International Journal of Innovative Computing, Information and Control
Volume **22**, Number **2**, April **2026**

ICIC International ©2026 ISSN 1349-4198
pp. **533–547**

ALGORITHM FOR BOOK SPINE SEGMENTATION AND MATCHING BASED ON DEEP LEARNING

Lifu Hu, Sixu Zhao, Lirong Tang, Xiaofei Ji* and Kexin Zhang

College of Automation
Shenyang Aerospace University
No. 37, Daoyi South Avenue, Shenbei New Area, Shenyang 110136, P. R. China
{ hulifu; zhaosixu; tanglirong; zhangkexin }@stu.sau.edu.cn
*Corresponding author: jixiaofei@stu.sau.edu.cn

Received May 2025; revised August 2025

Abstract. *With the advent of smart libraries, real-time book positioning based on computer vision has gained increasing attention. These systems typically rely on segmenting book spines from shelf images and matching them to a reference database. However, due to the complexity of library environments, challenges remain in achieving high accuracy, robustness, and efficiency. This paper proposes a deep learning-based framework for accurate and efficient book spine segmentation and matching. For segmentation, an enhanced DeepLabv3 plus network is developed, where the standard Atrous Spatial Pyramid Pooling (ASPP) module is replaced with Dense Atrous Spatial Pyramid Pooling (DenseASPP) to capture richer multi-scale features. Strip Pooling is introduced to better extract elongated spine structures, while a self-attention mechanism enhances global context awareness. For matching, a deep feature matching algorithm is designed using VGG16 for feature extraction and Euclidean distance for similarity computation. The Facebook AI Similarity Search (Faiss) framework is integrated to accelerate large-scale retrieval. Experiments were conducted on two datasets constructed from segmented spine images: one comprising books from the same series and the other from different series. The proposed method achieved over 95% matching accuracy with an average processing time of 2.33 seconds per sample on both datasets, demonstrating strong robustness and real-time potential.*
Keywords: Book spine segmentation and matching, Smart library, DeepLabv3 plus, DenseASPP, Faiss

1. **Introduction.** With the rapid advancement of the information society, libraries, which serve as the core platforms for knowledge dissemination and information management, are increasingly confronted with the dual challenges of expanding operational scale and growing collection resources. Traditional retrieval methods relying on manual search or barcode scanning have become insufficient to meet readers' demands for efficient and convenient book localization. Against this backdrop, computer vision-based automatic book recognition and positioning technologies have garnered significant attention and demonstrated promising potential in applications such as inventory management, mis-shelving detection, and intelligent navigation. Since only the spines of shelved books are typically visible, the information available in images is extremely limited, rendering accurate spine segmentation and efficient matching critical components of visual positioning systems. The former directly affects the accuracy of subsequent recognition and classification, while the latter determines the system's robustness in complex environments and its response efficiency in practical applications. Therefore, developing a spine segmentation and

DOI: 10.24507/ijicic.22.02.533

matching algorithm that balances accuracy and real-time performance is of paramount importance for advancing the intelligent upgrading of book management systems.

In recent years, considerable research efforts have been devoted to the segmentation and matching of book spines on shelves. Traditional spine segmentation methods predominantly rely on morphological processing, line detection, and classification techniques based on Hough transform and Support Vector Machine (SVM). For instance, Tabassum et al. [1] proposed a multi-row book segmentation approach that performs well under orderly arrangements but exhibits limited adaptability to thin books and cluttered placements. Nevetha and Baskar [2] designed a heuristic line detector to extract spine edges. Chen et al. [3] further enhanced segmentation robustness by integrating text recognition with image features. However, these approaches depend heavily on handcrafted features and suffer from limited segmentation performance and robustness when confronted with highly similar spine textures, densely packed arrangements, as well as challenging lighting conditions, tilting, and occlusions. With the rapid advancement of deep learning, Convolutional Neural Networks (CNNs) have demonstrated significant advantages in image segmentation tasks. Zhu et al. [4] empirically showed that CNNs possess superior generalization capabilities compared to traditional SVMs. Zhou et al. [5] employed Mask R-CNN combined with ResNet and Feature Pyramid Network (FPN), substantially improving detection accuracy for multi-scale targets. Nevertheless, the model's complex architecture and high computational demands limit its feasibility for real-time applications. Zeng et al. [6, 7] proposed a deep network structure integrating boundary information and rotation-invariant feature extraction to effectively enhance detection of tilted spines; however, this method relies on high-quality boundary annotations and involves a complex training process, hindering its scalability to diverse and annotation-scarce real-world scenarios.

In the realm of spine matching, traditional research primarily relies on handcrafted image feature extraction and matching strategies. Lee et al. [8] introduced the Color Difference of Gaussians (CDoG) algorithm, which demonstrates clear advantages over the conventional grayscale DoG in terms of feature quantity and stability, underscoring the efficacy of color information in spine matching. Building upon this, Fowers and Lee [9] proposed the CDoG-SIFT algorithm that integrates color features with scale invariance, effectively mitigating edge blurring issues present in grayscale images and enabling the recognition of misplaced and missing books. Chen et al. [10] developed a mobile book recognition system employing Hough transform to extract line features combined with segmentation-accelerated robust features (segSURF) for matching. This approach overcomes dependency on the book arrangement order and achieves edge deployment in practical scenarios; however, its matching accuracy remains limited at 74.5%. Although these methods exhibit adaptability in specific scenarios, they commonly rely on low-level image features, rendering them susceptible to disturbances from similar spine textures, lighting variations, and occlusions. Consequently, they struggle to achieve fast and accurate matching in large-scale, densely packed book environments. Moreover, traditional feature matching incurs substantial computational costs when handling high-dimensional image databases and lacks semantic understanding and abstract modeling capabilities. Therefore, how to enhance retrieval efficiency and robustness in large-scale spine repositories while maintaining matching accuracy remains a critical challenge demanding further breakthroughs in spine matching research.

In summary, although existing studies have made notable progress in spine image segmentation and matching, several critical challenges persist in real-world applications: 1) tightly packed books with varying thicknesses result in significant aspect ratio differences

in spine images; 2) complex backgrounds and uneven illumination often cause false positives and missed detections; 3) highly similar textures and layouts among books of the same series complicate accurate boundary identification. To address these issues, this paper proposes a deep learning-based method for spine image segmentation and matching, with the following key contributions.

