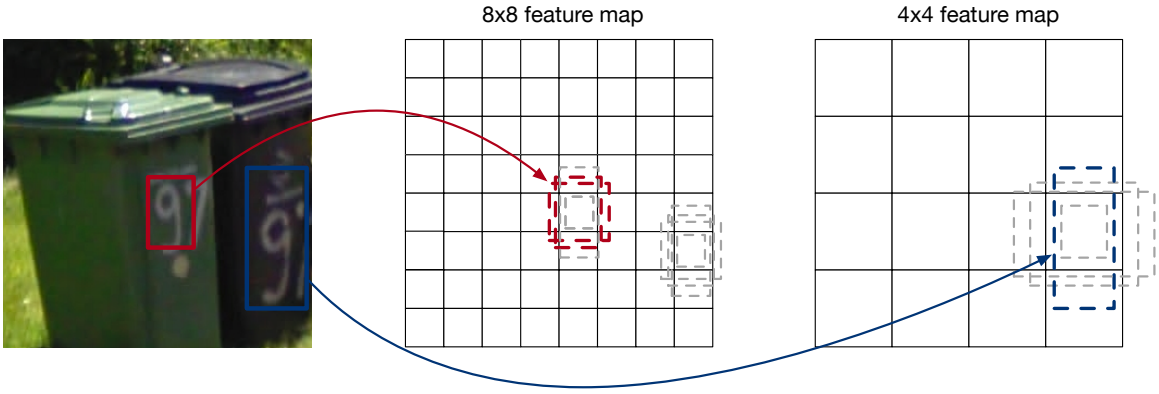

Figure 3.4: Default boxes for different feature maps for SSD.

In addition, after the $Conv4\_3$ layer of VGG16, SSD adds an L2 Normalization layer, which is designed to solve the problem that when the output of the $Conv4\_3$ layer is directly used as the input, large feature values may have a large influence on the subsequent output. With L2 Normalization, all feature values can be made to have the same influence, avoiding large feature values from having too much influence on the output results.

Table 3.1: Network outputs and anchors' number of SSD300 / SSD512

	SSD300			SSD512		
Name	Output Size	Default Boxes	Num	Output Size	Default Boxes	Num
conv4-3	38x38	4	5776	64x64	4	16384
conv7	19x19	6	2166	32x32	6	6144
conv8-2	10x10	6	600	16x16	6	1536
conv9-2	5x5	6	150	8x8	6	384
conv10-2	3x3	4	36	4x4	4	64
conv11-2	1x1	4	4	2x2	4	16
Total			8732			24528

3.3.2 Model training

In the training process of the SSD model, it is first necessary to determine which Default box matches the ground truth (GT) in the training image, and the bounding box corresponding to the matched Default box will be responsible for predicting the corresponding ground truth. For positive samples, find the Default box with the highest IOU for each gt in the image. The Default box is matched with it first, and if the remaining unmatched Default

boxes are greater than a certain threshold (usually 0.5), they are also matched with the GT. All remaining Default boxes are marked as negative samples.

To improve the accuracy of the model, SSD also uses the technique of hard sample mining. The idea is to use the network to process the samples and put the hard negative samples into the set of negative samples before continuing to train the network model.Specifically for the SSD model, there are steps like this:

1. Train the network using a 1:3 ratio of positive to negative samples.

2. Sort the input prediction results in descending order by class confidence, and take out the first k negative samples.

3. Add these k negative samples to the negative samples for the next iteration to train the network.

3.4 Loss Function

The SSD loss function is divided into two parts: the location loss (L_{loc}) corresponding to the predicted box and the classification confidence loss (L_{conf}). The overall loss function is a weighted sum of the two:

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g)) \tag{3.3}$$

where N is the number of positive samples in the default boxes, c is the predicted value of the classification confidence, l is the predicted value of the position of the corresponding bounding box of the default box, and g is the positional parameter of the GT, with the weighting coefficient α set to 1.

In the location loss function, Smooth L1 Loss is used for all positive samples:

$$L_{loc}(x, l, g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^{k} \operatorname{Smooth}_{\mathrm{L1}}\left(l_i^m - \hat{g}_j^m\right) \tag{3.4}$$

$$\operatorname{Smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases} \tag{3.5}$$

x_{ij}^k represents 1 when the i-th default boxes are specified to predict the k-th class object of the j-th true bounding box, otherwise it is 0. It is used to determine which default boxes are assigned to a GT, and only the errors of these assigned boxes will be included in the loss function. l_i^m represents the parameters of the predicted bounding box, and m represents different parameters (cx, cy of the center point, width w, height h). $\hat{g}_j^m$ denotes the parameters of the GT that match the predicted box.

The classification confidence loss function (L_{conf}) is used to measure the difference between the predicted confidence level of the model and the GT. It consists of two parts, one

deals with positive samples (Pos, which is the predicted box containing objects), and the other deals with negative samples (Neg, which is the predicted box containing only the background but not the objects).

$$L_{conf}(x, c) = -\sum_{i \in Pos}^{N} x_{ij}^{p} \log\left(\hat{c}_i^p\right) - \sum_{i \in Neg} \log\left(\hat{c}_i^0\right) \quad \text{where} \quad \hat{c}_i^p = \frac{\exp\left(c_i^p\right)}{\sum_p \exp\left(c_i^p\right)} \tag{3.6}$$

For positive samples, if the predicted box i matches the GT j with respect to category p, the higher the probability of predicting p, the smaller the loss. In the negative sample part, the prediction box actually does not have an object, so the higher the probability of predicting it as the background, the smaller the loss. The final $\hat{c}_i^p$ uses the soft-max function to convert the model's original score c_i^p for the i-th prediction box belonging to each category p into a normalized probability distribution.

