ISSN 2616-5775 Vol. 7, Issue 5: 94-101, DOI: 10.25236/AJCIS.2024.070512

Deep learning-based text recognition in natural scenes

Lizhi Cui[1,a], Honglei Tian[1,b,*], Shumin Fei[1,c]

[1]Henan Polytechnic University, Jiaozuo, Henan, China
[a]clzh0308@hpu.cn, [b]vskyi@qq.com, [c]smfei@seu.edu.cn
[*]Corresponding author

Abstract: *The rapid expansion of Internet technology into remote areas has not only broadened network coverage but has also facilitated the proliferation of terminal devices. This enhancement in infrastructure boosts data resources, which are essential for advancing intelligence and automation linked to the Fourth Industrial Revolution. However, a significant challenge remains: many older devices are still offline and require updates. Technologies such as scene text recognition, crucial for applications in autonomous driving and traffic sign recognition, can address these updates. The shift from traditional statistical methods and SVMs to deep learning has significantly enhanced text recognition capabilities. To further improve these advancements, this article introduces the Pre-Activated Haar Transform-Enhanced Pan model (PAHaar Pan). This model incorporates a pre-activation design, depthwise separable convolution, and the Haar wavelet transform, enhancing feature extraction and generalization while reducing memory usage. Additionally, it is bolstered by a multi-level hybrid attention mechanism, providing superior text recognition performance.*

Keywords: *Deep Learning, Text recognition, Natural Scenes*

1. Introduction

The rapid development of Internet technology has greatly expanded network coverage, extending even to pristine forests and desert areas. This progress has facilitated the spread of terminal devices and contributed to a vast amount of data resources for the digitized world. The combination of the Internet and data has fueled advances in intelligence and automation, which are central to the realization of the Fourth Industrial Revolution. Despite the ubiquity of the Internet, many older devices are still not connected to the cloud, and updating their intelligence is crucial for system upgrades. Scene text recognition technology enables these devices to synchronize information with the cloud by digitizing surveillance data. Moreover, this technology is also used in various fields such as autonomous driving and traffic sign recognition.

Traditional methods such as statistical analysis and support vector machines (SVMs) used to play an important role in text recognition technology, but with the development of deep learning techniques, especially the application of convolutional neural networks (CNNs) and recurrent neural networks (RNNs), the performance of text recognition has been significantly improved. Deep learning techniques show higher robustness and accuracy when dealing with text of different font sizes, blurred and complex backgrounds, making traditional methods gradually obsolete. The traditional OCR algorithm flow, see Figure 1.

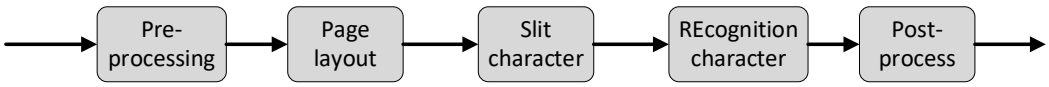

Figure 1: Traditional OCR Algorithm Process.

This article introduces the Pre-Activated Haar Transform-Enhanced Pan model (PAHaar Pan), which utilizes a pre-activation design to improve gradient propagation and mitigate internal covariate shift. It incorporates depthwise separable convolution to balance performance and size while enhancing generalization and reducing memory usage. The model also leverages the Haar wavelet transform for improved feature extraction at various granularities. Additionally, it features a multi-level hybrid attention mechanism that combines the Haar transform, residuals, and attention to enhance feature response sensitivity and discrimination, particularly for text recognition in scenes.

ISSN 2616-5775 Vol. 7, Issue 5: 94-101, DOI: 10.25236/AJCIS.2024.070512

2. PAHaar Pan Model Design

The PAHaar Pan model proposed in this paper consists mainly of four parts: Haar Depthwise Separable Pre-Activated Backbone (HaarDwisePreActivate Backbone), Light Weight Haar Wavelet Lossless Transformation Layer, Multi-Level Hybrid Attention, and Text Recognition Post-Processing Layer. The following sections will elaborate on these components in detail.