1) Spine Image Segmentation: We introduce three crucial improvements to the DeepLabv3 plus network – incorporating a DenseASPP module to enhance multi-scale feature extraction, thereby better adapting to books of different thicknesses; integrating Strip Pooling to strengthen the response to vertical textures through elongated receptive fields; and embedding a Multi-Head Self-Attention (MHSA) mechanism to improve the model's global context modeling capability and segmentation accuracy under complex backgrounds and irregular arrangements.

2) Spine Image Matching: We develop a deep feature matching approach based on the Faiss framework. By extracting high-dimensional feature vectors of spine images using VGG16 and leveraging approximate nearest neighbor search, the method achieves efficient large-scale feature comparison while balancing matching accuracy and retrieval speed, thus meeting practical deployment requirements for real-time performance and computational resource constraints.

2. **Book Spine Segmentation Model and Implementation.** The paper introduces a new network for segmenting book spines, based on the DeepLabv3 plus [11] framework. The network is improved by replacing the ASPP network with the DenseASPP structure and adding a Strip Pooling module. Furthermore, a self-attention mechanism is used in CNN networks based on Vision Transformer's [12] multi-head attention mechanism module.

2.1. **Book spine segmentation network model.** The proposed book spine segmentation network model, as illustrated in Figure 1, adheres to the original DeepLabv3 plus framework and uses MobileNetV2 [13] as the backbone network. Firstly, the book image (as shown in Figure 1) is input into the MobileNetV2 network for feature extraction. Subsequently, the feature maps of the middle three layers of the MobileNetV2 network

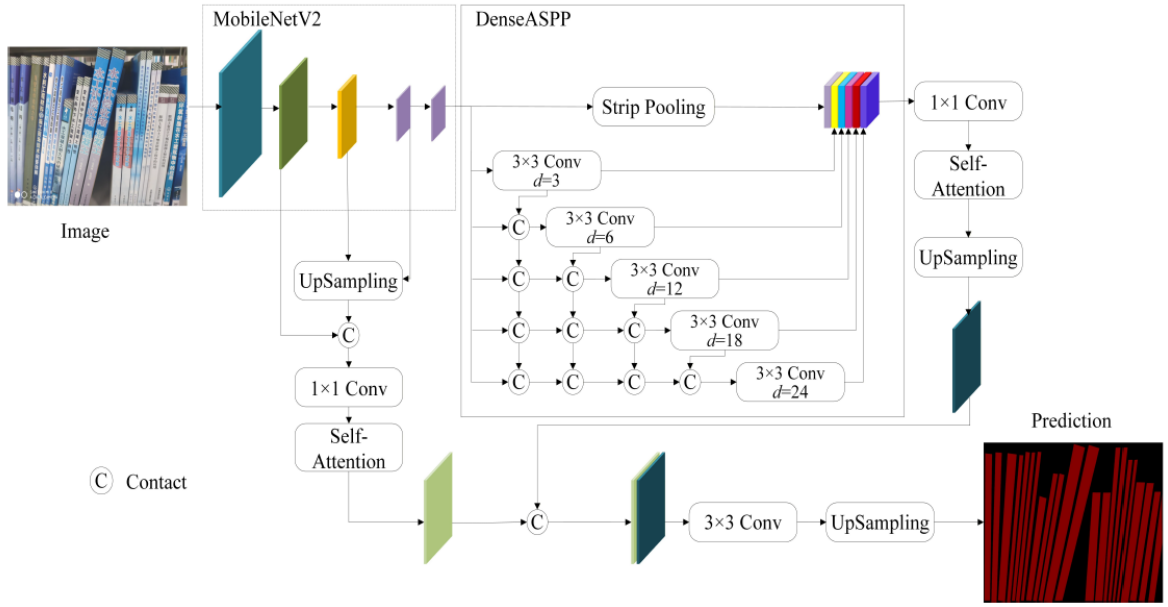


FIGURE 1. Book spine segmentation network model

are up-sampled and fused, and the resulting fusion is utilized as shallow features. Next, the output of the last layer of the MobileNetV2 network is directed to the DenseASPP module. During the encoding stage, the DenseASPP module replaces the traditional ASPP module, aiming to expand the receptive field and generate denser image features. In consideration of the large aspect ratio of book spine images, a stripe pooling module is specifically introduced within the DenseASPP module to better preserve long striped features, enhancing segmentation accuracy. After processing by the DenseASPP module, the features undergo 1×1 convolution to achieve channel compression. These compressed features are then inputted into the self-attention module to build deep features. In the decoding stage, shallow features undergo 1×1 convolution to adjust the number of channels. They are similarly directed to a self-attention module to enhance feature representation. The processed shallow features are then concatenated with the deep features. Finally, the final prediction result is obtained after two convolution operations and one upsampling operation. Figure 1 depicts this process.

2.2. **DenseASPP module.** For the intensive segmentation task of book images, the DenseASPP module [14] has been introduced to generate more dense features. Compared to parallel processing in ASPP, DenseASPP modules adopt a cascading approach to utilize the features of each layer. The dilation rate (denoted as "d" in Figure 1) increases layer by layer from top to bottom, and due to its small scale difference from the original image, feature fusion is not performed on it. Compared to the high-dilation-rate layer, the pixel separation is relatively wide during feature extraction. It is necessary to fuse the small dilation rate layer features to ensure that detailed information is not lost. Finally, the 5 feature maps of the same size will be concatenated and put into a 1×1 convolution to adjust the number of channels. After upsampling, the DenseASPP module output will be obtained. Its output is a feature map generated by convolution with multiple dilation rates and scales. Compared to the original ASPP module, DenseASPP brings two benefits: a denser feature pyramid and a larger receptive field.