3.5 Summary

This chapter focuses on the basic model used in the study. SSD is different from YOLO and Faster RCNN. Through its unique multi-layer prediction network structure and loss function design, it can quickly and accurately detect multiple types of objects in images. The network structure of SSD starts with pre-trained basic convolutional networks represented by VGG, which are used to extract low-level image features. On this basis, SSD adds a series of additional convolutional layers, which generate a series of feature maps with decreasing sizes. The obtained features are trained and the final detection results are obtained through post-processing processes such as NMS.

Another key feature of the SSD model is its use of the PriorBox generation mechanism. On each feature map cell, the model predicts a series of prior boxes of fixed size and scale. These anchor boxes serve as candidate regions for object detection, and the model matches the bounding boxes of real objects by adjusting the position and size of these boxes.

The loss function plays a crucial role in the optimization process of SSD. It consists of two parts: location loss and classification confidence loss. Location loss focuses on the accuracy of predicting bounding boxes, ensuring that the predicted boxes are as close as possible to the position of the real object. The classification confidence loss ensures that the model can accurately classify the objects within each prediction box, including correctly classifying boxes without objects as backgrounds.

In the next chapter, this study focuses on the post-processing part of SSD and the improvement of the basic network structure to achieve character detection in text.

Chapter 4

Proposed Method

4.1 Network Structure

This study mainly focuses on modifying post-processing processes such as NMS to use textual feature information. However, for the network structure, this study still optimized and adjusted the default SSD model structure to some extent.

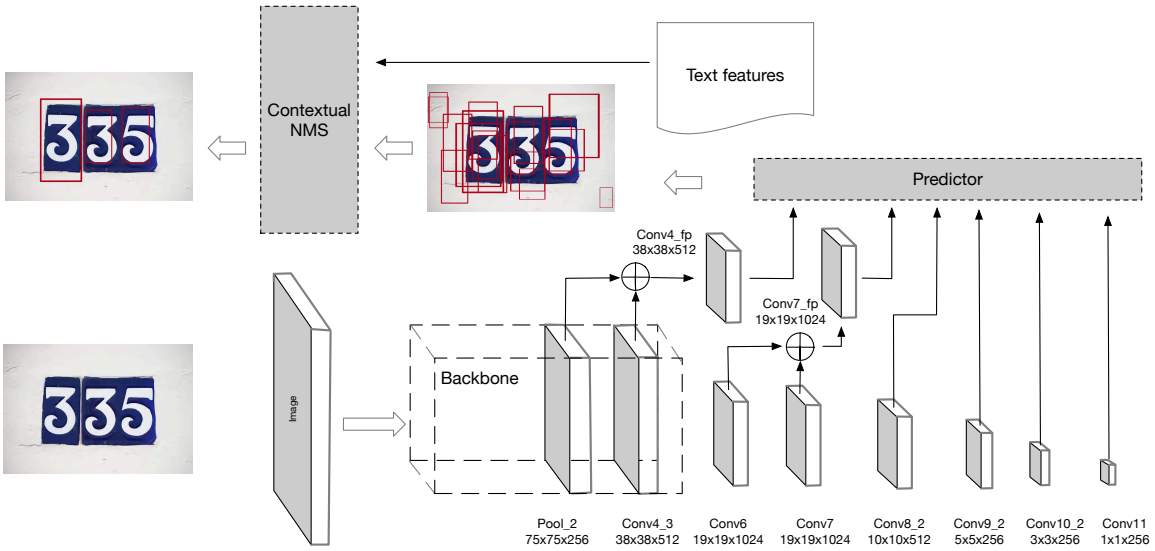

Figure 4.1: The structure of Single Shot MultiBox Detector.

From the detailed introduction of the SSD model structure in Chapter 3, it can be seen that the backbone network can be used not only including VGG16, based on its strong flexibility, which allows the choice of other different networks according to the needs of the scene. Scene text detection is usually used on portable platforms such as cell phones, so in addition to the original VGG16 backbone network, MobileNet is also added as a

lightweight network option.

For the rest of the network, based on the property of SSD to handle multi-scale target detection, the strategy of feature fusion can be used in the latter part of the backbone network and in the extra layers [14]. For this purpose, we design a simple feature fusion module and use the newly obtained feature layer fusing the upper information for the detector in the subsequent steps.

The process described above is essentially a routine modification of the network, with generalized performance-enhancing capabilities for various target detection, but not a means of targeting textual features. For textual features, in addition to the new feature layer can be utilized individually for some features of the text, but it is likewise possible to introduce text-specific information in the post-processing portion, which likewise obtains certain results in evaluation and actual detection. Therefore, a new Contextual NMS was designed to replace the original default NMS and applied to this network structure.

4.2 Feature Fusion Module

In the field of computer vision, Feature Pyramid Networks (FPN) is an effective multi-scale feature fusion method widely used in object detection tasks. FPN mainly combines feature maps of different resolutions to enable the network to simultaneously possess high-level semantic information and low-level detail information, thereby improving the detection ability of targets of different scales. This type of method performs well in dealing with scenes with multi-scale and objects of different sizes, especially in the detection of small objects.

