International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 41, No. 6, December, 2024, pp. 2839-2850
Journal homepage: http://iieta.org/journals/ts

Design of an Image Content Understanding and Information Extraction Algorithm Integrating Natural Language Processing

Ling Pang[1], Aihua Li[2*]

[1] School of Computer and Information Engineering, Hebei Finance University, Baoding 071051, China
[2] College of Data Science and Software Engineering, Baoding University, Baoding 071000, China

Corresponding Author Email: liaihua@bdu.edu.cn

https://doi.org/10.18280/ts.410606

Received: 15 June 2024
Revised: 28 September 2024
Accepted: 20 October 2024
Available online: 31 December 2024

Keywords:
image content understanding, natural language processing (NLP), end-to-end method, information extraction, multimodal learning

ABSTRACT

With the rapid development of artificial intelligence (AI) technologies, the integration of image content understanding and natural language processing (NLP) has become a hot research topic in the fields of computer vision and NLP. Image content understanding requires not only image classification and object detection capabilities but also the ability to perform in-depth semantic analysis and expression of complex information within the image. Advances in NLP have enabled computers to generate natural language descriptions related to the content of images, thereby facilitating cross-modal communication. In recent years, end-to-end image content understanding methods and NLP-based image information extraction algorithms have gradually become essential technologies for solving multimodal learning and information extraction problems. However, existing methods still have certain limitations in terms of multimodal fusion, information transfer accuracy, and contextual understanding, especially when handling complex scenes and applications where system robustness and accuracy often fail to meet practical requirements. To address these issues, this paper proposes an image content understanding and information extraction algorithm design that integrates NLP. The main contributions of this paper are twofold: first, a deep learning-based end-to-end image content understanding method is proposed, which can directly extract efficient features from images and generate accurate natural language descriptions; second, an NLP-integrated image content information extraction method is introduced, which achieves more efficient and precise multimodal information extraction through deep coupling of image and text information. Experimental results show that the proposed methods significantly improve the accuracy and efficiency of image description generation and information extraction tasks, providing strong support for the deep integration of images and language.

1. INTRODUCTION

With the rapid development of information technology, especially breakthroughs in AI and deep learning, image content understanding has become an important research direction in the field of computer vision [1-5]. Image content understanding not only involves classifying and detecting objects in images but also requires in-depth analysis, reasoning, and interpretation of the multi-layered information within images [6-8]. Against this backdrop, the integration of image content understanding with NLP has gradually become a research hotspot, aiming to enable computers to understand and express image content like humans. The core goal of this research is to automatically parse image content and generate natural language descriptions or directly interact with text information, thus enhancing the level of human-computer interaction. With the emergence of multimodal learning methods, joint understanding of image and language has gradually become a key technology for realizing more complex intelligent applications.

The combination of NLP and image content understanding and information extraction algorithms not only has significant research value in academia but also shows great potential in practical applications. Technologies such as automatic image caption generation, image question answering, and intelligent search engines are changing the way we interact with image data, bringing more convenient information retrieval, visual assistance, and intelligent decision-making applications [9-13]. For example, in e-commerce, automatic image descriptions can improve the accuracy of product search; in smart homes and medical imaging, technologies that combine image understanding and language generation can provide users with more intuitive operations and assist in diagnosis [14-17]. These applications not only drive the progress of AI but also bring about changes in real-life scenarios. Therefore, research on how to improve the accuracy and efficiency of image content understanding and explore its deep integration with NLP has become a current research focus.

Although existing research has made some progress in image content understanding and natural language generation, there are still many shortcomings [18-25]. First, traditional image understanding methods often rely on specific tasks and rules, making it difficult to adapt flexibly to diverse application scenarios. For example, generating precise natural language descriptions for different types of images still faces challenges such as insufficient semantic understanding and the loss of contextual information. Second, existing multimodal learning methods typically adopt a staged processing approach, with image feature extraction and text generation being relatively independent, leading to data loss and inconsistency in the information transfer process. In addition, many studies only focus on unidirectional interaction between image and language, neglecting the deep coupling and interaction between the two. For information extraction tasks, although some image-based feature extraction methods exist, there is still a lack of a unified and end-to-end solution for effectively combining language models with image content for information extraction.

To address the above issues, this paper proposes an image content understanding and information extraction algorithm design integrating NLP, aiming to enhance the deep integration and interaction between image and language through end-to-end image content understanding and information extraction methods. The first part of this paper focuses on end-to-end image content understanding, designing a multimodal model based on deep learning that can directly extract useful information from images and generate precise natural language descriptions without manual intervention. The second part proposes an image content information extraction method combined with NLP, which can more efficiently extract key data and information from images, conduct semantic analysis, and provide more accurate and comprehensive support for subsequent tasks. By combining these two methods, this paper aims to promote the deep integration of image and language understanding and provide more precise and efficient technical solutions for intelligent applications. This has important academic value and will have a broad impact in practical applications.

2. END-TO-END IMAGE CONTENT UNDERSTANDING METHOD

Image content understanding involves the processing and integration of information across two modalities: vision and language. Traditional methods often face bottlenecks in information conversion and integration between these modalities. Additionally, traditional image content understanding methods are typically divided into multiple independent steps, such as image detection, feature extraction, and text generation. These steps require individual optimization, and each step may introduce errors, leading to a decline in the overall system performance. Therefore, this paper chooses to conduct research on end-to-end image content understanding methods to achieve more efficient integration of image and text information and provide stronger capabilities for the combination of NLP and image content understanding.

End-to-end image content understanding methods have demonstrated strong potential in several practical application scenarios, particularly in automatic image caption generation and image question answering tasks. In automatic image caption generation, the end-to-end method can directly extract visual features from the input image using deep neural networks and generate natural language description text. This technology is widely used in assistive devices for the blind, converting information from images into text, thus helping visually impaired individuals "see" the world. Additionally, end-to-end image understanding methods have been widely applied in social media platforms, where they automatically generate descriptions for user-uploaded images, enhancing user experience and content accessibility. At the same time, this technology is applied in e-commerce to automatically generate product image descriptions, improving the accuracy of product search and recommendations, thus increasing user conversion rates. Figure 1 shows the basic structure of an end-to-end image content understanding model.

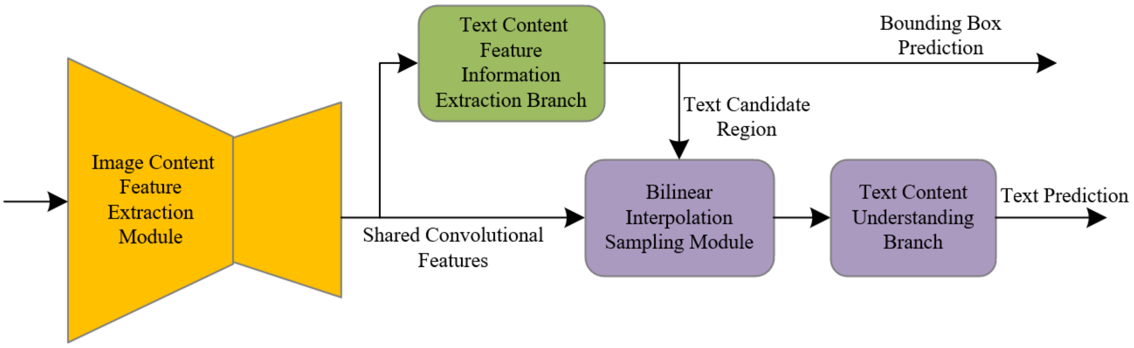
Image Content Feature Extraction Module
Text Content Feature Information Extraction Branch
Bounding Box Prediction
Text Candidate Region
Shared Convolutional Features
Bilinear Interpolation Sampling Module
Text Content Understanding Branch
Text Prediction

Figure 1. Basic structure of the end-to-end image content understanding model

2.1 Design of the bilinear interpolation sampling module

The end-to-end image content understanding method emphasizes completeness and efficiency, so the design of the bilinear interpolation sampling module is carried out first in this paper. By dynamically adjusting the sampling process, the module can effectively handle text candidate regions of various shapes and orientations, avoiding the deformation problems caused by fixed grids in traditional methods. However, in the field of computer vision, especially in image content recognition tasks, the traditional RoI Pooling method, with its fixed grid characteristics, often leads to feature distortion and information loss when handling text candidate regions of varying lengths and rotations. To overcome this limitation, this paper opts to construct a bilinear interpolation sampling module based on a spatial transformer network. This module can flexibly sample feature maps according to the actual shape and orientation of the input candidate regions, thus better preserving the original form and structural features of the text.