2.1 Haar Wavelet Down-Sampling Analysis and Discussion

The HWD is structured into two distinct sections: (1) an encoding module that ensures no loss of features, and (2) a learning module dedicated to feature representation. The encoding module leverages Haar wavelet transformations to effectively downscale the resolution of feature maps, thereby safeguarding the textural details inherent in images. Concurrently, the feature representation learning module is composed of conventional convolutional layers, complemented by batch normalization and ReLU activation layers, which are instrumental in isolating unique characteristics.

The Haar wavelet transformation, characterized by its compactness, binary nature, and orthogonality, is extensively employed across various domains including image encoding, edge extraction, and the design of binary logic systems. The definition of this transformation for one-dimensional signals is presented as follows:

Haar wavelet transformation is a compact, binary, and orthogonal transformation, widely used in fields such as image encoding, edge extraction, and binary logic design. For one-dimensional signals, the Haar wavelet transformation can be defined as follows:

$$\begin{cases} \phi_1(x) = \frac{1}{\sqrt{2}}\phi_{1,0}(x) + \frac{1}{\sqrt{2}}\phi_{1,1}(x) \\ \psi_1(x) = \frac{1}{\sqrt{2}}\phi_{1,0}(x) - \frac{1}{\sqrt{2}}\phi_{1,1}(x) \end{cases} \tag{1}$$

$\phi_{j,k}(x)$ is defined as follows:

$$\phi_{j,k}(x) = \sqrt{2^j}\phi(2^j x - k), k = 0,1,\cdots,2^j - 1 \tag{2}$$

The parameters j and k represent the order (or scale in the field of image processing) and sequence (or orientation for two-dimensional images) of the Haar basis functions, respectively. Here, $\phi_{0,0}(x)$ is defined as:

$$\phi_{0,0}(x) = \phi_0(x) = \begin{cases} 0, & x < 0 \\ 1, & 0 \le x < 1 \\ 0, & x \ge 1 \end{cases} \tag{3}$$

Thus, the first-order Haar transform can be represented using the zeroth-order Haar basis function:

$$\begin{cases} \phi_1(x) = \phi_0(2x) + \phi_0(2x-1) \\ \psi_1(x) = \phi_0(2x) - \phi_0(2x-1) \end{cases} \tag{4}$$

In the domain of signal processing, a signal characterized by a length L is typically subdivided into two subsections, each extending over a length of L/2. These segments represent the outputs processed through low-pass and high-pass filtering mechanisms, respectively. Furthermore, the application of Haar Wavelet Down-Sampling (HWD) to a grayscale image results in the generation of four discrete components. Notably, each of these components exhibits a spatial resolution that is precisely reduced to fifty percent of that observed in the original feature map.

2.2 Haar Wavelet Transform, Depthwise Separable and Preactivation Joint Design

Conventional convolutions merge feature maps in space and depth through multi-channel convolution kernels, while depthwise separable convolutions decompose this process, effectively extracting features with a lower parameter count and faster processing speed.

Assuming the convolution kernel size shown in the figure above is (k,k,c_{in}), stride is s, padding is p,

ISSN 2616-5775 Vol. 7, Issue 5: 94-101, DOI: 10.25236/AJCIS.2024.070512

input feature map size is (c_{in}, h, w) , number of parameters is , computational cost is $Flops$, output size is $(c_{out}, h_{out}, w_{out})$.

The parameter quantity for ordinary convolution is as follows:

$$\mathrm{W} = k \times k \times c_{in} \times c_{out} \tag{5}$$

The computational cost is as follows:

$$\begin{cases} h_{out} = \dfrac{h + 2p - k}{s} + 1 \\ w_{out} = \dfrac{w + 2p - k}{s} + 1 \\ Flops = h_{out} \times w_{out} \times c_{out} \times k^2 \times c_{in} \end{cases} \tag{6}$$

Depthwise separable convolution first uses depthwise convolution, then uses pointwise convolution to merge features. Its parameter quantity is as follows:

$$\begin{cases} \mathrm{W} = \mathrm{W}_1 + \mathrm{W}_2 \\ \mathrm{W}_1 = k \times k \times c_{in} \\ \mathrm{W}_2 = 1 \times 1 \times c_{in} \times c_{out} \end{cases} \tag{7}$$

The computational cost is as follows:

$$Flops = h_{out} \times w_{out} \times c_{in} \times k \times k + h_{out} \times w_{out} \times c_{in} \times c_{out} \times 1 \times 1 \tag{8}$$