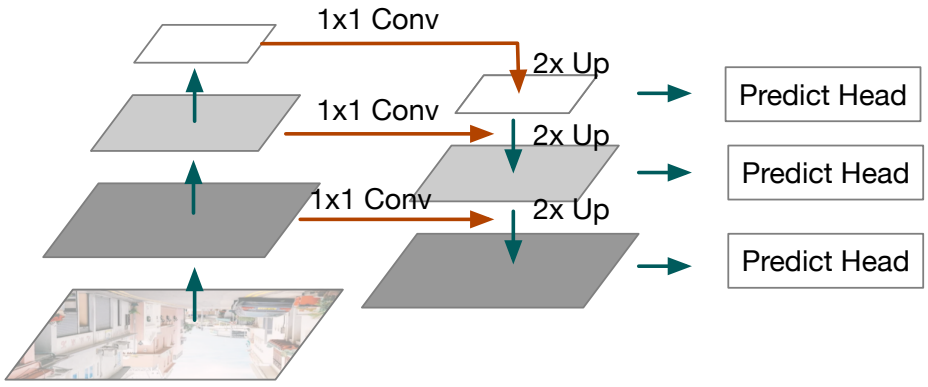

Figure 4.2: The structure of Single Shot MultiBox Detector.

In this study, we improved the classic SSD300 model by introducing a feature fusion module, aiming to enhance the model's detection ability for multi-scale targets, especially in character detection tasks in complex backgrounds. Our improvement is based on the following key steps:

1. **Selection and adjustment of feature layers**: We first selected several key feature layers in the SSD network, including $conv4\_3$ and $Conv\_7$. These layers provide a basis for fusing features of different scales due to their different receptive fields and levels. To make these feature layers more suitable for fusion, we adjusted them, including unification of channel numbers and matching of spatial dimensions. The goal of this method is to obtain $conv4\_fp$ and $conv7\_fp$, which maintain the same number of channels and space size as $Conv4\_3$ and $Conv\_7$.

2. **Feature context fusion**: Operations such as dilation convolution and transposition convolution are used to realize the scale transformation of the feature map. In this way, we retain the rich semantic information in the high-level feature maps and obtain the detailed information in the low-level feature maps, realizing the effective fusion of features.

3. **Smoothing**: After feature fusion, we introduce a Smoothing Layer (Convolutional Layer) to minimize sharp changes or discontinuities that may occur during the fusion process. This step helps to generate smoother and more coherent feature maps, thus improving the detection performance.

Such a feature fusion strategy brings significant benefits to the SSD model. First, by fusing different levels of features, the model is able to detect targets at different scales more effectively, especially small targets in complex scenes. Second, as the feature map fuses more semantic and detail information, the model has a more comprehensive characterization of the target, which is especially important for the character detection task, as characters tend to have small sizes and various patterns. In addition, feature fusion enhances the model's resistance to background interference, further improving the accuracy and robustness of character detection.

4.3 Contextual NMS

From the introduction in Section 2.3, we can see that the main function of NMS is to screen out prediction boxes with high confidence and improve the detection accuracy of the model. Nevertheless, the default NMS only takes into account the degree of overlap between a given prediction box and its corresponding truth value, IoU, and does not take into account the property that the text within the text is closely connected and the text size remains relatively stable. As a result, inaccurate position determination may occur in places with dense text, or false detection may occur in non-text areas. For this reason, we have designed a new NMS algorithm to import the properties of the text in the text and realize to get more accurate results.

In NMS, the algorithm first needs to prioritize the good boxes based on the $score$ value of each default box. SSD uses the default NMS's score simply indicates the classification confidence of each default box.For actual character detection, since our task mainly lies in the process of relative optimization for text features, without focusing on the number of categories of characters, we can add the distance score $score_{dist}$ of neighboring characters

and the size score $score_{size}$ of the size of the text on the basis of the original class score $score_{cls}$.

$$\text{score}_{\text{cont}} = \alpha * \text{score}_{\text{cls}} + \beta * \text{score}_{\text{dist}} + (1 - \alpha - \beta) * \text{score}_{\text{size}} \tag{4.1}$$

In the above equation, we weakened the original position score and adjusted the ratio of the increased distance score to the size score by two parameters, α and β. The values of them should be set in the range of 0 to 1. For Contextual NMS, we first try the simplest way to get the distance and size scores for the corresponding characters, as listed below:

1. To sort all the predicted boxes according to the position of their centers.
2. Add $width$ and $height$ information to all predicted boxes.
3. For the predicted box $Pred\_box[i]$, calculate the $dist$ from the next predicted box $Pred\_box[i+1]$. If $dist < width$, $dist$ uses the new distance from $Pred\_box[i+2]$, and so on.
4. Calculate the score of the predicted boxes using the contextual information.

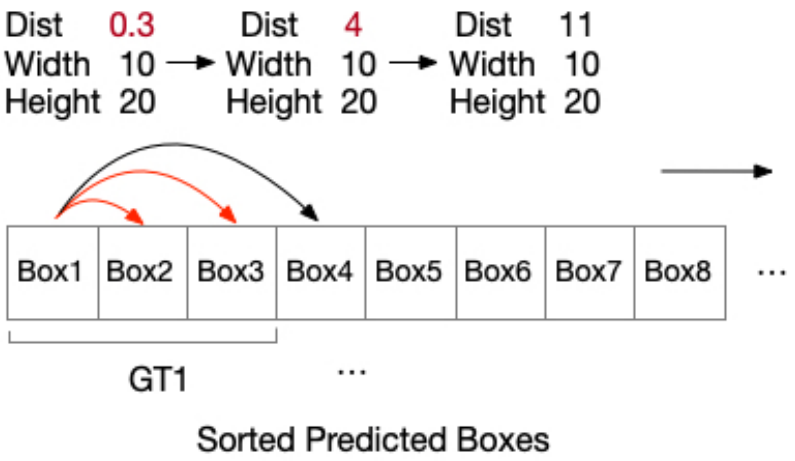

Figure 4.3: The selection of $dist$ in Contextual NMS

After the above pre-processing of the new score, due to the increase in the number of evaluation indicators, the original NMS algorithm can not be effective in determining the correct prediction boxes, the need for NMS algorithms also make some changes. For Contextual NMS, new threshold settings need to be added, as well as a good prioritization order for scores. The following is the exact procedure:

1. Sort the predicted boxes using the new score and pick the largest one as good box.
2. Calculates the overlap and context information of this box with other boxes.
3. Remove predicted boxes based on IoU threshold and contextual information. Basically, it requires the value of IoU to be greater than the default threshold, as well as $dist$ to be less than the specified threshold.

Algorithm 3 Proposed NMS Algorithm

Require: B: Default Box Set, I_t: IoU threshold, D_t: Distance threshold
Ensure: The final set of prediction boxes P and score S

$B \leftarrow Sort(B)$
$P \leftarrow \{\}$
while $B \neq empty$ **do**
  $m \leftarrow argmax(S)$
  $M \leftarrow b_m$
  $P \leftarrow P \cup M,\ B \leftarrow B - M$
  for $b_i\ in\ B$ **do**
    if $iou(M, b_i) \geq I_t$ **and** $dist(M, b_i, b_{i+1}) \leq D_t$ **then**
      $B \leftarrow B - b_i,\ S \leftarrow S - s_i$
    end if
  end for
end while
return $P,\ S$

4. Repeat the above steps for the remaining predicted boxes until all good boxes are selected.

4.4 Summary

In this study, we made simple modifications to the network study based on the specific requirements of the text detection task. This is mainly reflected in the use of two backbone networks using VGG16 and MobileNet, as well as the further improvement of the model's detection capability by designing a simple feature fusion module. In the post-processing part of the SSD, instead of the default NMS, we designed an NMS that uses text feature information to add a threshold of text distance information to the original IoU threshold, which works together to pick the correct bounding box, thus optimizing the accuracy of the model to finally detect the characters present in the text.

Chapter 5

Experience

5.1 Preparation

5.1.1 Dataset

The Street View House Numbers (SVHN) dataset used in this study, derived from the real-world house number of Google Street View Service, is a recognized standard data set in the field of target detection. The importance of this data set is that it reflects the complexity and diversity of digital recognition in the real world, so it has become an important foundation for text character detection research.

The SVHN dataset contains over 100000 color images, divided into 73257 training images and 26032 test images, in addition to 531131 additional images for deeper training. These images display various lighting conditions, backgrounds, and font styles, including not only individual numbers but also combinations of multiple digits, greatly enhancing the complexity and application value of the dataset. In terms of data format, SVHN images are RGB three channel color images with various resolutions. Each image is equipped with detailed annotation information, indicating the specific position and value of each number in the image. To enhance the generalization ability and robustness of the model, it is generally necessary to perform a series of prepossessing operations on the dataset, such as image size adjustment, normalization, and data augmentation operations (such as rotation and scaling).

The application of SVHN in this experiment can preliminary evaluate the effect of the SSD model on the processing of text characters under real-world conditions before and after modification. Compared with ICDAR, COCO-Text and other datasets specially designed for scene text recognition tasks, using this data set experiment is less difficult to achieve, and it is more meaningful than handwritten data sets such as MNIST-like. In addition, SVHN has two formats, Format 1 is all numbers on the complete image, similar to the real scene character recognition, while Format 2, like many MNIST-like datasets, is the

recognition of numbers on images that are well cropped to the target. Format 1 is more practical and is therefore used in this study.

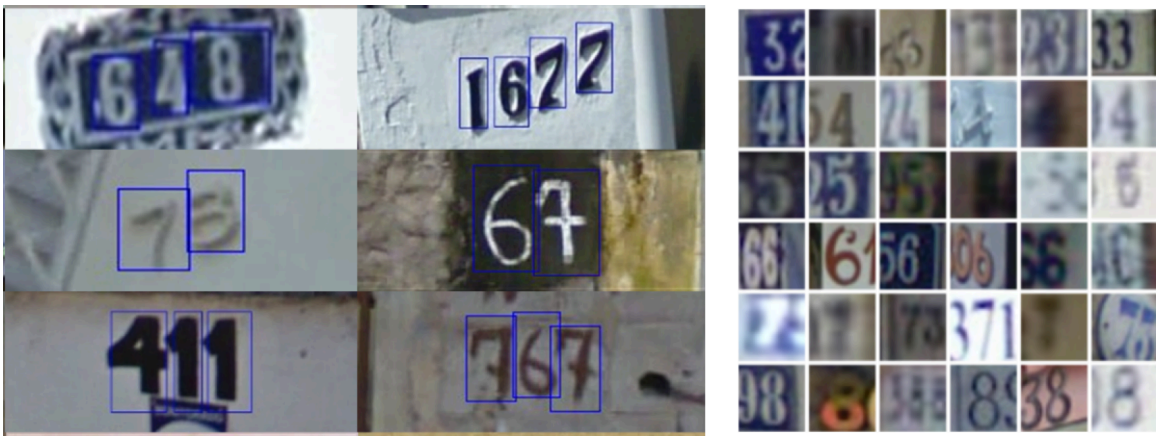

Figure 5.1: The Street View House Numbers (SVHN) dataset. The left image (Format 1) contains a complete street view image, and there will be character level annotations for multiple house numbers in one image. The right image (Format 2) is a cut image of house numbers, similar to MNIST.