The spatial transformer network can automatically perform spatial transformations on the input image or feature map, aligning it to a standard coordinate system, thus providing normalized input for subsequent text recognition. The main components of a spatial transformer network include three important parts: the localization network, the parameterized sampling grid generator, and the pixel sampler. The localization network generates the spatial transformation parameters θ through a combination of fully connected layers or convolutional layers and regression layers. This network is responsible for predicting the type of spatial transformation required (e.g., translation, rotation, scaling) based on the input feature map and outputting the corresponding transformation parameters. The parameterized sampling grid generator calculates the new coordinates of each pixel in the input image in the output feature map based on these spatial transformation parameters, thus achieving coordinate mapping from the input feature map to the output feature map. The pixel sampler then uses the bilinear interpolation method to sample the pixel values for each output feature map position according to the generated sampling grid.

The bilinear interpolation sampling operation consists of two key steps. The first step is to calculate the bilinear interpolation sampling parameter matrix *L*. For each text region *M*, by determining the rotation angle *M* of the region and the boundaries of the region (top, right, bottom, and left distances) *s*, *e*, *y*, *m*, the shape and position of the region can be uniquely defined. To map these text regions with rotation and varying lengths to a target region with no rotation and a fixed aspect ratio, it is necessary to first calculate the bilinear interpolation sampling matrix *L* using these parameters, which can achieve mapping from the original region to the target region. To ensure consistent output size during the mapping process, the height of the target feature map is fixed in this paper, and the width of the target region is adjusted by convolutional expansion so that all input regions can fit the final content understanding branch's feature map size. Assuming the relative offsets along the horizontal and vertical axes are represented by s_a and s_b, and the scaling transformation ratio is denoted by *t*, the height and width of the bilinearly interpolated feature map are denoted by g_s and q_s, respectively. The parameter matrix *L* can be calculated as follows:

$$s_a = 1 * COS\varphi - s * SIN\varphi - a \tag{1}$$

$$s_b = s * COS\varphi + 1 * SIN\imath - b \tag{2}$$

$$t = \frac{g_s}{s + y} \tag{3}$$

$$q_s = t * (1 + e) \tag{4}$$

$$L = \begin{bmatrix} COS\varphi & -SIN\varphi & 0 \\ SIN\varphi & COS\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} t & 0 & 0 \\ 0 & t & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & s_a \\ 0 & 1 & s_b \\ 0 & 0 & 1 \end{bmatrix}$$
$$= t \begin{bmatrix} COS\varphi & -SIN\varphi & s_a COS\varphi - s_b SIN\varphi \\ SIN\varphi & COS\varphi & s_a SIN\varphi + s_b COS\varphi \\ 0 & 0 & \frac{1}{t} \end{bmatrix} \tag{5}$$

The second step is to generate a fixed-size axis-aligned content understanding feature map D_{RE} through bilinear interpolation sampling. In this step, each input candidate region undergoes bilinear interpolation calculation to obtain the coordinates (*u*,*k*) of each point in the target region and the corresponding mapped coordinates (a^t_{uk}, b^t_{uk}). For each pixel (*u*,*k*) in the target region, it corresponds to a mapped point (a^t_{uk}, b^t_{uk}) in the original region, and the feature value at this position in the target region is obtained through interpolation. In this way, the bilinear interpolation operation can not only handle rotated and length-varying regions but also ensure that each feature point is precisely mapped to a fixed coordinate system in the target region. Finally, through this process, the resulting fixed-size and axis-aligned feature map D_{RE} can be input into the content understanding branch, providing a clear and consistent feature representation for subsequent text recognition and image content analysis. The coordinate correspondence between the two is given by the following equation:

$$\begin{pmatrix} a^t_{uk} \\ b^t_{uk} \\ 1 \end{pmatrix} = L^{-1} \begin{pmatrix} k \\ k \\ 1 \end{pmatrix}, \forall u \in [0, q_s - 1], \forall k \in [0, g_s - 1] \tag{6}$$

Assuming that the coordinates of the target candidate region's feature map are (*u*,*k*) and the number of channels is *z*, the value at position $(D_{RE})^z{}_{uk}$ can be computed as follows:

$$(D_{RE})_{uk} = \sum_{v=0}^{Q'-1} \sum_{l=0}^{G'-1} (D_{AH})^z_{vl} \, J(a^t_{uk}, v) \, J(b^t_{uk}, l) \tag{7}$$

To improve the model's adaptability to text regions of different scales, we introduce a multi-scale sampling strategy in the bilinear interpolation sampling module. First, we use a convolutional neural network (CNN) to extract feature maps at various scales. In images, text regions typically exhibit significant scale variation, so we perform gradual downsampling to generate feature maps at multiple levels. For each feature map at a particular scale, bilinear interpolation is used for region sampling. To handle text regions at different scales, we dynamically select the appropriate scale based on the features and size of each region, ensuring that text information is sufficiently preserved across different scales. The sampled feature maps are then fused and input into the subsequent model layers. The multi-scale sampling approach not only enhances the representational power of text regions but also improves the model's robustness on images with different resolutions.

2.2 Content understanding branch design

In designing the image content understanding branch, we selected CNN and Bidirectional Long Short-Term Memory (Bi-LSTM) networks as the core components. Specifically, we adopted ResNet-50 as the backbone network for the CNN part, primarily due to its residual connections, which alleviate the vanishing gradient problem in deep networks, and because ResNet excels in image classification and feature extraction tasks. For text information processing, we chose Bi-LSTM, as it can capture contextual information from both the forward and backward directions of the sequence, making it particularly suitable for handling the linguistic structure and semantic information in image descriptions. Through this bidirectional learning, the model is able to more accurately understand complex semantic relationships in the image.

