

**TITLE OF INVENTION: SAMPLE CONSTRUCTION METHOD AND DEVICE,
MODEL TRAINING METHOD AND DEVICE**

FIELD

5 **[0001]** The present disclosure relates to the technical field of artificial intelligence, and in particular to a sample construction method and device, and a model training method and device

BACKGROUND

10 **[0002]** With the rapid development of multimodal document recognition technology, its applications in layout analysis, table parsing, key information extraction, and document question answering have become increasingly widespread. The quality and diversity of training data directly determine the upper performance limit of a model. Currently, training datasets are constructed mainly relying on the collection, cleaning, and manual annotation of
15 real documents. In the conventional art, various paper or electronic documents are typically collected and converted into image formats, then text regions, table structures, paragraph categories, etc. in the images are framed and classified by annotators using annotation tools, to generate corresponding label files. In some automated solutions, it is attempted to use rule templates to fill text content to synthesize simple document images, or perform format
20 conversion and basic parsing based on existing document files to obtain preliminary image-text correspondences, to assist the model training. However, when faced with globally diverse writing modes, complex multi-page continuous document structures, and fine-grained recognition requirements at the character level or cell level, it is often difficult to generate effective training data that comprehensively covers the diversity and maintains highly
25 consistent labels. Therefore, an effective solution is urgently needed to solve the above problems.

SUMMARY

30 **[0003]** In view of this, a sample construction method is provided according to embodiments of the present disclosure. One or more embodiments of the present disclosure further relate to

a sample construction device, a model training method, a model training device, a computing device, a computer-readable storage medium, and a computer program product, so as to solve the technical deficiencies in the conventional art.

[0004] According to a first aspect of the embodiments of the present disclosure, a sample construction method is provided, which includes:

acquiring a document file and a text corpus, determining multiple document images and a document base corpus according to the document file, and determining a text base corpus according to the text corpus;

extracting element style information and page layout information corresponding to each of the document images, constructing a document template according to the element style information and the page layout information, and determining a target base corpus according to the document base corpus and the text base corpus;

fusing the document template and the target base corpus by an element renderer and a layout manager according to a preset writing mode and a font resource library, to obtain a document recognition sample and a document recognition label; and

constructing a document recognition sample pair based on the document recognition sample and the document recognition label, and storing the document recognition sample pair into a sample library.

[0005] According to a second aspect of the embodiments of the present disclosure, a model training method is provided, which includes:

acquiring a document file and a text corpus, determining multiple document images and a document base corpus according to the document file, and determining a text base corpus according to the text corpus;

extracting element style information and page layout information corresponding to each of the document images, constructing a document template according to the element style information and the page layout information, and determining a target base corpus according to the document base corpus and the text base corpus;

fusing the document template and the target base corpus by an element renderer and a layout manager according to a preset writing mode and a font resource library, to obtain a document recognition sample and a document recognition label; and

training a multimodal document recognition model based on the document recognition sample and the document recognition label, until a target multimodal document recognition model meeting a training stop condition is obtained.

[0006] According to a third aspect of the embodiments of the present disclosure, a sample construction device is provided, which includes:

an acquisition module, configured to acquire a document file and a text corpus, determine multiple document images and a document base corpus according to the document file, and determine a text base corpus according to the text corpus;

an extraction module, configured to extract element style information and page layout information corresponding to each of the document images, construct a document template according to the element style information and the page layout information, and determine a target base corpus according to the document base corpus and the text base corpus;

a fusion module, configured to fuse the document template and the target base corpus by an element renderer and a layout manager according to a preset writing mode and a font resource library, to obtain a document recognition sample and a document recognition label; and

a storage module, configured to construct a document recognition sample pair based on the document recognition sample and the document recognition label, and store the document recognition sample pair into a sample library.

[0007] According to a fourth aspect of the embodiments of the present disclosure, a model training device is provided, which includes:

a corpus acquisition module, configured to acquire a document file and a text corpus, determine multiple document images and a document base corpus according to the document file, and determine a text base corpus according to the text corpus;

an information extraction module, configured to extract element style information and page layout information corresponding to each of the document images, construct a document template according to the element style information and the page layout information, and determine a target base corpus according to the document base corpus and the text base corpus;

a corpus fusion module, configured to fuse the document template and the target base

corpus by an element renderer and a layout manager according to a preset writing mode and a font resource library, to obtain a document recognition sample and a document recognition label; and

a model training module, configured to train a multimodal document recognition model based on the document recognition sample and the document recognition label, until a target multimodal document recognition model meeting a training stop condition is obtained.