5.1.2 Experimental environment

This study is based on the SSD network model, which aims to improve the accuracy and efficiency of class text character detection through the NMS process of post-processing in the model. The choice of SSD model is due to its structural simplicity and excellent real-time detection ability. In order to meet the needs of different scenarios, this experiment adopts two different backbone architectures, VGG16 and MobileNetV2. Among them, the VGG16 architecture performs well in feature extraction with its deep continuous convolution structure, while MobileNetV2 has significant advantages in processing speed with its lightweight and efficiency advantages. The two backbones focus on accuracy or ensure real-time processing efficiency respectively. This strategy makes SSD as a regression target detection of boundary boxes, which is more practical than image segmentation methods.

The experimental hardware environment is configured with NVIDIA GeForce GTX 1080 graphics card, which is equipped with 8GB video memory to meet the needs of computing resources in the training and reasoning process of such a low-depth deep learning model as SSD. The CPU is an Intel Core i7-8700 with 8 cores and 16 threads, and the memory is 32GB, which meets the data processing capabilities required for the experiment. For the software environment, the experiment was conducted under the Ubuntu 23.04 LTS version, using the more efficient PyTorch2.0 deep learning framework for training.

The original format of the SVHN data set is significantly different from the data set format (such as PASCAL VOC) commonly used in the SSD model, mainly in the repre-

sentation of annotation information. Therefore, it is necessary to process the format of the annotation information in advance. The original annotation information of the SVHN data set is provided in Matlab format, which includes the position (border coordinates) of each digit in each image and the corresponding value label. SSD models generally use annotations in XML format similar to VOC. XML annotation files include basic image information: such as file name, size, etc. And annotation information: for the border coordinates of each number in the image (usually the four vertex coordinates of the bounding box) and the corresponding category label (the numeric value of the object). In order to reflect the effect of this improved method, the data set category is set to 2 classes, digit-class and background-class.

5.2 Training

In order to verify the effect of the improved NMS, the model needs to be trained in advance. As mentioned in the previous subsection, we used a local host containing an NVIDIA Ge-force GTX 1080 graphics card to train the three model setups containing SSD300, SSD512, and the refined and simplified SSD300-MobileNet.

The number of epoch was set to 200, Batch size to 32, Learning rate to $6 * 10^{-4}$, and Optimizer to Adam. Default NMS was used during training, where nms_topk was specified as 200, and the IoU threshold was set to 0.25. For the Contextual NMS used for evaluating the results, the hyper-parameters α and β are set to 0.4 and 0.3, respectively, and the distance threshold is set to 0.2 based on experience. In addition, data enhancement processes such as cropping, rotating, and scaling were performed for the images in the SVHN dataset. As shown in Figure 5.2 and 5.3, training is in a smooth state and the 3 models are in the usable state.

It is clear from the figure that the model using MobileNet as the backbone network was trained more quickly, and at the same time the model fit was reached very quickly. For the comparison between SSD300 and SSD512, it can also be seen that in the case of epoch 100, the training of SSD512 may not be sufficiently completed or due to the increase in model parameters, it is less suitable for this simpler dataset, which leads to a high loss in its evaluation.

5.3 Visualization Results

In order to visually see the impact of improved NMS on the results, we visualized the predicted bounding boxes of the model before and after using NMS during the evaluation of AP@50. The visualization results are based on three different SSD models: the SSD300-VGG16, the SSD512-VGG16, and the SSD300-MobileNetV2.For the visualization test images, we chose two images from the validation set in the original SVHN dataset. They