2.3. **Strip Pooling module.** The introduction of the Strip Pooling (SP) module [15] in the network structure, as shown in Figure 2. The primary idea behind this module is to use a long Strip Pooling convolution kernel in the spatial dimension to enhance the ability to capture long-distance information, aiming to preserve the long strip features of the book spine. The formulas for horizontal and vertical pooling are as follows:

$$y_j^w = \frac{1}{H} \sum_{0 \le i \le H} x_{i,j} \tag{1}$$

$$y_j^h = \frac{1}{H} \sum_{0 \le j \le W} x_{i,j} \tag{2}$$

In Equations (1) and (2), H and W denote the height and width of the feature map, while $x_{i,j}$ represents the pixel value at the ith row and jth column within the feature map. To obtain the two parts within the graph box, apply Equations (1) and (2) of average Strip Pooling to the local feature values of the pixels within the input tensor. Subsequently, use one-dimensional convolution on each part, upsample the obtained results to the size of the input tensor, and then perform feature fusion. After the 1×1 convolution and Sigmoid operation, the result is pixel-wise multiplied with the input tensor to yield the output tensor. Consequently, each position in the output tensor corresponds to a position in the input tensor. The square bounded by a red box in the output tensor is connected to all positions with the same horizontal or vertical positions, thereby preserving the long strip information.

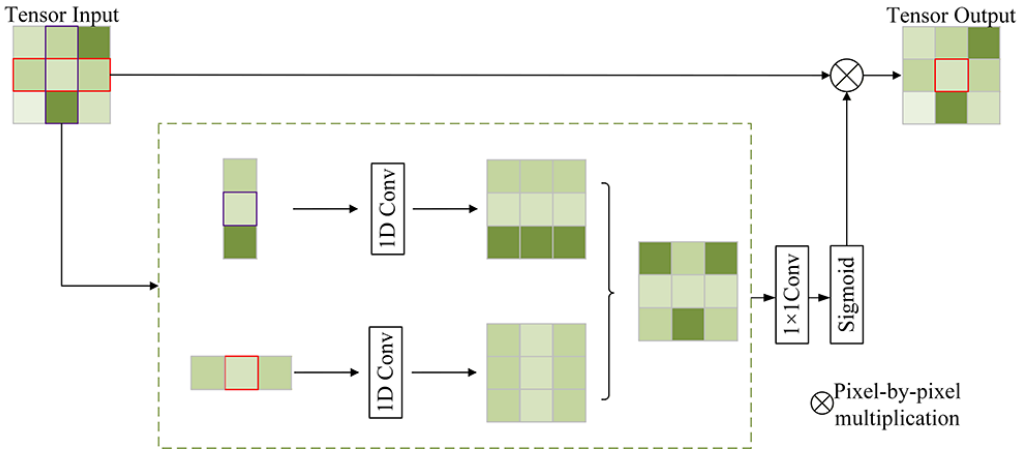


FIGURE 2. Strip Pooling module

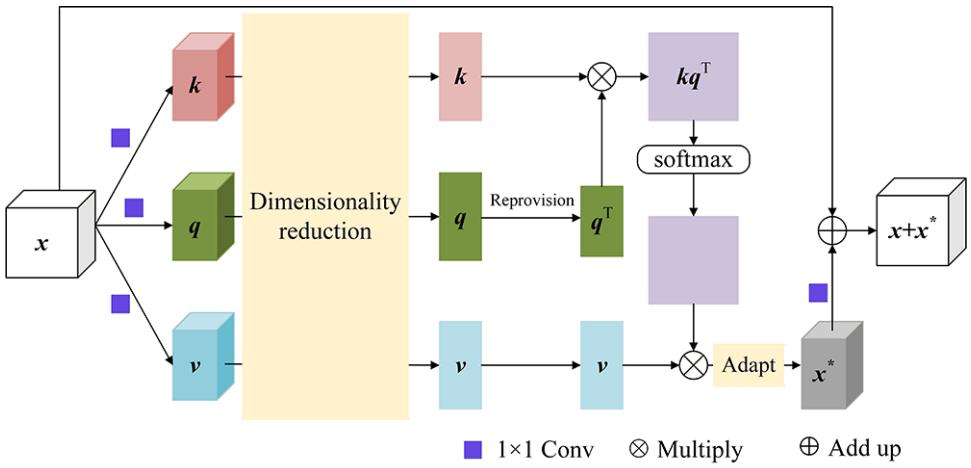


FIGURE 3. Self-attention mechanisms

2.4. **Global feature association attention module.** A self-attention mechanism has been introduced to enhance the CNN's ability to extract global information, as shown in Figure 3. DeepLabv3 plus generates deep and shallow features separately for use in the self-attention module. As an example, the deep feature channel is first adjusted by 1×1 convolution, and then k, v, and q are generated by the module to facilitate spreading and dimensionality reduction. After this, the resulting feature map is obtained by multiplying k with the transposed q, resulting in a size of $[W \times H, W \times H]$. Subsequently, the softmax module is applied, followed by multiplication with v to yield a feature size of $[C/N, W \times H]$. The final feature map is then obtained by processing this through 1×1 convolution and adding it to x. Inputs for this network are features, with the purpose of enhancing global information.

2.5. **Spine matching model architecture.** To improve book spine matching accuracy and speed, a deep feature matching algorithm based on the Faiss framework is designed [16]. This algorithm utilizes deep features to define the feature space, employs Euclidean distance as the matching metric, and introduces the Faiss model to optimize the search space and search strategy, thereby enhancing the speed of the matching algorithm. The proposed deep feature matching model based on the Faiss framework is shown in Figure 4. Specifically, the network is divided into three processes:

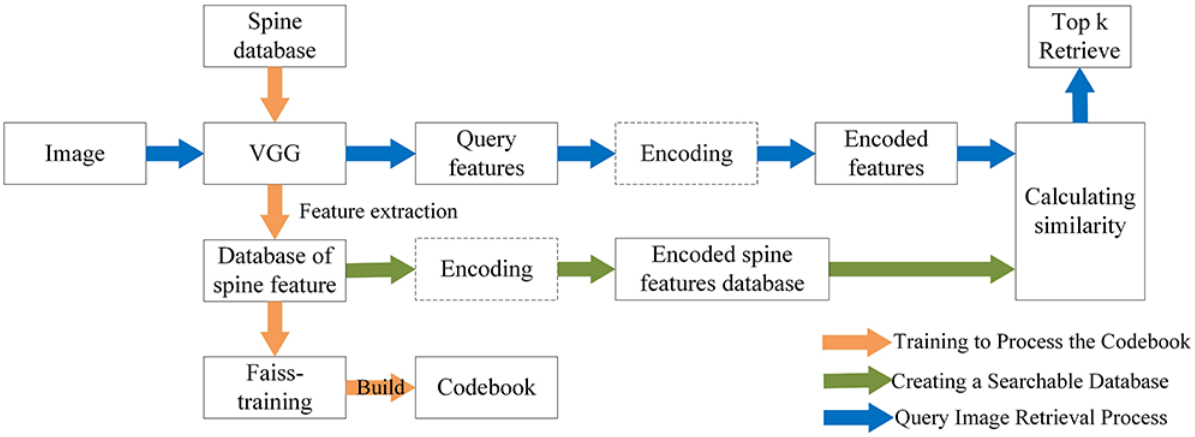


FIGURE 4. Deep feature matching model based on Faiss framework

1) Obtain codebook: As shown by the yellow arrows in Figure 4, the VGG model is used to extract all book spine features in the database, and the feature codebook is obtained through the Faiss training process;