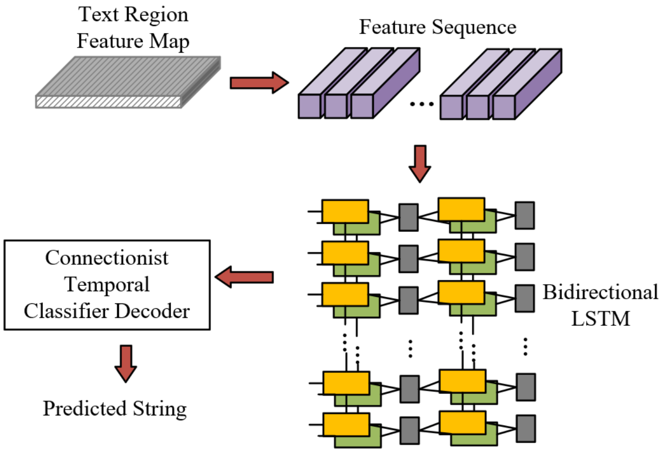

Figure 2. Process flow of the image content understanding branch

To validate the effectiveness of this design, we conducted multiple comparative experiments. In terms of CNN architectures, we compared ResNet-50 with other CNN networks, such as VGG-16 and PVANet. The experimental results show that ResNet-50 outperforms other architectures in terms of accuracy and stability, particularly excelling in extracting image content from complex backgrounds. Regarding Bi-LSTM versus LSTM, we also compared the performance of Bi-LSTM and traditional LSTM in image description tasks. The results indicate that Bi-LSTM, through bidirectional learning, significantly improves the accuracy and fluency of descriptions, especially when handling text with high ambiguity. These comparative experiments demonstrate that the selected CNN and Bi-LSTM architectures effectively enhance the accuracy of image content understanding and the quality of language descriptions.

The content understanding branch designed in this paper consists of two key parts: the encoder and the decoder. The process flow diagram is shown in Figure 2. The core principle of this design is to use an end-to-end architecture that combines the advantages of CNN and recurrent neural network (RNN) to automatically parse and understand complex text images. In this process, the encoder extracts high-level semantic information from the input feature map, while the decoder converts this information into a string, ultimately achieving text recognition and generation. Specifically, the encoder is responsible for extracting higher-level semantic information from the input feature map. The design of this part uses a combination of CNN and Bi-LSTM networks. The convolutional layers are mainly used to capture local features and perform convolutions along the spatial dimensions of the feature map, extracting spatial hierarchical information of the features. The Bi-LSTM network plays an important role in sequence modeling, learning the feature sequence simultaneously from both forward and backward directions, fully considering contextual information, thereby effectively enhancing the understanding of the text content. The output of the encoder is a feature sequence D_r that has been processed by deep learning, containing semantic information of the text in the image, providing sufficient contextual support for the decoder.

$$D_r = d_{EN}\left(D_{RE}\right) \tag{8}$$

The task of the decoder is to convert the text feature sequence output by the encoder into the final text content. This paper uses the Connectionist Temporal Classification (CTC) as the core model of the decoder. The key idea of this method is to output a probability distribution of possible label sequences at each time step of the input sequence, and then select the label with the highest probability as the final output. The advantage of CTC lies in its ability to solve the challenges faced by traditional RNN when handling sequences with varying lengths, especially in cases without explicit alignment information. In image content understanding tasks, CTC can handle variable-length text sequences and the temporal relationships between characters in the image, ensuring that the model does not generate errors due to the lack of alignment information during training. The decoding formula is given as follows, where v represents the sequence length.

$$B_{ST} = d_{DE}\left(\left(D_r\right)_1, \left(D_r\right)_2 \cdots \left(D_r\right)_v\right) \tag{9}$$

The work process of the content understanding branch can be summarized in two key steps. The first step is to encode the input text region feature map using CNN and Bi-LSTM. Specifically, after processing by the front-end shared convolution network, text detection branch, and bilinear interpolation module, the system generates fixed-size, axis-aligned text region feature maps. These feature maps are first input along the height axis into six sequential convolutional layers and downsampling pooling layers, with the size of the convolutional layers gradually decreasing, aiming to extract deeper semantic information from the feature map. To maintain the simplicity of the model, the convolutional layers adopt a structure similar to VGG-16. Through this series of convolution and pooling operations, the feature map size gradually decreases, and finally, a feature map of size 1×1 with 256 channels is output. Next, these feature maps are concatenated along the channel dimension to form a feature vector sequence of length 1, where each vector has a dimension of 256. Finally, the feature vector sequence is further encoded through a BiLSTM, which can extract contextual information from both the forward and backward temporal dimensions. The two 256-dimensional output vectors at each time step are concatenated, resulting in a 512-dimensional feature vector sequence. The output of the encoder is this processed feature vector sequence, with a length of 1, representing the high-level semantics of the input text region features.

The second step is to decode the feature vector sequence output by the encoder using the CTC, thus transforming it into the final text prediction sequence. In this process, the content understanding branch inputs the feature vector sequence output by the encoder into the CTC decoder. The CTC decoder generates a probability distribution for each input feature vector, representing the possible corresponding character or symbol at that time step. Finally, the CTC decoder constructs the possible text sequence by selecting the series of labels with the highest probabilities. This paper uses the CTC loss function, with the goal of ensuring that the decoding result is as close as possible to the real label sequence, thus optimizing the network training process. Specifically, let the character probability sequence to be transcribed be $b=b_1,\ldots,b_v$, and the transcribed label sequence be b^*_v, where V represents the number of text candidate regions in the input image. The calculation formula for the branch's loss function is as follows:

$$o\left(1|b\right)=\sum\nolimits_{\tau\cdot Y(\tau)=1}o\left(\tau|b\right) \tag{10}$$

$$M_{RE}=-\frac{1}{V}\sum\nolimits_{v=1}^{V}\log o\left(b_v^*|b\right) \tag{11}$$

In designing the end-to-end image content understanding model, we conducted experiments on various hyperparameter configurations to evaluate their impact on model performance. Specifically, we focused on the following key hyperparameters:

Learning Rate: We tested multiple learning rates (0.001, 0.01, 0.1) and found through experimental comparison that a learning rate of 0.01 achieved the optimal balance between model convergence speed and performance.

Batch Size: Batch size plays a crucial role in training stability and computational efficiency. By comparing model performance under different batch sizes (16, 32, 64), the experiment showed that a batch size of 32 provided the best performance during training while avoiding memory overflow issues.

Network Depth: In the experiments, we tried different network depths, ranging from ResNet-50 to ResNet-101. The results indicated that increasing the depth brought some performance improvement, but beyond 50 layers, the performance gains became less significant.

The experimental results demonstrate that the end-to-end image content understanding model with the above configurations outperforms models with other hyperparameter setups, showing superior accuracy and robustness.

3. IMAGE CONTENT INFORMATION EXTRACTION METHOD COMBINED WITH NLP

Furthermore, this paper proposes an image content information extraction method combined with NLP techniques, aiming to enhance the accuracy of text understanding in images by combining the image text recognition results with the word segmentation technique of NLP. The specific implementation idea of this method is as follows: first, the image content information extraction network performs preliminary recognition of the text in the image to obtain a rough text sequence. Then, an NLP word segmentation tool is used to segment the recognized text, identifying the words and their corresponding grammatical structure. Based on the relationships between the texts, especially by analyzing the contextual information in the segmentation results, it can be determined which parts of the text need to be supplemented due to recognition errors or omissions. For those areas with incomplete or inaccurate recognition results, the judgment is made based on the last word in the segmentation sequence. For example, a single-character word with fewer contextual associations should be supplemented together with the previous word. By combining the backend supplement recognition module, the recognition results are corrected and supplemented to generate more complete and accurate text content.

To ensure the accuracy and diversity of information extraction, we adopted a multi-dimensional approach in constructing the vocabulary. We performed frequency analysis on the image content in the training dataset and selected the most frequently occurring terms as the base vocabulary set. For example, in the dataset of human activities, terms like "running" and "jumping" were commonly observed. We then used a pre-trained word embedding model to calculate the semantic similarity between words. This approach allowed us to not only select frequently appearing terms but also ensure that these terms accurately reflected the semantic relationships within the images. For instance, the word "dog" has a high semantic similarity to "pet," so we included "pet" as a related term. To accommodate different types of image content, we dynamically expanded the vocabulary by incorporating contextual information from the images. For example, for images related to vehicles, we introduced terms like "car," "bus," and "train." The vocabulary is regularly updated based on new image data inputs. We employed an incremental update approach to ensure that new data could enrich the existing vocabulary in a timely manner. During the update process, vocabulary additions were based on the relevance of image labels and natural language descriptions, as well as the semantic richness of the terms.