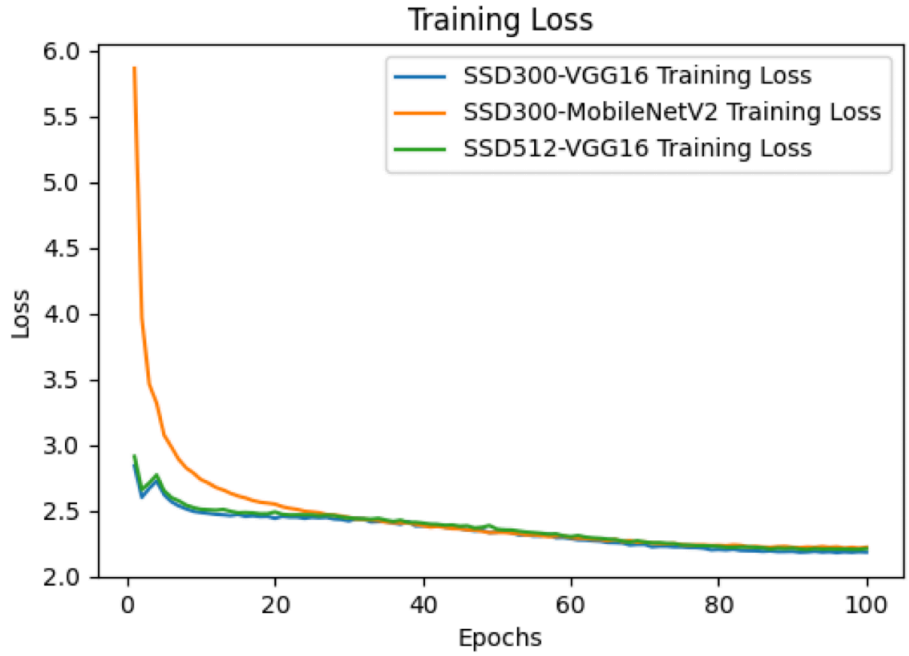

Figure 5.2: Training loss of multiple SSD-derived models

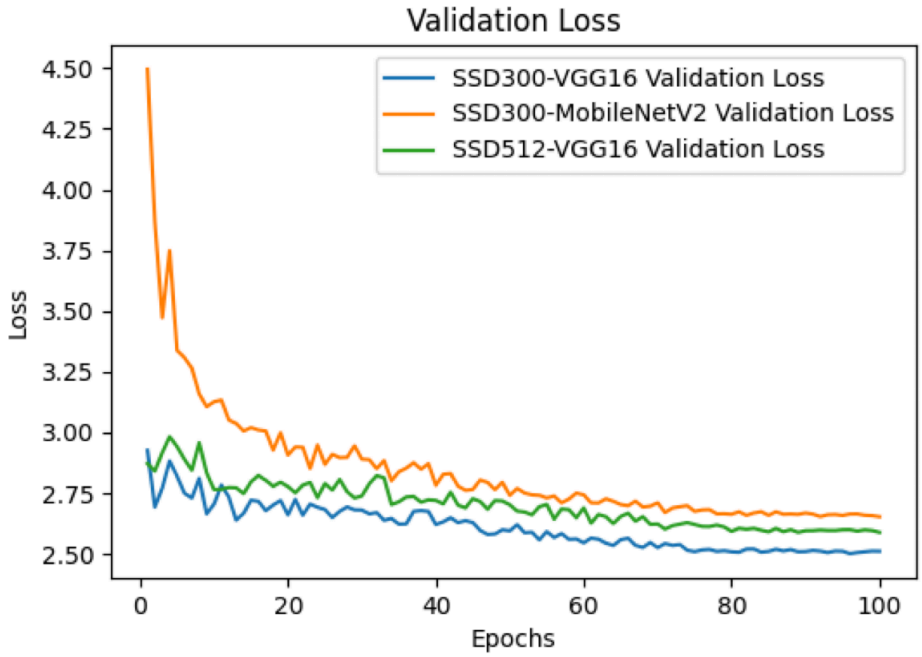

Figure 5.3: Validation loss of multiple SSD-derived models

provide the simplest visualization of the model's and NMS's ability to filter bounding boxes for single characters as well as in the presence of multiple characters.

As shown in Figure 5.3, from left to right, all the obtained bounding boxes before the NMS processing, the good bounding boxes obtained after the default NMS processing, and the good bounding boxes obtained after our improvement are shown in order as the model is evaluated. In order to visualize the results more clearly, the number of bounding boxes generated in the series of images was set at a certain value. the upper limit of the number of bounding boxes was set at 300 for the SSD300 series of models, while it was set at 1000 for the SSD512 model.

Comparing the results of the two NMSs, it is obvious that our improved Contextual NMS can effectively utilize the spacing information of the numbers in a string (e.g., the picture with "85") as well as the shape characteristics of the numbers themselves. Compared to the default NMS, it removes most of the invalid bounding boxes and basically retains the high confidence bounding boxes around the numbers.

A longitudinal comparison of the three models shows that the models are more conducive to capturing more pre-selected bounding boxes as the scale increases or the number of network layers deepens. In such a case, the default NMS's bounding box selection mechanism likewise filters out more good bounding boxes, while the improved Contextual NMS's ability to select the correct bounding boxes is further demonstrated.

5.4 Detection Results

When evaluating an object detection model, similar to SSD, a series of standardized indicators are usually used to measure its performance. These indicators include Precision, Recall, F1 Score, and Mean Average Precision (mAP).

Precision is the proportion of predicted positive classes (such as detected objects) that are actually positive, like $\frac{TP}{TP+FP}$. Where TP (True Positives) is the number of correctly detected objects and FP (False Positives) is the number of objects incorrectly labeled. High accuracy means fewer false positives. Recall is the proportion of positive classes that are correctly predicted as positive classes, like $\frac{TP}{TP+FN}$. FN (False Negatives) is the actual number of objects not detected. High recall means that the model is able to find most of the real objects. The F1 score is the harmonic mean of precision and recall, used to consider both precision and recall simultaneously. MAP is the average accuracy at different recall levels, providing a comprehensive evaluation of the overall performance of the model. When calculating the AP for each category, consideration is given to the accuracy and recall at different confidence thresholds.

This section utilizes the SSD300-VGG16 as a representative to test the detection results of five sets of cases with AP values of 0.25, 0.4, 0.5, 0.6, and 0.75. Among them, Contextual NMS is our proposed scheme, while Contextual NMS+ is a scheme that incorporates Soft-NMS based on Contextual NMS.