By synthesizing formulas (6), (8), the relative ratio of the computational cost of depthwise separable convolution to traditional convolution can be derived:

$$\begin{aligned} R_{Flops} &= \frac{h_{out} \times w_{out} \times c_{in} \times k \times k + h_{out} \times w_{out} \times c_{in} \times c_{out} \times 1 \times 1}{h_{out} \times w_{out} \times c_{out} \times k^2 \times c_{in}} \\ &= \frac{1}{c_{out}} + \frac{1}{k \times k} \end{aligned} \tag{9}$$

Since the feature maps processed by depthwise separable convolution generally have a high number of channels, while the convolution kernels are generally conservative, the above formula is further simplified:

$$R_{Flops} \approx \frac{1}{k^2} \tag{10}$$

Further analysis of the relative ratio of parameter quantity:

$$\begin{aligned} R_W &= \frac{c_{in} \times k \times k + c_{in} \times c_{out} \times 1 \times 1}{c_{in} \times c_{out} \times k \times k} \\ &= \frac{1}{c_{out}} + \frac{1}{k^2} \approx \frac{1}{k^2} \end{aligned} \tag{11}$$

This section proposes a series of improvements to ResNet with Haar depthwise separable pre-activated ResNet. Firstly, the activation function in the network is moved from its original location to the input layer of the network, which allows gradients to propagate more directly from the output layer to the input layer during the backpropagation process, effectively solving the problem of gradient vanishing, and by pre-normalizing data to accelerate model convergence and training stability. Secondly, depthwise separable convolution is used in the residual network to merge features, improving parameter efficiency and reducing computational costs. Lastly, the Haar transform is used as the Identity branch to achieve lossless compression encoding when downsampling is needed, as shown in Figure 2.

ISSN 2616-5775 Vol. 7, Issue 5: 94-101, DOI: 10.25236/AJCIS.2024.070512

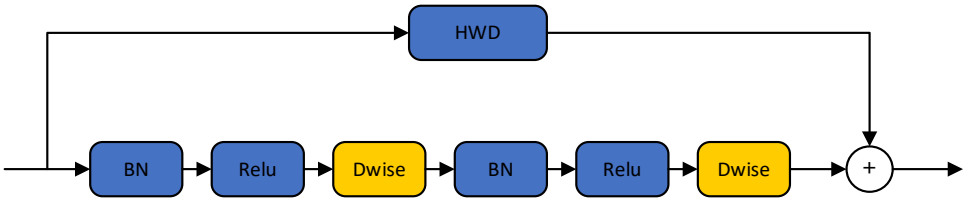

Figure 2: Haar deep separable pre-activation structure.

2.3 Hybrid Attention Module Design

Channel attention emphasizes the importance of feature channels, which may overlook key spatial information; while spatial attention, although able to locate text areas, may not fully consider the contributions of different feature channels. Therefore, the hybrid attention mechanism combines the advantages of both, improving the model's efficiency and accuracy in natural scene text recognition.

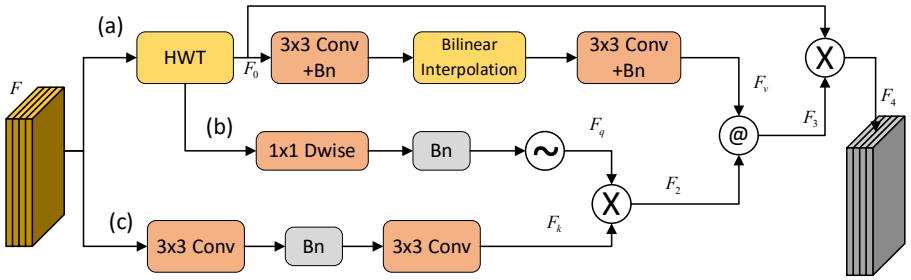

Figure 3: Hybrid Attention Module Design.