2) Create a retrieval database: As shown by the green arrows in Figure 4, use the feature codebook obtained in the previous step to encode the features of the book spine database, and create an encoded book spine feature database;

3) Image retrieval: As shown by the blue arrows in Figure 4, the query image is input into the VGG model to obtain the query features. Then, the feature codebook obtained in step 1) is used for encoding. Finally, calculate the similarity between the encoded query features and the encoded book spine feature database to obtain the final matching result. Faiss uses Inverted File Product Quantization (IVF-PQ) for faster matching.

2.6. **Spine matching algorithm based on Faiss architecture.** Faiss (Facebook AI Similarity Search) is an open-source library developed by Facebook for efficient similarity search, supports fast retrieval of similar vectors from large-scale datasets, and has very high performance and scalability. Further, Inverted File Product Quantization (IVF-PQ) is one of the algorithms accelerated by the Faiss model. The following is a detailed description of its specific implementation.

2.6.1. *Product quantization.* Product quantization is based on clustering. The process of generating codebooks and quantization through Product Quantization (PQ) is shown in Figure 5.

The book spine image is put into the VGG16 network to obtain $N \times 512$ dimensional deep features. N is the number of training samples and 512 is the feature dimension. Based on the above training data, Faiss training includes two steps: cut clustering and quantization encoding. In the cut clustering stage, the sample space is divided into 4 subspaces. Each sub vector in each subspace is clustered using K-Means, with 256 cluster centers. After the above clustering operation, each sub vector can obtain a cluster center, which corresponds to one or more sub vectors. This results in 256×4 cluster centers, as shown in the gray part of Figure 5. In the quantization encoding stage, the purpose is to transform the original 512 dimensional vector into a 4-dimensional vector. Firstly, the sample vector is divided into 4 segments, and then the nearest cluster center to each sub vector is found in the corresponding subspace. The cluster center ID is used as an encoding result. In this way, each of the four sub vectors can find four cluster center IDs in their respective subspaces. This achieves a dimensionality reduction from 512 to 4, achieving quantization encoding.

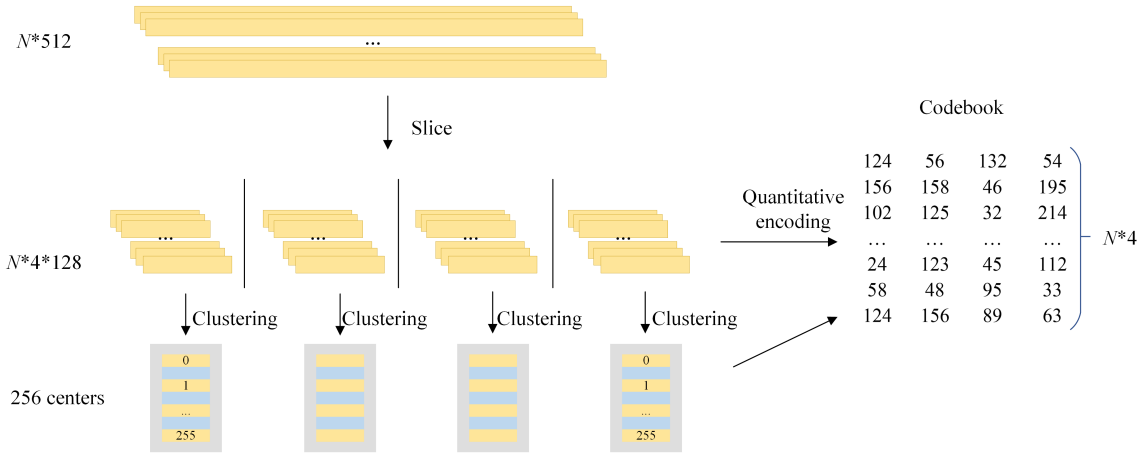


FIGURE 5. PQ product quantization

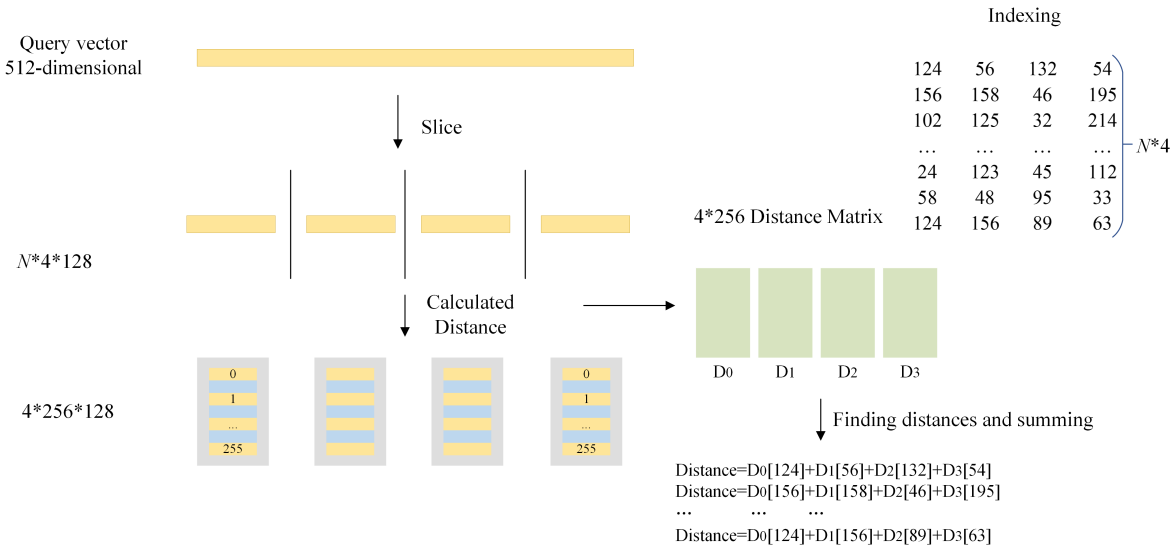


FIGURE 6. Matching of query samples

In the process of querying sample matching, PQ uses indirect approximation to replace the distance between sample vectors for matching calculation. With the sample distance calculation method, the distance between the query vector and the known vector is approximated by using the distance encoded between the query vector and the quantized known vector, which is closer to the true distance between the two samples. The specific matching process is shown in Figure 6.