3.1 Implementation principle of the image content information extraction network

To evaluate the performance of different backbone networks in image content information extraction tasks, we compared the performance of PVANet and ResNet-50. The experimental results show that ResNet-50 outperforms PVANet by approximately 5% in terms of accuracy when handling complex scenes and multi-object images. Specifically, ResNet-50 excels in recognizing fine details and small objects in images, particularly in low-contrast or high-noise environments. Despite having more layers, ResNet-50's inference speed is only about 8% slower than PVANet due to reasonable optimizations such as BatchNorm and convolutional improvements. Furthermore, the speed difference becomes negligible with multi-threading and GPU acceleration. While ResNet-50 has slightly more parameters than PVANet, its hierarchical structure and feature extraction capabilities make it more advantageous in complex tasks, especially for multi-class and multi-scale image recognition.

In the image content information extraction method combined with NLP, the image content information extraction network uses a general deep CNN as the backbone network. Here, ResNet-50 is used instead of the traditional PVANet, mainly because ResNet-50 has strong feature extraction capabilities and can effectively capture text information in the image at multiple scales. The network structure is shown in Figure 3. Through this network, the system can extract feature maps at different levels to obtain multi-scale text features. In the feature fusion layer, a feature fusion strategy similar to U-net is adopted, where the four feature maps extracted from the image content information extraction network are fused according to specific rules. Specifically, the top feature map is upsampled and merged with the lower-level feature map, which not only preserves the high-level semantic information but also ensures that low-level detail features are effectively utilized. The network output layer consists of two main parts: the text score map and the text shape information. The text score map has the same size as the input image, where each pixel indicates whether that position is part of the text. For texts with different shapes, the system outputs different formats: if the text box is a rotated rectangle, it outputs 5 channels representing the distance of the four edges of the text box and the rotation angle; if it is a quadrilateral, it outputs 8 channels representing the coordinates of the four corners of the rectangle. This structure can effectively handle different forms of text regions and optimize through the Dice loss function to improve the accuracy of text region detection. In addition, a linear learning rate decay method is used instead of the traditional staged learning rate decay method to ensure smoother learning rate changes during training, improving model convergence speed and stability. Through this structure and implementation method, the image content information extraction network can efficiently extract multi-scale text features from the image and provide an accurate foundation for subsequent text recognition and supplementation.

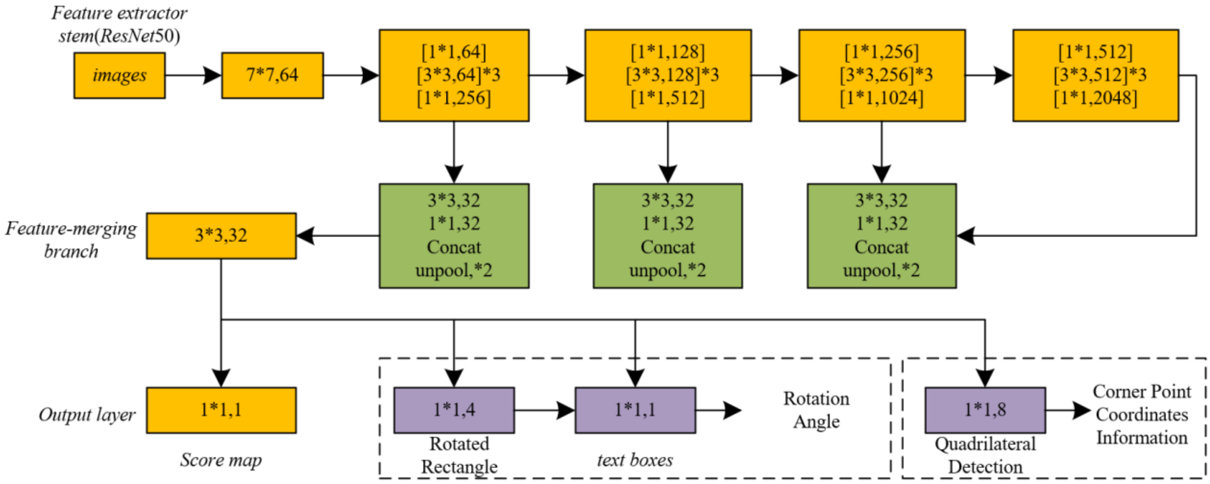

Figure 3. Image content information extraction network structure

The details of the implementation of image content information extraction based on the ResNet-50 network are first reflected in the image feature extraction phase. By calling the '*resnet_vl*_50' method, the image is passed into the four main blocks of the ResNet-50 network for processing. These Block blocks correspond to the four convolutional layers of ResNet-50, and the output of each Block generates feature maps at different scales, which are stored in the '*end_points*' dictionary, specifically including '*end_points*['*pool*2']', '*end_points*['*pool*3']', '*end_points*['*pool*4']', and '*end_points*['*pool*5']'. The sizes of these feature maps gradually decrease to 1/32, 1/16, 1/8, and 1/4 of the original image, respectively. Through these feature maps, the network can capture different levels of text information in the image, from details to high-level semantics, providing a multi-scale feature foundation for subsequent text region detection and understanding.

In the feature fusion process, to effectively combine the feature maps obtained from different levels, upsampling operations are used. The purpose of upsampling is to enlarge the low-level feature maps so that they can be aligned and fused with the high-level feature maps. This process begins with the lowest-level feature map '*d*1', which is first upsampled and then fused with '*d*2' on the channel dimension. Next, the fused feature map continues to be upsampled and fused with higher-level feature maps, with '*d*2' upsampled and fused with '*d*3', and '*d*3' further fused with '*d*4'. Through this layer-by-layer fusion process, features at different scales are effectively combined to form a multi-scale feature map. This fused feature map can effectively recognize and predict text lines at different scales, ensuring that both large and small texts can be accurately captured by the system. This fusion operation embodies the idea of U-Net, which uses layer-by-layer upsampling and fusion of feature maps at different levels to obtain richer contextual information.

Finally, in the output layer, the network generates three main output parts: one for the classification task output—the '*score map*', which is a single-channel image indicating whether each position belongs to the text region. Text region pixels are marked as 1, while non-text region pixels are 0. The remaining two outputs are used for text box detection: one is for the rotated rectangle output, which has 5 channels representing the distances of the four edges of the text box and the rotation angle of the rectangle; the other is for quadrilateral detection, which has 8 channels representing the coordinates of the four corners of the rectangle. These output parts can detect the text region according to its different shapes and precisely locate the position and shape of the text box. Through this multi-task output design, the network can not only recognize the presence of text but also provide the precise boundaries of the text, ensuring that the system can obtain accurate text region information for subsequent text understanding and recognition tasks. The loss definition during implementation is as follows.