This section's hybrid attention module mainly consists of three branches, as shown in Figure 3. Considering that scene text has a variety of detailed features, simple feature maps cannot effectively accommodate sufficient detail information, and excessive features may affect the model's performance in real-time scenarios. Therefore, the HWD module proposed in section 2.1 is used to perform lossless encoding of original information to generate low-level feature F_0, to maximize the balance between performance and loss while retaining text features' detail information in various directions. Then, two 3x3 ordinary convolutions and bilinear interpolation upsampling are used to construct the text's nonlinear feature F_v. In branch (a), 1x1 separable convolution and bn layer are used to generate "redundant feature information" F_q at minimal cost, and smoothing is performed using $Sigmoid$. In branch (c), two 3x3 ordinary convolutions are used to generate feature F_k, using features F_k and F_v to generate F_2. Then from bottom to top, cascading feature F_2 and F_v to generate F_3, finally using residual structures to merge encoded feature F_0 to generate F_4. Through these improvements, not only is feature information integrity and channel and spatial information flow enhanced, but also information lossless encoding is achieved through Haar transformation to improve the model's accuracy in recognizing scene text.

2.4 Training Label Generation

The CTW 1500 dataset uses rectangular and polygonal frames to annotate text areas, where polygonal frames can be calculated based on the offset from the top left corner of the rectangular frame. The ICDAR 2015 dataset uses a fixed 8-point polygon to annotate text areas, and Total Text does not limit the number of points on the annotated polygonal frames. As no annotations for text kernels and text areas are provided, these three datasets cannot be used in this paper. This section introduces the method of generating training labels for datasets.

By shrinking the actual text area bounding box by a specified amount, the text kernel is generated, as shown in Figure 4. The blue frame represents the original frame, using b_o to represent, the red frame represents the shrunk text kernel frame, using b_k to represent, m representing the number of shrinking pixels. Specifically, by using the Vatti clipping algorithm, the original frame is shrunk by pixels to obtain the kernel frame b_k. Then, using 0,1 pixel filling to generate a binary mask, using G_{ker} to represent.

ISSN 2616-5775 Vol. 7, Issue 5: 94-101, DOI: 10.25236/AJCIS.2024.070512

Figure 4: Generating text kernel labels.

The shrinkage rate r represents the ratio between the text area and the text kernel. Text shapes are diverse, and using a formulaic ratio can better quantify the number of pixels scaled, achieving controlled and unified label generation. Wherein, when r=1 representing no shrinkage, when r=0 reducing all pixels (in fact, completely unnecessary), therefore the domain of r is given $(0,1]$. m is defined by the following formula:

$$m = \frac{\text{Area}(b_o) \times (1 - r^2)}{\text{Perimeter}(b_o)} \tag{12}$$

Wherein, $Area(\bullet)$ calculates the polygon area, $Perimeter(\bullet)$ calculates the polygon perimeter. Thus, through a unified shrinkage rate, the global image's shrinkage ratio can be quantified uniformly.

2.5 Loss Function

The total loss function is defined by the following formula:

$$L = L_{\text{det}} + L_{rec} \tag{13}$$

Wherein, L_{det} represents the loss of text detection, L_{rec} represents the loss of text recognition. Text detection loss L_{det} is defined by the following formula:

$$L_{\text{det}} = L_{\text{tex}} + \alpha L_{\text{ker}} + \beta(L_{\text{agg}} + L_{\text{dis}}) \tag{14}$$

Wherein, L_{tex} represents the loss of segmenting text areas, L_{ker} represents the loss of segmenting text kernels, α and β are coefficients balancing L_{tex}, L_{ker}, L_{agg}, and L_{dis}. In the experiments of this paper, α and β are respectively 0.5 and 0.25.

Dice loss is used to quantify the segmentation results of text areas P_{tex} and text kernels P_{ker}. L_{tex} and L_{ker} are calculated by the following formulas:

$$L_{\text{tex}} = 1 - \frac{2\sum_i P_{\text{tex}}(i) G_{\text{tex}}(i)}{\sum_i P_{\text{tex}}(i)^2 + \sum_i G_{\text{tex}}(i)^2} \tag{15}$$

$$L_{\text{ker}} = 1 - \frac{2\sum_i P_{\text{ker}}(i) G_{\text{ker}}(i)}{\sum_i P_{\text{ker}}(i)^2 + \sum_i G_{\text{ker}}(i)^2} \tag{16}$$

Wherein, $P_{tex}(i)$ and $G_{tex}(i)$ respectively represent the segmentation result and the value of the i-th pixel in the label of the text area (either 0 or 1). $P_{\text{ker}}(i)$ and $G_{\text{ker}}(i)$ respectively represent the segmentation result and the value of the i-th pixel in the label of the text kernel (either 0 or 1).