When inputting the query vector, it is first divided into the same number of sub vectors, and then the distance between each sub vector and the 256 cluster centers in the corresponding subspace is calculated to obtain a distance matrix D of size 4×256. Next, calculate the distance between the query vector and the known vector in the database by looking up the table and summing the distances.

In summary, PQ product quantization can accelerate indexing due to the fact that it greatly reduces the number of distance calculations between query samples and database samples. Taking the above process as an example, traditional brute-force search methods need to perform N distance calculations to achieve 1-to-N matching. However, after PQ encoding, only $4 \times 256 = 1024$ operations are required. This reduces computing time.

2.6.2. *Inverse product quantization.* The previous text described that PQ product quantization needs to calculate the distance between the four segmented sub vectors of the query sample and their corresponding 256 cluster centers during the query process. Such brute-force calculation of the distance between all cluster centers and sub vectors still wastes a lot of time. IVF-PQ can quickly lock the region of interest before the query process through clustering algorithms, reducing unnecessary global calculations and sorting. The process of inverse product quantization is shown in Figure 7.

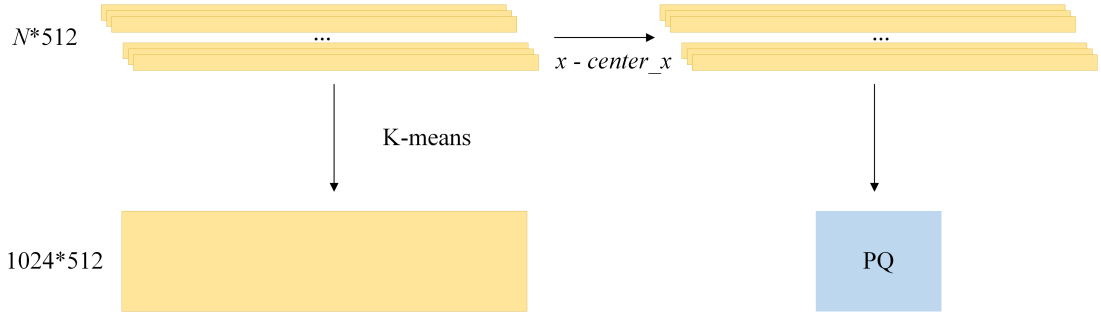


FIGURE 7. Inverse product quantization process

Prior to the PQ product quantization, a coarse quantization process is added. Specifically, K-Means [17] clustering is applied to N training samples first, and the number of clusters should generally not be set too large to achieve the clustering process quickly. After obtaining the cluster center, for each sample x, find the nearest cluster center x_c, and subtract the two to obtain the residual vector $(x - x_c)$ of the sample.

2.6.3. *The PQ product quantization process for $(x - x_c)$ follows.* In the same process as described above for the PQ product quantization distance calculation, coarse quantization allows query vectors to be instantly identified by class.

3. **Experimental Testing and Analysis.** The book spine database is sourced from an open-source database referenced in [6]. The dataset consists of 661 book images, including different shooting angles and tilted angles. To facilitate subsequent algorithm testing, the images are divided into two groups based on their tilt status. Specifically, there are 283 tilted images and 378 images with near-vertical angles (tilted angle less than 5°). The test set and training set are partitioned in a 1 : 3 ratio.

3.1. Analysis and verification of segmentation algorithms.

3.1.1. *DenseASPP validation.* DenseASPP module and ASPP module are used in the DeepLabv3 plus network framework to achieve segmentation results of 91.2% and 89.3%, respectively. The accuracy of this network segmentation is improved by 0.9% after replacing the ASPP module with the DenseASPP module, proving its advantages.

In order to reduce the complexity of the model, this paper selects convolutional kernels of size 3 with different dilation rates to form dilated convolutional layers, and cascades between different layers. Considering the impact of the number of network layers in the DenseASPP module on segmentation performance, the following experiments are conducted. The experimental results are shown in Table 1.

Table 1 shows that network models' accuracy depends on the number of network layers. When the number of network layers is low, there is less detailed information, features are not obvious, and accuracy is low. However, when the number of network layers is high, overfitting may occur, leading to accuracy decreases. Therefore, selecting the appropriate

TABLE 1. Effect of DenseASPP module network layers on segmentation

Test	Network layers	$I_{\mathrm{MoU}}/\%$
1	3	79.4
2	4	88.5
3	5	91.2
4	6	89.9

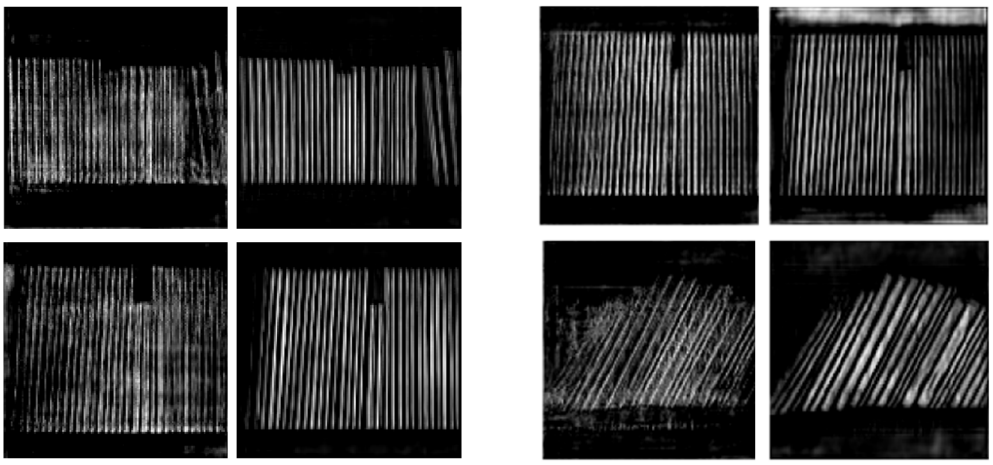
convolutional layer is necessary to ensure segmentation accuracy. The number of network layers in this paper is set to 5.