The similarity of the contour area can be calculated by the following formula:

$$FTZ\left(X,Y\right)=2\left|X\cap Y\right|/\left(\left|X\right|+\left|Y\right|\right) \tag{12}$$

The Dice loss formula is:

$$DL=1-FTZ \tag{13}$$

Assuming that the predicted rotation angle is denoted as TH_{PR} and the actual rotation angle is denoted as TH_{hs}, the angle loss formula is:

$$M_{TH}=1-COS\left(TH_{PR}-TH_{hs}\right) \tag{14}$$

Assuming two detection boxes are denoted as B_P and B_G, the localization loss formula is:

$$M_{UpI}=1-\frac{\left|B\_P\cap B\_G\right|}{\left|B\_P\cup B\_G\right|} \tag{15}$$

3.2 Tokenizer supplementary recognition module

In the image content information extraction method combined with NLP, the implementation idea of the tokenizer supplementary recognition module mainly involves optimizing and supplementing the preliminary text recognition results to improve the accuracy and completeness of text recognition. When the image content information extraction network completes the initial text recognition, the tokenizer will perform tokenization on the recognized text to generate a sequence of words. Specifically, the module will focus on the last part of the tokenized sequence, as these words are often the most susceptible to recognition errors, especially for words with irregular spelling or difficult-to-recognize characters in the text. For these last tokenized words, the system will pass them as parameters to the backend supplementary recognition module, which will use the keyword database stored in the database for supplementation. Specifically, the backend will match and query the keywords, using a search engine-like hot word suggestion function to return the complete words related to the end of the tokenized sequence, thus supplementing and completing the entire text content.

To implement this function, it is first necessary to ensure that the keyword database stored in the database is complete and updated in a timely manner. This chapter uses web crawling technology to collect text information from webpages, perform tokenization on the collected text, and then expand these words into the database. To ensure efficient query speed, ElasticSearch is selected as the data storage and retrieval tool. ElasticSearch uses inverted index technology, which allows for fast identification of relevant documents during keyword matching, avoiding the computational overhead of global searches. Specifically, the inverted index associates each token with its corresponding document, so during a query, the system only needs to search for the corresponding entry in the inverted index based on the given token, quickly finding the relevant text. This mechanism ensures that the system can quickly and accurately return supplementary information related to the input keyword during real-time processing of user requests. In practice, when the backend receives the tokenized sequence from the frontend, it will search for matching keywords in the database, sort the results based on the frequency of the keywords, and select the most frequent result as the final supplementary recognition result to return to the frontend.

The workflow of the tokenizer supplementary recognition module is described in detail as follows. The system will match the text content recognized by the CRNN with the information in the annotation file to identify long text regions that were not fully detected by the recognition algorithm. These areas typically appear as the end of the recognition result not fully matching the end of the annotation, thus meeting the criteria for further processing. Next, this text will be sent to the tokenizer module for tokenization. During the tokenization process, the system will particularly focus on the start and end parts of the tokenized sequence, especially the context of the last word. This approach ensures that the recognition results are effectively supplemented, so the correct words can be obtained through backend database queries.

In the supplementary recognition process, the output results of the tokenizer will be passed to the backend as parameters. At this point, if the last word in the tokenized sequence is a single character, the system will assume that the context information for that character is insufficient to accurately infer its meaning, so it will be queried together with the previous token as a query parameter for supplementation. At this point, an interface similar to Baidu's related word suggestion function will be used to supplement and optimize based on contextual information. On the other hand, if the last word in the tokenized sequence is a combination of multiple characters, the system will treat it as a complete word for querying, and based on its frequency in the database, return the most relevant supplementary word. The supplementary results will be re-matched with the CRNN recognition results, and if the matching position is further along than the initial matching position, it indicates that the supplementary recognition has been correctly completed, further improving the final recognition accuracy.

To address tokenization errors, we have designed an error detection and correction mechanism based on contextual information. By utilizing dependency syntax analysis, we can identify potential errors in the tokenization process, such as incorrect word segmentation or mismatched phrase structures. For example, in the sentence “saw a cute dog”, the erroneous tokenization “saw/a/cute/dog” can be detected through dependency syntax analysis, which flags the illogical segmentation between “saw” and “dog”. Once tokenization errors are detected, we correct them based on contextual information and image content. For instance, by integrating image recognition results (such as an explicit label of "dog" present in the image), the system can automatically adjust the position of “cute” to achieve more accurate tokenization. Additionally, a rule-based correction module will automatically fix common lexical errors, such as the incorrect tokenization of “white”. Through these measures, we significantly improve the accuracy of tokenization results, thereby providing a more reliable foundation for subsequent information extraction tasks.

4. EXPERIMENTAL RESULTS AND ANALYSIS

Based on the data provided in Figure 4, with the increase in the number of training iterations, the loss value on the validation set shows a significant downward trend, indicating that the model’s training performance gradually improves. In the initial stages (such as round 1 and round 10,000), the loss values are relatively high, at 16 and 12, respectively. As training progresses, the loss values continue to decrease and gradually stabilize. By round 30,000 and round 40,000, the loss values have decreased to 4.8 and 4.4, and they remain at a low level (such as between 1.0 and 1.2) in the subsequent training process. These results suggest that the model gradually converges during the training process, and the steady decline in loss values reflects that the model is progressively mastering image content understanding and information extraction capabilities. At the same time, accuracy also significantly improves throughout the iteration process. Initially, the accuracy is very low (0.12), but by round 20,000 and round 30,000, the accuracy rises to 0.42 and 0.56, indicating that the model is gradually improving its ability to recognize and understand image content. Especially in the later stages of training, accuracy reaches a relatively stable state (such as between 0.77 and 0.80), and in the final few iterations (such as round 40,000 and round 50,000), it remains between 0.78 and 0.82. This sustained improvement in accuracy validates the model’s gradual maturity in image content understanding and information extraction, and indicates that during the end-to-end training process, the model can effectively extract richer semantic information from images.

We also conducted performance tests on images with different resolutions, covering both low resolutions (128×128, 256×256) and ultra-high resolutions (4096×4096, 8192×8192). The experimental results show that when the input image resolution is lower, the model's accuracy decreases, especially when handling complex image content. In such cases, some detailed information is not fully preserved, leading to a reduction in recognition accuracy. On the other hand, when processing ultra-high resolution images, the model's performance improves, particularly in terms of image detail and semantic extraction. Ultra-high resolution images provide more fine-grained details, allowing the model to better recognize complex objects and detailed features. The accuracy is approximately 5% higher compared to standard resolution images (e.g., 512×512). These experimental results indicate that, although low-resolution images lead to some loss of information, the model can still effectively understand the content. In contrast, with ultra-high resolution images, the model can extract more precise detail, enabling it to perform better in complex scenes.