In calculating L_{tex}, online hard example mining (OHEM) is used to ignore simple non-text pixels. However, in calculating L_{ker}, L_{agg}, and L_{dis}, only text pixels are considered. The loss function for text recognition is:

$$L_{rec} = \frac{1}{|w|} \sum_{i=0}^{|w|} \text{CrossEntropy}(y_i, w_i) \tag{17}$$

ISSN 2616-5775 Vol. 7, Issue 5: 94-101, DOI: 10.25236/AJCIS.2024.070512

Wherein, w is the text label with an additional symbol EOS, y represents the text prediction result. $|\bullet|$ represents the total number of characters, w_i represents the i-th value of the text label, y_i represents the nth value of the text prediction result, Cross Entropy represents the cross-entropy loss function.

3. Experiments and Data Analysis

3.1 Dataset

The Total Text Dataset, introduced by Chee Kheng Ch'ng and Chee Seng Chan at ICDAR 2017 conference, comprises 1555 images with various text characteristics, such as horizontal, multi-directional, and curved text. It uses coordinate points and words for annotation. The ICDAR 2015 dataset includes 1500 images with 2077 cropped text instances, often irregular, tilted, or blurred. The SynthText dataset simulates text in natural scenes with transformations and effects. The COCO-Text dataset, currently the largest for natural scene text detection, features 1.7 million text instances. The ICDAR 2017 MLT dataset covers nine languages with 18,000 images annotated with quadrilateral word-level annotations.

3.2 Experimental Settings

Experiments were conducted using public datasets Total Text and ICDAR 2015 and compared with previous methods. The instance vector dimension is 4, the negative-positive ratio of OHEM is 3, the shrinkage rate r of the text kernel is 0.7, and the distance threshold d of PA is 3.

During training, blurry text areas labeled "DO NOT CARE" were ignored, and training images were subjected to random scaling, random horizontal flipping, random rotation, and random cropping. All models were optimized using the ADAM optimizer. Batch size was 16, number of GPUs was 1. The initial learning rate was 1×10^{-3}. A "poly" learning rate adjustment strategy was used, as follows:

$$l = l_0 \times \left(1 - \frac{iter}{\max iter}\right)^{power} \tag{18}$$

Where $power$ is 0.9. During the testing phase, testing was conducted using a single thread and batch size of 1. Since the hardware varies in different experiments, the FPS results in this paper are for reference only.

3.3 Experimental Analysis

In the text recognition task, two main training strategies were adopted: the first involves pre-training the model using additional text datasets, then fine-tuning on specific target datasets; the second involves training the model on a joint dataset combining multiple datasets including SynthText, COCO Text, IC17-MLT, Total Text, and ICDAR 2015.

In the testing results on the Total Text dataset, as shown in Table 1, "None" represents not using a dictionary; "Full" represents using a dictionary containing all the words in the test set.

Table 1: Test results on the Total Text dataset.

Module	Backbone	Strategy	Extra dataset	Total Text		
				None	Full	FPS
TextNet[1]	ResNet50	Finetune	SynthText	54.0	-	2.7
TextDragon[2]	VGG16	Finetune	SynthText,IC15	48.8	74.8	-
ABCNet[3]	ResNet50	Finetune	SynthText, IC19MLT	64.2	75.7	17.9
Ours	ResNet	Finetune	SynthText,COCO-Text,IC17-MLT	67.8	76	45
TextBoxes[4]	ResNet50	Jointly	SynthText,IC13,IC15	36.3	48.9	1.4
Mask TextSpotter[5]	ResNet50	Jointly	SynthText,IC13,IC15	52.9	71.8	4.8
Mask TextSpotter V2[5]	ResNet50	Jointly	SynthText,IC13,IC15	65.3	77.4	-
Qin[6]	ResNet50	Jointly	SynthText,IC15, IC17-MLT	67.8	-	4.8
Ours	ResNet	Jointly	SynthText,COCO Text,IC17-MLT,Total Text,ICDAR 2015	68.9	78.6	21.0

In the absence of a dictionary, the accuracy is 67.8%, and with a dictionary, it is 76%, which is an increase of 3.4% and 0.3 percentage points respectively compared to ABCNet. This indicates that the model has good generalizability. In the joint training strategy, the accuracy without a dictionary is 1.1%

ISSN 2616-5775 Vol. 7, Issue 5: 94-101, DOI: 10.25236/AJCIS.2024.070512

higher than the network proposed by Qin, and the accuracy with a dictionary is 1.4 % higher than Mask TextSpotter V2.