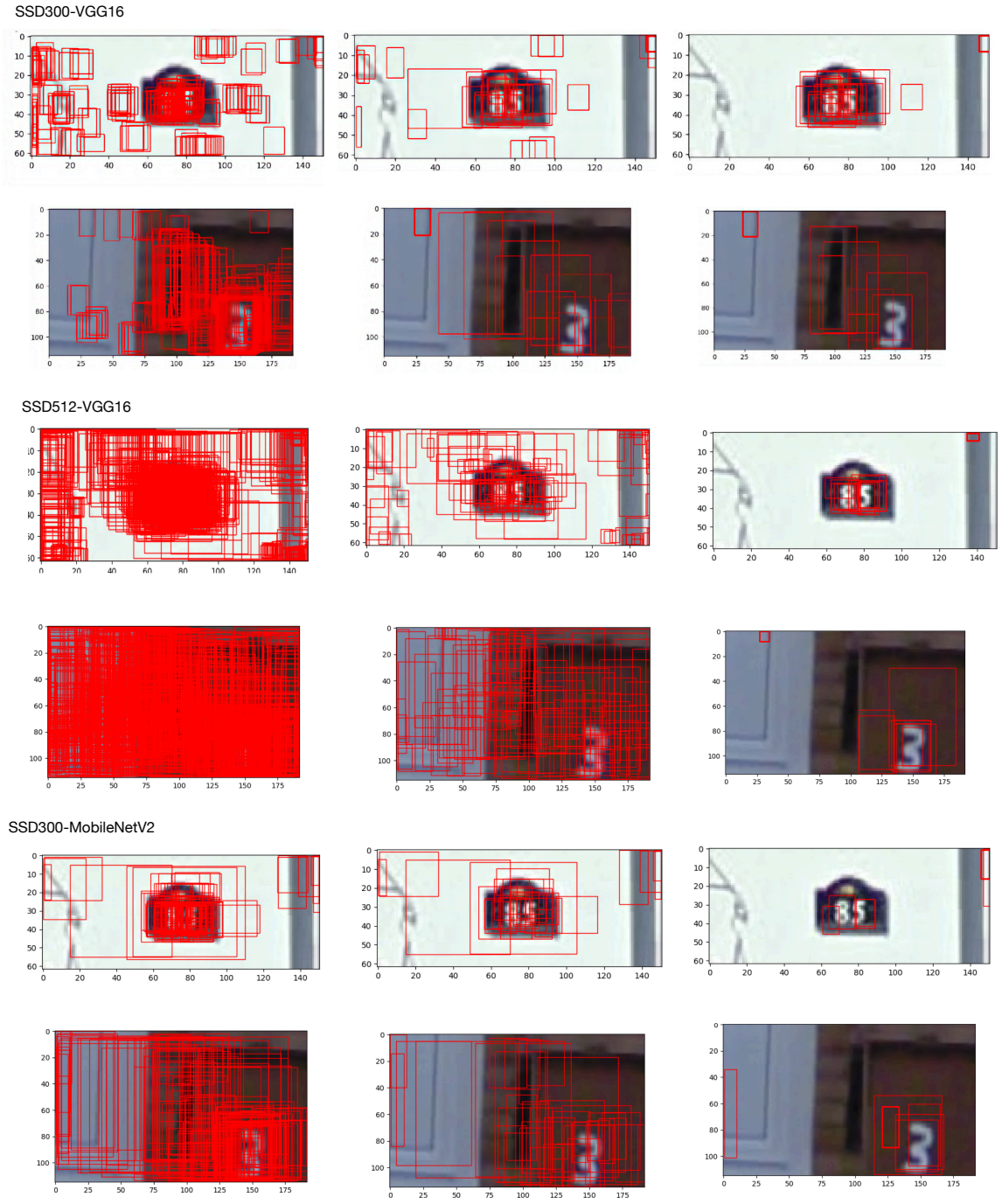

Figure 5.4: The image on the left represents all the prediction boxes(Default Boxes) obtained by the model before carrying out this step of NMS. The middle picture represents the good prediction boxes obtained after using the default NMS. The picture on the right side shows the good prediction boxes obtained by using our improved NMS.

Table 5.1: Performance comparison with different NMS methods

	F1	Recall	Precision	mAP
AP@0.25				
Default NMS	0.88	**81.40%**	95.46%	**94.77%**
Contextual NMS	0.88	81.17%	**95.74%**	94.08%
Contextual NMS+	0.63	50.89%	82.42%	68.88%
AP@0.4				
Default NMS	0.86	**79.88%**	93.68%	**92.14%**
Contextual NMS	0.86	79.66%	**93.95%**	91.42%
Contextual NMS+	0.62	49.96%	80.92%	65.74%
AP@0.5 (Default)				
Default NMS	0.82	**75.98%**	88.79%	**84.45%**
Contextual NMS	0.82	75.76%	**89.02%**	83.92%
Contextual NMS+	0.51	45.70%	58.45%	51.98%
AP@0.6				
Default NMS	0.69	**63.86%**	74.90%	**62.84%**
Contextual NMS	0.69	63.68%	**75.11%**	62.32%
Contextual NMS+	0.50	40.75%	66.01%	80.92%
AP@0.75				
Default NMS	0.31	**28.72%**	33.68%	**14.54%**
Contextual NMS	0.31	28.65%	**33.78%**	14.47%
Contextual NMS+	0.24	19.06%	30.87%	11.36%

As shown in Table 5.1, overall, the Contextual NMS shows a slight increase in Precision or at least equal to the Default NMS in almost all IoU threshold settings, indicating that the Contextual NMS is more accurate in identifying positive classes and reduces false positives.For IoU thresholds of 0.25 and 0.4, the F1 scores of Contextual NMS and Default NMS are the same, but there is a slight improvement in accuracy. This indicates that Contextual NMS can recognize objects more accurately while maintaining the same recall rate, meaning that under these threshold settings, it can correctly recognize objects with a lower false positive rate. When the IoU threshold increases, the performance of all NMS methods decreases because higher IoU thresholds require more accurate bounding box alignment. We can observe that the performance degradation of Default NMS and Contextual NMS is relatively gradual. But the same contextual NMS also has a slightly better accuracy.