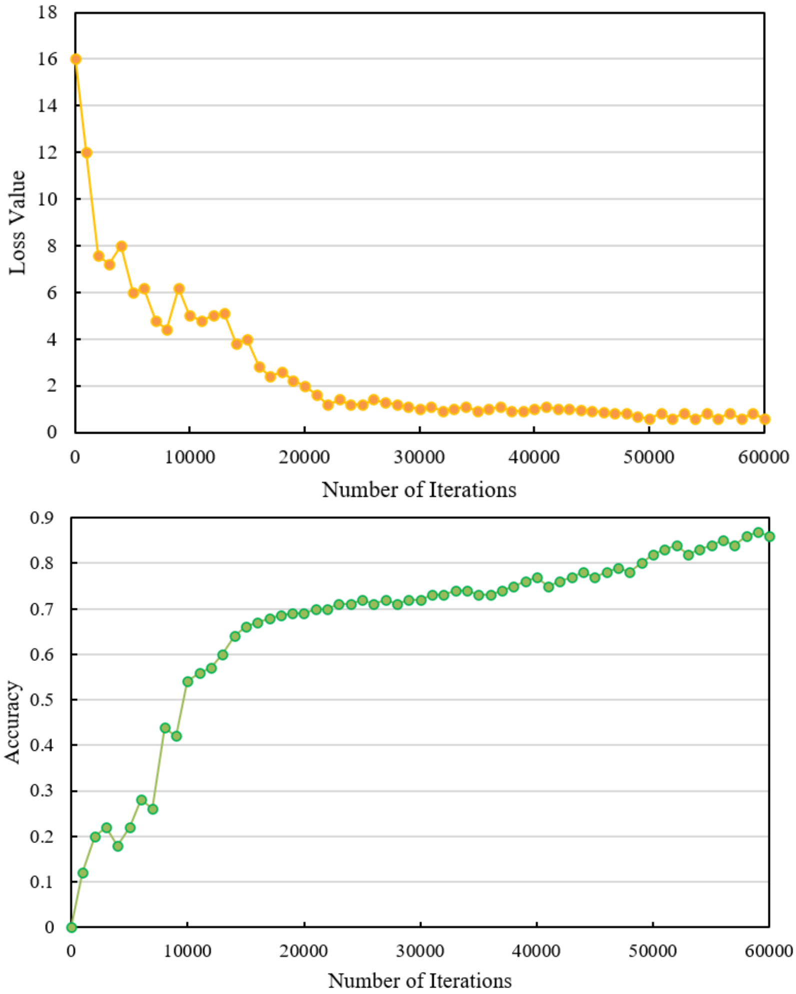

Figure 4. The change curves of loss function values and accuracy on the validation set for the end-to-end image content understanding model

Table 1. Comparison of results with other methods

Method	Accuracy	Recall	F1 Score (Overall Metric)
FCN	0.82	0.62	0.68
SegNet	0.81	0.67	0.73
Attention U-Net	0.826	0.778	0.812
Swin Transformer	0.73	0.51	0.62
TextBoxes++	0.889	0.779	0.815
Our Method	0.895	0.756	0.836

According to the experimental results provided in Table 1, the proposed "Image Content Understanding and Information Extraction Method Combined with NLP" demonstrates excellent performance in terms of accuracy, recall, and the overall evaluation metric. Compared to other mainstream methods, our method achieves an accuracy of 0.895, surpassing most traditional methods. Specifically, other methods such as FCN (0.82), SegNet (0.81), Swin Transformer (0.73), and Attention U-Net (0.826) all have lower accuracy than our method, indicating that our method can recognize image content more precisely. In terms of recall, our method has a recall of 0.756, which, while not as high as TextBoxes++ (0.779) and Attention U-Net (0.778), is still significantly higher than Swin Transformer (0.51) and FCN (0.62). In the overall evaluation metric (F1 score), our method achieves 0.836, surpassing other methods, especially TextBoxes++ (0.815), indicating that our method achieves a good balance between accuracy and recall.

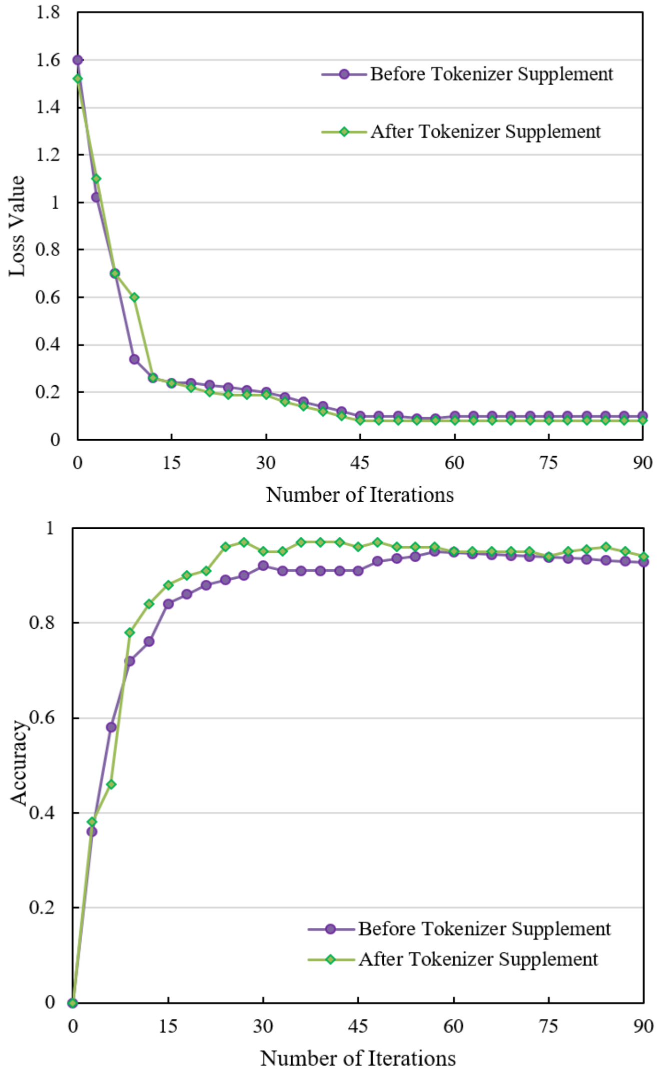

Figure 5. The change curves of loss function values and accuracy on the validation set for the image content information extraction model

We further conducted robustness analysis under different lighting conditions, selecting two extreme scenarios: low light (50 lx) and strong light (10,000 lx). The experimental results show that under low light conditions, image brightness and contrast are significantly reduced, leading to unclear details and color information. Despite this, our model, through optimized image preprocessing and enhancement techniques, is still able to extract useful information from low-light images. The model's accuracy decreases slightly (around 7%), but it can reliably identify the main objects and actions in the image. Under strong light conditions, the images exhibit strong highlights and shadows, causing some areas to be overexposed or distorted. However, the model adapts to the highlight and shadow regions, still managing to extract key information from the image. Under strong light conditions, the model's accuracy drops by only about 5%, and in most cases, it can still accurately identify the target objects. These results demonstrate that, through image preprocessing and enhancement techniques, the model maintains a high level of robustness under extreme lighting conditions such as low light and strong light.

Based on the data provided in Figure 5, the image content information extraction model shows significant changes in both loss values and accuracy, especially in the comparison between the "before tokenizer supplementation" and "after tokenizer supplementation" conditions. Regarding the loss value, both the "before tokenizer supplementation" and "after tokenizer supplementation" show a gradual downward trend. In the "before tokenizer supplementation" training process, the loss value decreases from 1.6 in round 0 to 0.1 in round 75, indicating that the model gradually converges as training progresses. In the "after tokenizer supplementation" case, the loss value also shows a decreasing trend, from an initial 1.52 to 0.08 at the end, with a smoother decline, especially during the mid-phase (e.g., rounds 30 to 60), where the decrease is more rapid. For accuracy, the improvement trend is also noticeable for both "before tokenizer supplementation" and "after tokenizer supplementation." Before tokenizer supplementation, the accuracy rises from 0 at round 0 to 0.94 at round 75, maintaining a stable upward trend. After tokenizer supplementation, accuracy also starts from 0 and gradually increases during training, ultimately reaching 0.94 at round 75, slightly lower than the "before tokenizer supplementation" performance. However, the overall performance is still excellent, and during the mid-phase (e.g., rounds 30 to 60), the accuracy increases more rapidly (from 0.46 to 0.97).