In the tests on the ICDAR 2015 dataset, as shown in Table 2, when using a pre-training strategy, the accuracies of S, W, and G are 82.6%, 78.7%, and 68.9% respectively, compared to ABCnet, they have increased by 0.1%, 0.5%, and 3.7% respectively. During joint training, these values are 82.9%, 79.2%, and 70.3% respectively, showing advantages compared to other models, and compared to Qin's results, S, W, and G have increased by 2.4%, 3.4%, and 4.1% respectively.

Table 2: Test results on the ICDAR 2015 dataset. S: Provides a total of 100 words per test image, including true annotations. W: A dictionary composed of all words in the test set. G: A dictionary containing 90k common words.

Module	Backbone	Strategy	Extra dataset	ICDAR 2015			
				S	W	G	FPS
TextNet[1]	ResNet50	Finetune	SynthText	81.1	75.9	60.8	7.5
CharNet[7]	ResNet50	Finetune	SynthText,IC15,IC17-MLT	82.4	78.9	67.6	1.2
ABCNet[3]	ResNet50	Finetune	SynthText, IC19MLT	82.5	78.3	65.2	-
Ours	ResNet	Finetune	SynthText,COCO-Text,IC17-MLT	82.6	78.7	68.9	45
TextBoxes[4]	ResNet50	Jointly	SynthText,IC13,IC15	54.0	51.0	47.0	9.0
Mask TextSpotter[5]	ResNet50	Jointly	SynthText,IC13,IC15	74.2	69.2	63.5	3.8
Qin[6]	ResNet50	Jointly	SynthText,IC15, IC17-MLT	80.5	75.8	66.2	4.8
Ours	ResNet	Jointly	SynthText,IC15, IC17-MLT	82.9	79.2	70.3	20.1

This section of the experiment verifies the effectiveness of the algorithm proposed in the article in dealing with various complex challenges of scene text images, which is due to the reconstructed efficient backbone using pre-activated depth-separable design and the reconstructed multi-level hybrid attention optimized jointly with Haar transform, residuals, and attention structures.

3.4 Conclusion

In this section, an algorithm for text recognition is introduced, utilizing a residual backbone constructed with Haar pre-activated depth-separable design and multi-level hybrid attention technology optimized jointly with Haar transform, residuals, and attention structures. Experimental results show that the model achieves good results on multiple public datasets, especially in recognizing text in complex environments. However, due to issues like blurring and deformation in the datasets themselves, there is still room for further exploration in image denoising and model generalization in the future.

References

[1] Sun Y, Zhang C, Huang Z, et al. TextNet: Irregular text reading from images with an end-to-end trainable network[M/OL]. arXiv, 2018.
[2] Feng W, He W, Yin F, et al. TextDragon: An End-to-End Framework for Arbitrary Shaped Text Spotting [J/OL]. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, null: 9075-9084.
[3] Liu Y, Chen H, Shen C, et al. ABCNet: Real-Time Scene Text Spotting With Adaptive Bezier-Curve Network[J/OL]. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, null: 9806-9815.
[4] Liao M, Shi B, Bai X, et al. TextBoxes: A fast text detector with a single deep neural network[C/OL]//Proceedings of the AAAI Conference on Artificial Intelligence: Vol. 31. 2017.
[5] Lyu P, Liao M, Yao C, et al. Mask TextSpotter: An End-to-End Trainable Neural Network for Spotting Text with Arbitrary Shapes[J/OL]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 43: 532-548.
[6] Qin S, Bissacco A, Raptis M, et al. Towards Unconstrained End-to-End Text Spotting[J/OL]. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, null: 4703-4713.
[7] Peng D, Jin L, Ma W, et al. Recognition of handwritten Chinese text by segmentation: A segment-annotation-free approach[J/OL]. IEEE Transactions on Multimedia, 2023, 25: 2368-2381.