It is worth noting that under all IoU threshold settings, the Default NMS and Contextual NMS methods perform relatively well on the mAP metric, which means they maintain robust detection accuracy at different confidence thresholds. However, the Contextual NMS+,

which introduced the Soft-NMS mechanism, unexpectedly showed significant performance degradation at various IoU thresholds. This may be due to the mixed use of multiple NMS or the conflict between Soft-NMS and Contextual NMS mechanisms, which leads to overly strict suppression of detection boxes and thus results in poor performance.

5.5 Conclusion

In this study, we propose an improved Non Maximum Suppression algorithm aimed at optimizing detection accuracy in text detection tasks. Through experiments on different models, our method has demonstrated significant improvements over traditional NMS algorithms on specific datasets, particularly in terms of accuracy. However, our research also has some limitations, which indicate the direction of future research.

Firstly, due to time constraints, our experimental validation is limited to a single dataset. Future work will include validating the performance of improved NMS algorithms on a variety of datasets to ensure their generalization ability and stability. The experiment on multiple datasets will help us understand the effectiveness of algorithms in different scenarios and conditions, including text detection for different fonts, languages, and layout formats.

Secondly, although our algorithm considers features such as horizontal spacing and size of the text, other features of the text, such as the angle between internal characters and the inclusion relationship between characters and text boxes, have not been fully utilized. Future improvements can include more complex geometric and contextual features to enhance the model's recognition ability for complex text scenes.

Finally, considering that different NMS strategies may be complementary in different aspects, combining multiple effective NMS algorithms, such as Adaptive-NMS [15] that adaptively adjusts the detection threshold according to information density, may further improve detection performance. At the same time, the combination with the advanced text detection model based on bounding box regression is also expected to further improve the accuracy and robustness of text detection.

Based on the above points, although our research has achieved preliminary results, there is still significant room for improvement. By testing on a wider dataset, introducing more text features, and combining more advanced methods, we believe that the performance of text detection can be significantly improved. Future work will focus on these areas in order to develop more powerful and reliable text detection systems.

Acknowledgments

Two years at Hiroshima University is both short and long. When I think about how I crossed the coast and came to Japan as a freshman, I was very ignorant about everything, and how I have grown up to the point where I now have a good understanding of research and life in Japan, all of which could not have been accomplished without the proactive help of my advisors and lab members. I would like to thank Mr. Kurita and Mr. Aizawa, who have always brought positive suggestions for my research. From the time I entered the Kurita lab, I began to develop my research skills. When I was at a loss as to what to do with my research, it was the guidance of the sensei that allowed me to clarify the direction of my research and to search for appropriate research methods. I would like to thank my Chinese seniors and Japanese seniors, represented by Mr. Zheng. They gave me a lot of advice and solved a lot of practical problems in addition to research and code. Not only studying and researching, but also helping me to quickly adapt to my study abroad life in Hiroshima University. Thanks to my peers, I had study exchanges and thought collisions with them, as well as participated in various activities together, which also enriched my daily life outside of research.

Last but not least, I would like to give special thanks to my family, without all their support and trust, it would be very difficult for me to achieve my goal of studying abroad and realize my extraordinary life. Due to all these people who have helped me, they have made me what I am, and enabled me to face the difficulties, overcome the thorns, and march forward.

Reference

[1] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9365–9374, 2019.

[2] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S. Davis. Soft-nms – improving object detection with one line of code, 2017.

[3] Dan Deng, Haifeng Liu, Xuelong Li, and Deng Cai. Pixellink: Detecting scene text via instance segmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[4] Kunihiko Fukushima, Sei Miyake, and Takayuki Ito. Neocognitron: A neural network model for a mechanism of visual pattern recognition. *IEEE transactions on systems, man, and cybernetics*, (5):826–834, 1983.

[5] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[6] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] Pan He, Weilin Huang, Tong He, Qile Zhu, Yu Qiao, and Xiaolin Li. Single shot text detector with regional attention. In *Proceedings of the IEEE international conference on computer vision*, pages 3047–3055, 2017.

[9] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[11] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[12] Minghui Liao, Baoguang Shi, and Xiang Bai. Textboxes++: A single-shot oriented scene text detector. *IEEE transactions on image processing*, 27(8):3676–3690, 2018.

[13] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[14] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection, 2017.

[15] Songtao Liu, Di Huang, and Yunhong Wang. Adaptive nms: Refining pedestrian detection in a crowd, 2019.

[16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

[17] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE transactions on multimedia*, 20(11):3111–3122, 2018.

[18] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[19] Baoguang Shi, Xiang Bai, and Serge Belongie. Detecting oriented text in natural images by linking segments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2550–2558, 2017.

[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[21] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[22] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[23] Jun Tang, Zhibo Yang, Yongpan Wang, Qi Zheng, Yongchao Xu, and Xiang Bai. Seglink++: Detecting dense and arbitrary-shaped scene text by instance-aware component grouping. *Pattern recognition*, 96:106954, 2019.

[24] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image with connectionist text proposal network. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 56–72. Springer, 2016.

[25] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *2011 International conference on computer vision*, pages 1457–1464. IEEE, 2011.

[26] Yongchao Xu, Yukang Wang, Wei Zhou, Yongpan Wang, Zhibo Yang, and Xiang Bai. Textfield: Learning a deep direction field for irregular scene text detection. *IEEE Transactions on Image Processing*, 28(11):5566–5579, 2019.

[27] Jian Ye, Zhe Chen, Juhua Liu, and Bo Du. Textfusenet: Scene text detection with richer fused features. In *IJCAI*, volume 20, pages 516–522, 2020.

[28] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: an efficient and accurate scene text detector. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5551–5560, 2017.