[0008] According to a fifth aspect of the embodiments of the present disclosure, a computing device is provided, which includes a memory and a processor. The memory is configured to store computer executable instructions, the processor is configured to execute the computer executable instructions, where the computer executable instructions, when executed by the processor, perform the steps of the sample construction method or the model training method described above.

[0009] According to a sixth aspect of the embodiments of the present disclosure, a computer-readable storage medium is provided, which includes computer executable instructions stored thereon, where the computer executable instructions, when executed by a processor, perform the steps of the sample construction method or the model training method described above.

[0010] According to a seventh aspect of the embodiments of the present disclosure, a computer program product is provided, which includes computer programs or instructions, where the computer programs or instructions, when executed by a processor, perform the steps of the sample construction method or the model training method described above.

[0011] With the sample construction method according to the embodiments, a document file and a text corpus are acquired through a dual-source input mechanism, and are separately parsed into high-quality document images, document base corpus, and text base corpus. Then, element style information and page layout information are precisely extracted from the document images to construct reusable document templates. On this basis, the document template is deeply fused with the screened and enhanced target base corpus by an element renderer working collaboratively with a layout manager in combination with a preset writing mode and a font resource library, which not only realizes automatic generation of document recognition samples with multiple elements, multiple layouts and multiple writing modes, but also synchronously derives accurate document recognition labels during the rendering process,

and finally constructs and stores document recognition sample pairs containing samples and labels. In this way, high accuracy of the generated samples in visual presentation, structural logic, and label alignment can be effectively ensured, and the authenticity and diversity of training data can be significantly improved, which can provide high-quality data support with strong supervision signals for the multimodal document recognition model, thereby accelerating model training convergence and improving generalization performance, thus realizing a closed-loop iteration from data construction to model optimization.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] Figure 1 is a flowchart of a sample construction method according to an embodiment of the present disclosure;

[0013] Figure 2a is a schematic diagram of sample construction in a sample construction method according to an embodiment of the present disclosure;

[0014] Figure 2b is a schematic diagram of page layout extraction in a sample construction method according to an embodiment of the present disclosure;

[0015] Figure 2c is a schematic diagram of template construction in a sample construction method according to an embodiment of the present disclosure;

[0016] Figure 2d is a schematic diagram of corpus management in a sample construction method according to an embodiment of the present disclosure;

[0017] Figure 2e is a schematic diagram of corpus augmentation in a sample construction method according to an embodiment of the present disclosure;

[0018] Figure 2f is a schematic diagram of sample construction in a sample construction method according to another embodiment of the present disclosure;

[0019] Figure 2g is a schematic diagram of data augmentation in a sample construction method according to an embodiment of the present disclosure;

[0020] Figure 3 is a flowchart of a model training method according to an embodiment of the present disclosure;

[0021] Figure 4 is a schematic structural diagram of a sample construction device according to an embodiment of the present disclosure;

[0022] Figure 5 is a schematic structural diagram of a model training device according to an embodiment of the present disclosure; and

[0023] Figure 6 is a structural block diagram of a computing device according to an embodiment of the present disclosure.

5

DETAILED DESCRIPTION OF EMBODIMENTS

[0024] Many specific details are set forth in the following description to facilitate a full understanding of the present disclosure. However, the present disclosure may be implemented in many other ways different from those described herein, and those skilled in the art may
10 make similar generalizations without violating the connotations of the present disclosure, and thus the present disclosure is not limited by the specific embodiments described below.

[0025] Terms used in one or more embodiments of the present disclosure are used only for the purpose of describing a particular embodiment and are not intended to limit one or more embodiments of the present disclosure. The singular forms of "a", "the", and "the" used in one
15 or more embodiments of the present disclosure and the claims are also intended to encompass the plural form, unless the context clearly indicates otherwise. It should also be understood that the term "and/or" used in one or more embodiments of the present disclosure refers to and encompasses any or all possible combinations of one or more of the listed items in association.