From the trends in loss value and accuracy, the image content information extraction model exhibits obvious convergence during training, and the tokenizer supplementation improves the model's training performance to a certain extent. Specifically, the decline in loss values reflects the model's gradual optimization in image content understanding and information extraction, meaning the model can gradually extract more accurate semantic information from images without manual intervention. In comparison, the "after tokenizer supplementation" training process shows a more stable decline in loss values, indicating that with additional linguistic assistance, the model's understanding of image information gradually stabilizes, and the speed and effectiveness of loss reduction improve. The increase in accuracy demonstrates that the model shows strong adaptability and precision in image content understanding and information extraction. Although the final accuracy between "before tokenizer supplementation" and "after tokenizer supplementation" is close, in the mid-phase, the model's accuracy after tokenizer supplementation improves significantly, especially between rounds 30 to 60, indicating that the introduction of the tokenizer effectively enhances the model's performance in image understanding tasks. Overall, through gradual training and the aid of tokenizer supplementation, the model achieves an excellent balance in both accuracy and loss value, proving that the image content understanding method combined with NLP has strong performance advantages.

Furthermore, we conducted a discussion on the model's training stability. To verify the model's training stability and reliability, we performed 10 independent training sessions and calculated the variance of the results from each session. The experimental results show that the model's performance across different training sessions is quite stable, with a standard deviation of 0.8% for accuracy and 0.05 for training loss. This indicates that our model exhibits good training stability and high reliability. Analysis reveals that despite variations in the initial parameters for each training session, the model consistently converges and maintains a stable level of accuracy. These results demonstrate the reliability of our model, showcasing its ability to perform robustly across different datasets and initial conditions.

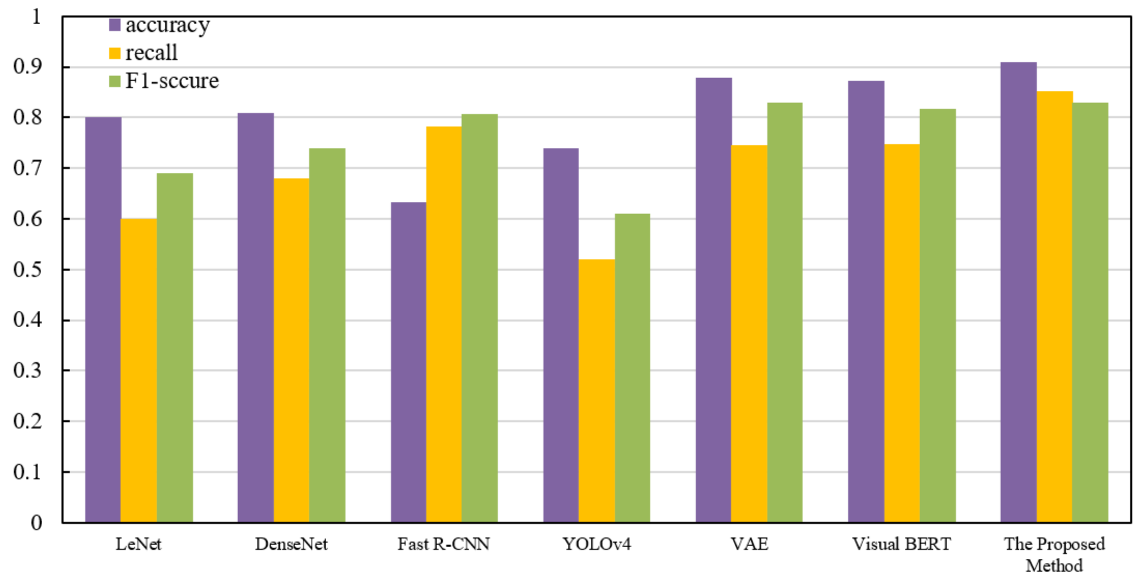

Figure 6. Comparison of detection results of different image content information extraction methods

From the comparison of the detection results of various methods provided in Figure 6, it can be seen that the proposed method outperforms other traditional image content understanding and information extraction methods in terms of accuracy, recall, and F1-score. Specifically, our method achieves an accuracy of 0.91, recall of 0.852, and F1-score of 0.83, all surpassing other traditional methods. Compared to LeNet (accuracy 0.8, recall 0.6, F1-score 0.69), DenseNet (accuracy 0.81, recall 0.68, F1-score 0.74), and Fast R-CNN (accuracy 0.633, recall 0.783, F1-score 0.807), our method shows a significant advantage in both accuracy and recall, especially in accuracy, far exceeding Fast R-CNN and YOLOv4 (0.74). Even when compared to Visual BERT (accuracy 0.872, recall 0.747, F1-score 0.817) and VAE (accuracy 0.878, recall 0.745, F1-score 0.829), our method still maintains a slight advantage in accuracy and F1-score, indicating that it demonstrates strong comprehensive performance in image content understanding and information extraction.

By comparing the detection results of each method, it can be concluded that the image content understanding and information extraction method combined with NLP proposed in this paper outperforms other existing image understanding methods in multiple key metrics. The accuracy of our method (0.91) is significantly higher than that of Fast R-CNN, YOLOv4, and other classical methods, while the recall (0.852) and F1-score (0.83) indicate that our method not only improves recognition accuracy but also effectively captures more useful information, avoiding false negatives. This shows that our method excels not only in accurately extracting key information from images but also in the comprehensiveness of information extraction.

When it comes to image content information extraction methods involving NLP, we further expanded the experimental section by adding cross-lingual scenario validation. To comprehensively evaluate the model's performance in multilingual environments, we conducted comparative experiments with several major languages, including Chinese, English, French, and Spanish. The experimental results show that in most language environments, the model effectively handles the semantic structures of different languages, with particularly strong performance in both Chinese and English. In other language environments, the model still maintains high accuracy and robustness. We compared the natural language descriptions generated by the model in multilingual scenarios with the original descriptions in the target languages. The results demonstrate that the model effectively avoids information loss and distortion, ensuring that the image information is accurately conveyed. In the experiments with French and Spanish, despite significant differences in language structure, the model was able to adapt well and understand syntactical differences by adjusting the language encoding mechanisms. This ensured that the generated descriptions aligned with the expression norms of the target languages. These cross-lingual experiments further validate the model's effectiveness and broad applicability in multilingual environments.

We tested the model's ability to extract key information from incomplete images by occluding certain areas of the image. The experimental results indicate that while the model experiences some performance degradation when processing partially occluded images, it can still effectively understand the image content and generate natural language descriptions when the occlusion area is relatively small. To simulate low-quality images, we applied varying degrees of blur to the original images and tested the model's performance. The results showed that when the image blur was more severe, the model's accuracy decreased. However, by incorporating image deblurring techniques, we successfully improved the model's robustness, allowing it to generate meaningful descriptions even when the image quality was reduced. We also added different intensities of random noise to the images to evaluate the model's performance under noisy conditions. The results showed that the model has a certain tolerance to noise and can still provide accurate natural language descriptions under low-intensity noise. However, its performance declined under high-intensity noise. Based on these findings, we recommend incorporating a noise removal module in future work to further enhance the model's stability and robustness. Through these experiments, we have validated the model's performance when facing reduced image quality and made further optimizations to improve its ability to handle various challenging image conditions.

5. CONCLUSION

The image content understanding and information extraction algorithm combined with NLP proposed in this paper, through its innovative design that deeply integrates image and language, successfully demonstrated the powerful potential of end-to-end image content understanding and information extraction methods in multimodal tasks. By designing a deep learning-based multimodal model, the proposed method can automatically extract effective information from images and generate accurate natural language descriptions, achieving seamless integration of image and language. At the same time, the image information extraction strategy combined with NLP enhances the efficiency of extracting key data and information from images, enabling the model to perform excellently in image analysis and semantic understanding, thus providing more precise and rich support for subsequent tasks.