3.1.2. *Self-attention module validation.* Based on the original framework of the DeepLabv3 plus network (using the DenseASPP module instead of the ASPP module), the experiments are conducted on two backbone networks, Xception and MobileNetV2, respectively. Table 2 shows the results of adding only the self-attention module to the DeepLabv3 plus network.

TABLE 2. Comparative experimental results before and after the introduction of the self-attention module

Backbone	Self-attention module	$I_{\mathrm{MoU}}/\%$
Xception	✗	92.2
	✓	92.7
MobileNetV2	✗	93.1
	✓	93.8

Figure 8(a) in the left shows the effect of the self-attention module on book spine contextual features before and after its introduction, respectively. Compared with the left image in Figure 8(a), the right image shows clearer book spine features. Based on the results presented in Table 2 and Figure 8(a), it is evident that the accuracy rate consistently improves regardless of which backbone is used. This demonstrates the effectiveness of the self-attention module in correlating global information and its significant role in segmentation.



(a) Comparison between no (left) and with (right) self-attention module

(b) Comparison between no (left) and with (right) Strip Pooling modules

FIGURE 8. Effective validation of self-attention module and Strip Pooling module

3.1.3. *Effective validation of Strip Pooling module.* To evaluate the contribution of the Strip Pooling module to book spine segmentation, we compared the segmentation results with and without the Strip Pooling module when using the original DeepLabv3 plus network framework (where the standard ASPP module was replaced by the DenseASPP module). Figure 8(b) shows the visualization results after the fusion of deep and shallow features, with the left image depicting the outcome without the Strip Pooling module and the right image showing the outcome with the Strip Pooling module. By comparing the left and right images in Figure 8(b), it is observable that the introduction of the Strip Pooling module significantly enhances the long strip features of the book spines. However, this enhancement effect might also apply to other environmental factors that similarly exhibit strip-like shapes, such as the crossbars of bookshelves, potentially introducing irrelevant features. To suppress these irrelevant features, the self-attention module was integrated into the overall framework, and its effectiveness is demonstrated in Figure 8(a) (where the left image is without the self-attention module, and the right image is with the self-attention module). This further substantiates the importance of the self-attention module in optimizing feature maps and improving segmentation accuracy.

4. **Results.** In the comparison of various network segmentation algorithms, the book spine database is divided into two categories: near-vertical book spine data and tilted book spine data. The training set is then trained using tilted and near-vertical data. In order to examine the impact of book spine tilt on various algorithms, comparative results are presented on two test databases: near vertical and tilted. The comparative test results of different network segmentation algorithms are shown in Table 3. The * in Table 3 represents the test results obtained by re-training on the dataset of this paper using open source code and default parameters provided by the corresponding literature.

Table 3. Experimental results of various semantic segmentation algorithms on the spine test set

Algorithm	Backbone	$I_{\mathrm{MoU}}/\%$		t/s
		Near-vertical	Tilted	
PSPNet (8s)*	MobileNetV2	81.9	79.9	0.120
U-Net*	ResNet50	87.5	83.1	0.223
SegFormer*	Transformers	86.6	86.4	0.249
DeepLabv3 plus (8s)*	MobileNetV2	92.3	89.7	0.233
Ours	MobileNetV2	**94.1**	**93.3**	0.225

Table 3 shows algorithm values in parentheses representing the downsampling multiplicity, 8s. This means that the feature maps, which have been downsampled by 8, are restored to their original input size through inverse convolution. Theoretically, the smaller the number, the more up-sampling operations are performed by the network using the inverse convolution layer. This results in a more complex model structure and theoretically provides more precise segmentation. Therefore, an 8-fold downsampling network is used for testing. As shown in Table 3, the improved DeepLabv3 plus segmentation algorithm achieved the highest segmentation accuracy for both the near-vertical book spine database and the tilted book spine database. Furthermore, it meets real-time processing requirements due to its fast segmentation speed.

Figure 9 illustrates the segmentation performance of different algorithms. Among them, the PSPNet (Pyramid Scene Parsing Network) network architecture is similar to the encoding module of DeepLabv3 plus, using pooling features of different scales to construct feature pyramids for semantic segmentation. Figure 9 shows that PSPNet segmentation