[0026] It should be understood that while the terms "first", "second", etc. may be used in one or more embodiments of the present disclosure to describe various information, such information should not be limited to these terms. These terms are used only to distinguish the same type of information from one another. For example, without departing from the scope of the one or more embodiments of the present disclosure, "a first..." may also be referred to as
20 "a second...", and similarly, "a second..." may be referred to as "a first...". Depending on the context, the phrase "if" as used herein may be interpreted as "at the time of" or "when" or "in response to a determination".

[0027] In addition, it should be noted that the user information (including but not limited to user device information, user personal information, etc.) and data (including but not limited to data used for analysis, stored data, displayed data, etc.) involved in one or more embodiments
30 of the present disclosure are authorized by the user or sufficiently authorized by the parties,

and that the collection, use, and processing of the relevant data are subject to relevant laws, regulations, and regulations of relevant countries and regions, and a corresponding operation portal is provided for the user to choose to authorize or reject.

5 [0028] First, the terminologies involved in one or more embodiments of the present disclosure are explained below.

[0029] Multimodal document recognition model refers to a model that jointly processes multiple types of information such as document images, text, and layout structures, and is used to perform tasks such as text recognition, layout analysis, table parsing, key information extraction, document question answering, and document classification.

10 [0030] Effective data refers to data that not only has high annotation accuracy, but also can substantially improve the capability of the target model.

[0031] High-quality data refers to data with accurate labels, consistent structure, standardized format, and ready for direct training.

15 [0032] Element style refers to visual attributes of various elements in a document, including font, font size, font weight, color, italicization, alignment, paragraph spacing, line spacing, borders, shading, transparency, etc.

[0033] Page layout refers to the region division, relative positional relationships, reading order, and hierarchical structure of various elements on a page.

20 [0034] Template refers to a reusable document structure definition, including at least one or more of page layout constraints, element type constraints, and element style constraints.

[0035] Active learning refers to a class of methods that prioritize selecting the most valuable samples for training based on model uncertainty, error rate, boundary, or information gain on samples.

25 [0036] Hard samples refer to document samples, layout samples, style samples, or element combination samples that the existing model performs poorly on, tends to misclassify, or cannot stably recognize.

[0037] Synchronous label generation refers to the automatic generation of corresponding annotation information, such as bounding boxes, categories, reading order, structure trees, text content, key-value correspondences, etc., at the same time as generating document images,

document files, or page content.

[0038] Multi-page continuous document refers to a document in which multiple pages maintain continuity in subject, content order, chapter structure, chart references, page numbers, and reading logic.

5 **[0039]** Writing mode refers to the organization and expansion mode of text on a page, including at least one or more of horizontal left-to-right, horizontal right-to-left, vertical top-to-bottom with columns right-to-left, and vertical top-to-bottom with columns left-to-right.

10 **[0040]** Character-level controllable generation refers to the ability to finely control and record attributes such as position, size, font, color, rotation, character spacing, line, and paragraph of individual characters during document generation.

[0041] Cell-level controllable generation refers to the ability to finely control and record attributes such as position, span, border, shading, text content, intra-cell character attributes, and relationships with adjacent cells of each cell during table generation.

15 **[0042]** In the present disclosure, a sample construction method is provided. One or more embodiments of the present disclosure further relate to a sample construction device, a model training method, a model training device, a computing device, a computer-readable storage medium, and a computer program product, which are described in detail one by one in the following embodiments.

20 **[0043]** In practice, with the rapid development of multimodal document recognition models, training data has become a core factor restricting the improvement of model performance. In the conventional art, construction of training data mainly relies on manual collection, screening, and annotation. However, real documents have complex sources, non-uniform formats, and involve copyright sensitivity issues, resulting in long data acquisition cycles and
25 high costs. In addition, document recognition tasks involve multiple types of elements such as text, titles, tables, formulas, and complex reading orders. Manual annotation is not only time-consuming and labor-intensive, but also difficult to ensure consistency and accuracy of label specifications. In addition, existing synthetic data solutions are often limited to simple layouts or single writing modes, and it is difficult to generate hard samples covering global
30 multi-language, multi-writing-direction, and complex layout structures. As a result, the

generated data has limited improvement on model capabilities and cannot form a data closed loop oriented to model weaknesses.