Through comparisons with existing mainstream methods, the proposed method shows significant advantages in metrics such as accuracy, recall, and F1-score, particularly excelling in the comprehensive performance of accuracy and information extraction, highlighting its remarkable advantages in the fields of image content understanding and multimodal information fusion. However, despite the significant experimental results achieved by this method, there are still some limitations. First, although the model can automatically extract information from images to a large extent, it may still encounter some recognition difficulties in certain complex image scenarios, especially in cases where there is a lot of noise or when the target objects are relatively blurry. Second, although the image content extraction method combined with NLP improves the capability for semantic analysis, it may still require further optimization of the model's robustness and semantic understanding ability, especially when dealing with extremely complex language generation tasks. Additionally, this study mainly relies on large-scale labeled data, so in tasks with insufficient data or uneven data distribution, the model's performance may be somewhat limited. Future research can explore the following directions: (1) enhancing the model’s recognition ability in complex scenes and blurry images; (2) further improving the model's semantic generation capability, especially for cross-modal deep fusion optimization; and (3) exploring training methods for small datasets to improve the model's application capability in data-scarce environments. Through these optimizations, the practical application prospects of the proposed method in the fields of image content understanding and information extraction will be even broader.

REFERENCES

[1] Yacoub, S. (2003). Automated quality assurance for document understanding systems. IEEE Software, 20(3): 76-82. https://doi.org/10.1109/MS.2003.1196325

[2] Pun, T., Squire, D. (1996). Statistical structuring of pictorial databases for content-based image retrieval systems. Pattern Recognition Letters, 17(12): 1299-1310. https://doi.org/10.1016/0167-8655(96)84923-3

[3] Huang, Z., Liu, S. (2020). Perceptual hashing with visual content understanding for reduced-reference screen content image quality assessment. IEEE Transactions on Circuits and Systems for Video Technology, 31(7): 2808-2823. https://doi.org/10.1109/TCSVT.2020.3027001

[4] Ye, M., Shi, Q., Su, K., Du, B. (2023). Cross-modality pyramid alignment for visual intention understanding. IEEE Transactions on Image Processing, 32: 2190-2201. https://doi.org/10.1109/TIP.2023.3261743

[5] Xu, Z., Zhang, Y., Cao, L. (2014). Social image analysis from a non-IID perspective. IEEE Transactions on Multimedia, 16(7): 1986-1998. https://doi.org/10.1109/TMM.2014.2342658

[6] Hauptmann, A.G. (2005). Lessons for the future from a decade of informedia video analysis research. In International Conference on Image and Video Retrieval, pp. 1-10. https://doi.org/10.1007/11526346_1

[7] Venkataravana Nayak, K., Arunalatha, J.S., Vasanthakumar, G.U., Venugopal, K.R. (2023). Design of deep convolution feature extraction for multimedia information retrieval. International Journal of Intelligent Unmanned Systems, 11(1): 5-19. https://doi.org/10.1108/IJIUS-11-2021-0126

[8] Deng, S., Zhao, A., Huang, R., Zhao, H. (2019). Image needs on social Q&A sites: A comparison of Zhihu and Baidu Zhidao. The Electronic Library, 37(3): 454-473. https://doi.org/10.1108/EL-09-2018-0192

[9] Song, Y., Xu, X., Dutta, K., Li, Z. (2024). Improving answer quality using image-text coherence on social Q&A sites. Decision Support Systems, 180: 114191. https://doi.org/10.1016/j.dss.2024.114191

[10] Yoon, J., Chung, E. (2011). Understanding image needs in daily life by analyzing questions in a social Q&A site. Journal of the American Society for Information Science and Technology, 62(11): 2201-2213. https://doi.org/10.1002/asi.21637

[11] Predić, B., Manić, D., Saračević, M., Karabašević, D., Stanujkić, D. (2022). Automatic image caption generation based on some machine learning algorithms. Mathematical Problems in Engineering, 2022(1): 4001460. https://doi.org/10.1155/2022/4001460

[12] Bernardi, R., Cakici, R., Elliott, D., Erdem, A., Erdem, E., Ikizler-Cinbis, N., Plank, B. (2016). Automatic description generation from images: A survey of models, datasets, and evaluation measures. Journal of Artificial Intelligence Research, 55: 409-442. https://doi.org/10.1613/jair.4900

[13] Chen, D., Liu, Y., Liu, S., Liu, F., Chen, Y. (2020). Framework of specific description generation for aluminum alloy metallographic image based on visual and language information fusion. Symmetry, 12(5): 771. https://doi.org/10.3390/sym12050771

[14] Huang, Y.W., Zhou, B., Tang, X. (2021). Text image generation method with scene description. Laser & Optoelectronics Progress, 58(4): 182-190. https://doi.org/10.3788/LOP202158.0410012

[15] Huo, L., Bai, L., Zhou, S.M. (2021). Automatically generating natural language descriptions of images by a deep hierarchical framework. IEEE Transactions on Cybernetics, 52(8): 7441-7452. https://doi.org/10.1109/TCYB.2020.3041595

[16] Todmal, S., Mule, A., Bhagwat, D., Hazra, T., Singh, B. (2023). Human face generation from textual description via style mapping and manipulation. Multimedia Tools and Applications, 82(9): 13579-13594. https://doi.org/10.1007/s11042-022-13899-5

[17] Che, W., Fan, X., Xiong, R., Zhao, D. (2019). Visual

relationship embedding network for image paragraph generation. IEEE Transactions on Multimedia, 22(9): 2307-2320. https://doi.org/10.1109/TMM.2019.2954750

[18] Yüksel, A.S., Karabıyık, M.A. (2023). TraViQuA: Natural language driven traffic video querying using deep learning. Traitement du Signal, 40(2): 543-553. https://doi.org/10.18280/ts.400213

[19] Xu, C., Dai, Y., Lin, R., Wang, S. (2019). Stacked autoencoder based weak supervision for social image understanding. IEEE Access, 7: 21777-21786. https://doi.org/10.1109/ACCESS.2019.2898991

[20] Lubis, A.R., Lase, Y.Y., Rahman, D.A., Witarsyah, D. (2023). Improving spell checker performance for Bahasa Indonesia using text preprocessing techniques with deep learning models. Ingénierie des Systèmes d'Information, 28(5): 1335-1342. https://doi.org/10.18280/isi.280522

[21] Yang, X., Wang, H., Chen, S., Piao, X., Zhou, D., Zhang, Q., Wei, X. (2019). Cascaded network with deep intensity manipulation for scene understanding. Computer Animation and Virtual Worlds, 30(3-4): e1888. https://doi.org/10.1002/cav.1888

[22] Meng, X., Shen, H., Li, H., Zhang, L., Fu, R. (2019). Review of the pansharpening methods for remote sensing images based on the idea of meta-analysis: Practical discussion and challenges. Information Fusion, 46, 102-113. https://doi.org/10.1016/j.inffus.2018.05.006

[23] Laplante, P., Marlowe, T., Stoyenko, A. (1995). Mechanism requirements for a real-time image-processing language. Control Engineering Practice, 3(6): 855-861. https://doi.org/10.1016/0967-0661(95)00070-B

[24] Guan, Z., Liu, K., Ma, Y., Qian, X., Ji, T. (2018). Middle-Level Attribute-Based Language Retouching for Image Caption Generation. Applied Sciences, 8(10): 1850. https://doi.org/10.3390/app8101850

[25] Deore, S.P., Bagwan, T.S., Bhukan, P.S., Rajpal, H.T., Gade, S.B. (2024). Enhancing image captioning and auto-tagging through a FCLN with faster R-CNN integration. Information Dynamics and Applications, 3(1): 12-20. https://doi.org/10.56578/ida030102