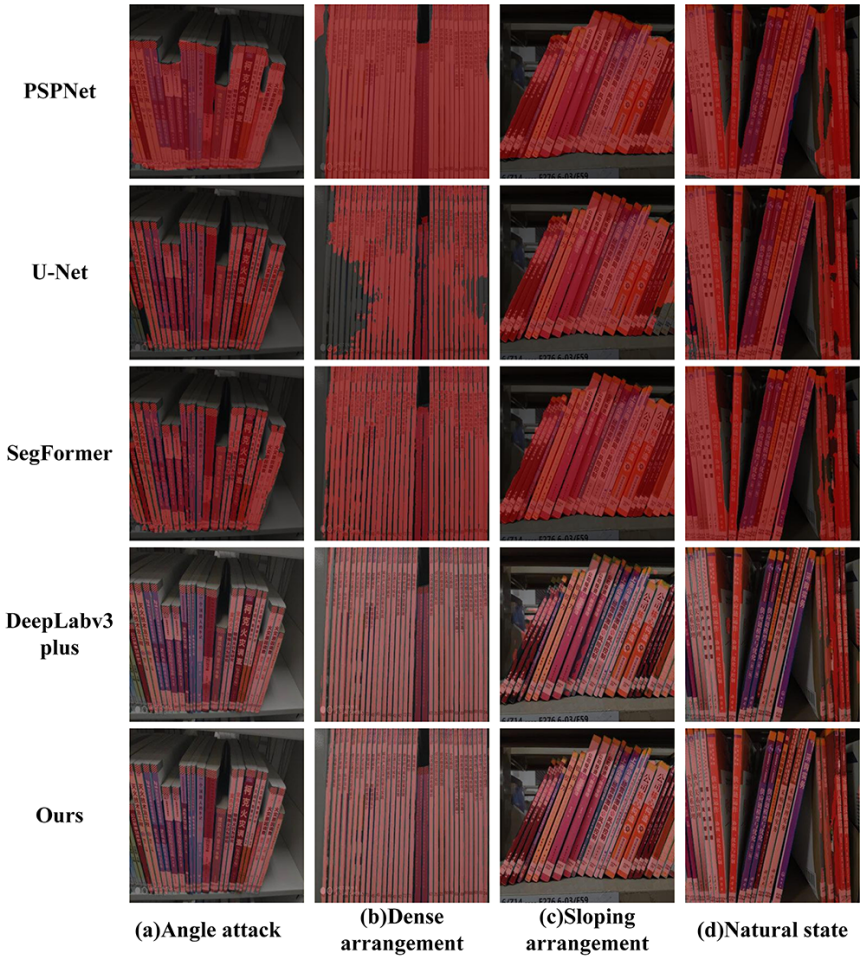
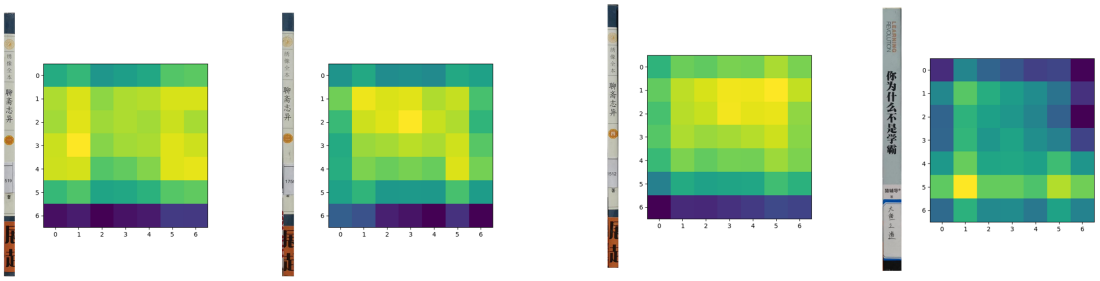


FIGURE 9. The segmentation effect of various algorithms

effect is poor, with large adhesion patches and low segmentation accuracy. The U-Net network uses an encoding and decoding structure, which pays more attention to segmentation details than PSPNet networks. However, its receptive field perception range is smaller, which reduces the algorithm's accuracy when detecting and segmenting objects with large aspect ratio differences. This has a significant impact on the dense distribution of the target book spine, as shown in Figure 9(b). Although its segmentation accuracy has improved over the PSPNet network, its segmentation effect is relatively poor for dense book spines. The SegFormer model uses Transformers as the backbone network, combined with multi-layer perceptual networks to achieve semantic segmentation. From Figure 9, it still performs well in segmenting dense targets, but for tilted targets, the adhesion is more severe. The original DeepLabv3 plus network performs better than the first three segmentation algorithms. However, there is still the problem of unclear segmentation of book spine region boundaries, as shown in Figure 9(c). The above further proves that other segmentation algorithms lack applicability to long strip feature targets. In contrast, the improved DeepLabv3 plus algorithm performs better in book spine segmentation.

Book spine feature visualization results. For book spine images, the VGG16 model based on the Keras framework is used, with Imagenet as the pre-training model. The classification module is removed and global pooling is added for feature extraction. Figure 10 shows the visualization results of the $7 \times 7 \times 512$ dimensional features obtained by VGG16 without global pooling, and then sums them up.



(a) Same series spines (b) Different series spines

FIGURE 10. Visualization results of VGG16 features

Figure 10 shows the visualization results for the same series and different series of book spine features. According to Figure 10, the book spine features of the three versions of 'Liaozhai Zhiyi' are highly repeatable and separate.

This paper proposes a deep feature matching algorithm based on the Faiss framework. The matching database comes from the book spine images obtained by the segmentation algorithm. The segmented book spine images are divided into "same series book spine" databases and "different series book spine" databases. Table 4 shows comparisons of tests with other experiments.

Table 4 shows that the two perform better at matching different series of book spines. However, for the same series of book spines, the latter has better matching performance. [18] utilized image and text features for preprocessing. Specifically, the shape and color features of the book spine are first extracted, and the text information on the book spine is extracted by the Tesseract OCR. Image and text string matching improves matching accuracy. Due to the high repeatability of book spine color features, this algorithm relies on string matching. However, there are many types of Chinese fonts, and the string recognition process is very complex. This algorithm is not suitable for practical applications. Deep feature matching based on the Faiss framework greatly improves matching results due to the use of more detailed features. In Table 4, the proposed matching algorithm stands out in the same series of book spine matching, but it still consumes more time than traditional algorithms. The reason is that this algorithm takes a long time to load the VGG network weights, but it still meets many practical requirements.

TABLE 4. Comparison between the algorithm proposed in this paper and other matching algorithms

Matching algorithm	Different series book spine Acc/%	Same series book spine Acc/%	T/(s/book)
[18]	90.0	–	2.3
Faiss deep feature matching	98.4	95.3	2.33

Based on this analysis, the deep feature matching algorithm based on the Faiss framework is superior to the traditional algorithm proposed in [18]. For different series of book spines, the accuracy has improved by 8.4%, and for the same series of book spines, the accuracy has also reached 95.3%. It has improved book spine matching accuracy while ensuring speed.

5. **Conclusion.** This paper proposes a book spine segmentation network based on an improved DeepLabv3 plus, which effectively enhances feature extraction and segmentation of densely arranged and tilted book spines by incorporating a global self-attention module,

DenseASPP, and stripe pooling. Meanwhile, a Faiss-based deep feature matching algorithm combined with VGG16-extracted features is developed to achieve efficient and highly accurate spine matching. Experimental results demonstrate that the proposed method outperforms existing mainstream algorithms in both segmentation accuracy and matching speed, showing promising application potential.