[0044] In view of this, with the sample construction method according to the embodiment, a document file and a text corpus are acquired through a dual-source input mechanism, and are
5 separately parsed into high-quality document images, document base corpus, and text base corpus, then element style information and page layout information are precisely extracted from the document images to construct reusable document templates. On this basis, the document template is deeply fused with the screened and enhanced target base corpus by an element renderer working collaboratively with a layout manager in combination with a preset
10 writing mode and a font resource library, which not only realizes automatic generation of document recognition samples with multiple elements, multiple layouts and multiple writing modes, but also synchronously derives accurate document recognition labels during the rendering process, and finally constructs and stores document recognition sample pairs containing samples and labels. In this way, high accuracy of the generated samples in visual
15 presentation, structural logic, and label alignment can be effectively ensured, and the authenticity and diversity of training data can be significantly improved, which can provide high-quality data support with strong supervision signals for the multimodal document recognition model, thereby accelerating model training convergence and improving generalization performance, thus realizing a closed-loop iteration from data construction to
20 model optimization.

[0045] Referring to Figure 1, Figure 1 shows a flowchart of a sample construction method according to an embodiment of the present disclosure. Specifically, the method includes the following steps S102 to S108.

[0046] In step S102, a document file and a text corpus are acquired, multiple document
25 images and a document base corpus are determined according to the document file, and a text base corpus is determined according to the text corpus.

[0047] Specifically, the acquiring a document file and a text corpus may refer to a process of receiving raw input data from multiple data sources. The document files includes but is not limited to electronic device files in formats such as DOC, XLS, PPT, PDF, HTML, and
30 document image files in formats such as JPG, PNG, TIF. The text corpora include plain text or structured text data such as TXT, JSONL, MD, LaTeX, XML. These data may come from

business historical sample libraries, public datasets, manually uploaded libraries, or web-crawled data.

[0048] Based on this, the determining multiple document images and a document base corpus according to the document file specifically includes: first, classifying the document files according to the formats, converting a document file in a non-image-format into a unified preset document format (e.g., PDF), and then converting the document file in the preset document format into multiple initial document images, to ensure that each initial document image corresponds to one page in the document file; then performing image detection on the initial document images, filtering out images that are blurred, excessively rotated, or do not meet a sample construction condition, to obtain multiple high-quality document images; finally, parsing the content of the document images by a multimodal parsing model, extracting paragraph texts and corresponding categories (such as titles, body text, tables, figure captions, etc.) from the documents, where the binary tuple information consisting of paragraph texts and paragraph categories constitutes the document base corpus.

[0049] Correspondingly, the determining a text base corpus according to the text corpus specifically includes: splitting the text corpus into paragraphs to obtain multiple candidate text paragraphs, classifying the candidate text paragraphs, determining text paragraph texts and corresponding text paragraph categories, to form the text base corpus.

[0050] For example, if the inputted document file is a PDF file of an academic paper containing both Chinese and English, the system converts the document file into multiple PNG images, and extracts paragraph texts such as abstract and introduction along with their categories as the document base corpus through a parsing model. In addition, the inputted text corpus is a TXT file from a legal knowledge base, and the system splits text corpus into independent legal article paragraphs and marks legal article paragraphs as legal provision categories, to form the text base corpus. Through the dual-source input and separate parsing approach, the structural assets of real documents and abundant textual content resources can be fully utilized, providing a rich and diverse material foundation for subsequent sample generation.

[0051] In step S104, element style information and page layout information corresponding to each document image are extracted, a document template is constructed according to the element style information and the page layout information, and a target base corpus is

determined according to the document base corpus and the text base corpus.

[0052] Specifically, element style information may refer to a collection of visual attributes of various elements in a document, including font name, font size, font weight, italicization, font color, background color, alignment, line spacing, paragraph spacing, indentation, border style, shadow, transparency, and writing mode of paragraphs. For image elements, the element style information further includes aspect ratio; and for table elements, the element style information further includes line styles and cell styles, etc. The page layout information may refer to region division, relative positional relationships, reading order, and hierarchical structure of various elements on a page, including headers, footers, body regions, sidebars, single-column/double-column/multi-column layouts, grid layouts, floating regions, and image-text association relationships, etc.