Despite these significant achievements, certain limitations remain. Specifically, segmentation performance under extreme spine inclinations and severe occlusions requires further improvement, and the computational efficiency and robustness of the matching model can be further optimized. Future work will focus on multimodal fusion techniques and lightweight network design to enhance system adaptability and real-time performance, as well as extend the algorithm's applicability to more complex scenarios, thereby advancing automated book inventory technologies in intelligent library systems.

Acknowledgment. This work was supported by the Liaoning Provincial Key Research and Development Project (No. LJKZZ20220033) and the Science and Technology Innovation Project in the Artificial Intelligence Field of Liaoning Province (Applied Basic Research Plan Project), (No. 2023JH26/10300013).

REFERENCES

[1] N. Tabassum, S. Chowdhury, M. K. Hossen and S. U. Mondal, An approach to recognize book title from multi-cell bookshelf images, *Proc. of the 2017 IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, Dhaka, Bangladesh, pp.1-6, 2017.

[2] M. P. Nevetha and A. Baskar, Automatic book spine extraction and recognition for library inventory management, *Proc. of the 3rd International Symposium on Women in Computing and Informatics*, Kochi, India, pp.44-48, 2015.

[3] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk and B. Girod, Robust text detection in natural images with edge-enhanced maximally stable extremal regions, *Proc. of the 2011 18th IEEE International Conference on Image Processing*, Brussels, Belgium, pp.2609-2612, 2011.

[4] B. Zhu, L. Yang, X. Wu and T. Guo, Automatic recognition of books based on machine learning, *Proc. of the 2015 3rd International Symposium on Computational and Business Intelligence (ISCBI)*, Bali, Indonesia, pp.74-78, 2015.

[5] S. Zhou, T. Sun, X. Xia, N. Zhang, B. Huang, G. Xian and X. Chai, Library on-shelf book segmentation and recognition based on deep visual features, *Information Processing & Management*, vol.59, no.6, 103101, 2022.

[6] W. Zeng, Y. Yang and X. Zhong, A shan-shaped network for semantic segmentation of book spines on bookshelves, *Journal of Image and Signal Processing*, vol.9, pp.218-225, 2020.

[7] W. Zeng, Y. Yang and X. Zhong, An improved mask R-CNN-based method for instance segmentation of book spine images on bookshelves, *Computer Applications and Research*, vol.38, no.11, pp.3456-3459, 2021.

[8] D.-J. Lee, Y. Chang, J. K. Archibald and C. Pitzak, Matching book-spine images for library shelf-reading process automation, *Proc. of the 2008 IEEE International Conference on Automation Science and Engineering*, Washington, D.C., USA, pp.738-743, 2008.

[9] S. G. Fowers and D.-J. Lee, An effective color addition to feature detection and description for book spine image matching, *International Scholarly Research Notices*, vol.2012, no.1, 945973, 2012.

[10] D. M. Chen, S. S. Tsai, B. Girod, C.-H. Hsu, K.-H. Kim and J. P. Singh, Building book inventories using smartphones, *Proc. of the 18th ACM International Conference on Multimedia*, Florence, Italy, pp.651-654, 2010.

[11] Y. Liu, X. Bai, J. Wang, G. Li, J. Li and Z. Lv, Image semantic segmentation approach based on DeepLabV3 plus network with an attention mechanism, *Engineering Applications of Artificial Intelligence*, vol.127, 107260, 2024.

[12] R. Liang and F. Li, Design and research of a multi-view graph deep learning 3D model retrieval system based on fusion vision-transformer, *International Journal of Innovative Computing, Information and Control*, vol.20, no.6, pp.1775-1788, 2024.

[13] R. Sigit, A. Yuliyanto, M. Rochmad and I. S. Azhar, COVID-19 aerosol suction robot to assist dentist surgery based on mouth openness detection using deep learning, *International Journal of Innovative Computing, Information and Control*, vol.19, no.5, pp.1377-1391, 2023.
[14] M. Yang, K. Yu, C. Zhang, Z. Li and K. Yang, DenseASPP for semantic segmentation in street scenes, *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp.3684-3692, 2018.
[15] Q. Hou, L. Zhang, M.-M. Cheng and J. Feng, Strip pooling: Rethinking spatial pooling for scene parsing, *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA (virtual), pp.4003-4012, 2020.
[16] M. D. Rahman, F. Humayara, S. M. E. Rabbi and M. M. Rashid, Efficient medical image retrieval using DenseNet and FAISS for BIRADS classification, *arXiv Preprint*, arXiv: 2411.01473, 2024.
[17] Y. Liu, Q. Wang, H. Zhang, Y. Liu and K. Zhao, Real-time defect detection of metal surface based on improved YOLOv4, *International Journal of Innovative Computing, Information and Control*, vol.18, no.4, pp.1329-1338, 2022.
[18] L. Cao, M. Liu, Z. Dong and H. Yang, Book spine recognition based on OpenCV and Tesseract, *Proc. of the 2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Hangzhou, China, vol.1, pp.332-336, 2019.

Author Biography

Lifu Hu received the M.S. degree from the Northeastern University, China, in 2008. From 2001 to 2011, he was an Experimenter with the College of Automation, Shenyang Aerospace University, China. Since 2011, he has been a Senior Experimenter with the College of Automation, Shenyang Aerospace University. His research interests include computer detection technology and intelligent system, image collection and processing, and intelligent robot design.

Sixu Zhao received his B.S. degree from Shenyang Aerospace University, China, in 2023. He is pursuing a master's degree at Shenyang Aerospace University. His research directions include video analysis and processing, pattern recognition, and image analysis.

Lirong Tang received his B.S. degree from Shenyang Aerospace University, China, in 2022. He is pursuing a master's degree at Shenyang Aerospace University. His research directions include video analysis and processing, pattern recognition, and image analysis.

Xiaofei Ji received her B.S. and M.S. degrees from Liaoning Petrochemical University, China, in 2000 and 2003, and her Ph.D. degree from the Portsmouth University, Britain, in 2010. Since 2013, she has been an associate professor at Shenyang Aerospace University, China. Her research interests include video analysis and pattern recognition theory, etc.

Kexin Zhang received her M.S. degree from Shenyang Aerospace University, China, in 2023. She is currently employed at the Beijing Mechanical Equipment Research Institute. Her research interests include video analysis and processing, pattern recognition, and image analysis.