[0053] The extraction process specifically includes: identifying and recording sub-style information corresponding to different element style types from the document image according to a preset element style extraction rule; simultaneously determining the document type corresponding to the document image (such as certificates, books, papers, etc.), deriving inferred layout information according to the document type and performing verification, and extracting page hierarchy structure information and region relationship information if the verification passes.

[0054] Correspondingly, the constructing a document template according to the element style information and the page layout information specifically includes: storing the extracted style information into an element style library, and storing the layout information into a page layout library; selecting style and layout information to be combined from the library according to a preset template construction rule, constructing a combined document template, or performing enhancement processing on a reference document template (such as adjusting font size, changing margins, etc.) to obtain a new document template.

[0055] Correspondingly, the determining a target base corpus according to the document base corpus and the text base corpus specifically includes: merging the two types of base corpora to construct a candidate base corpus, performing deduplication on the candidate base corpus, and dividing the deduplicated candidate base corpus into specific document corpus and non-specific document corpus; performing similar corpus expansion or identical corpus enhancement processing on the specific document corpus (for example, using a large

language model to rewrite sentence patterns or generate synonymous expressions), and finally using the processed corpus as the target base corpus.

[0056] For example, a title style of bold, 14-point, black font and a layout with fixed upper-left and floating lower-right positions are extracted from a real invoice image, to construct an invoice template; simultaneously, product description texts from a knowledge base are expanded to generate product names in multiple expressions as the target base corpus. By automatically extracting and depositing style and layout assets from real documents, combined with targeted corpus expansion, it can be ensured that the generated samples are highly similar to real scenes in visual style and logical structure, while avoiding model overfitting caused by duplicate corpora.

[0057] In specific implementation, in a case that the subsequently constructed sample pairs are to be used for training a multimodal document recognition model, the parsing of documents may be performed by a multimodal document recognition model trained in a previous stage to extract page layout information and element style information, and then high-quality sample pairs may be generated in combination with the base corpus, thereby optimizing the model and progressively enhancing the training effect.

[0058] In step S106, the document template and the target base corpus are fused by an element renderer and a layout manager according to a preset writing mode and a font resource library, to obtain a document recognition sample and a document recognition label.

[0059] Specifically, the element renderer is an engine component for accurately rendering document elements, internally integrating a rollback algorithm, a truncation algorithm, a pre-rendering algorithm, a writing mode control algorithm, and a first information extraction algorithm. The rollback algorithm is used to roll back the state of a local or global paragraph element in case of page planning conflicts; the truncation algorithm is used to decide and manage the truncation state of elements to be rendered when region space is insufficient; the pre-rendering algorithm is used to obtain geometric information such as width and height and character-level position information of elements before formal layout; the writing mode control algorithm is used to determine the sub-writing mode corresponding to the element to be rendered (e.g., horizontal left-to-right, horizontal right-to-left, vertical top-to-bottom with columns right-to-left, vertical top-to-bottom with columns left-to-right, etc.); the first information extraction algorithm is used to extract detailed attributes of the element to be

rendered according to a preset granularity.

[0060] Correspondingly, the layout manager is an engine component for managing the overall page layout logic, internally integrating a writing mode algorithm, a reading layout algorithm, a region occupation algorithm, a conflict rollback algorithm, a cutting algorithm, and a second information extraction algorithm. The writing mode algorithm is used to manage the writing direction and advancement logic of a whole page or columns; the reading layout algorithm is used to strictly maintain the relative reading order among text elements, images, and tables; the region occupation algorithm is used to dynamically manage the occupied state of regions, determining whether an element should be moved to the next column or page; the conflict rollback algorithm is used to undo a previous state to reallocate space in case of rendering conflicts; the cutting algorithm is used to handle cutting logic of elements crossing columns or pages; the second information extraction algorithm is used to extract a final rendering result including page and region information. The preset writing mode supports unified modeling of multiple global writing habits, and the font resource library contains font files covering multiple global languages and ancient/modern text forms.

[0061] The fusion process specifically includes: rendering the target base corpus by the element renderer according to the preset writing mode and the font resource library, in combination with the template style information in the document template, to generate paragraph elements with precise coordinates and styles; then rendering these paragraph elements into specified regions of the document template by the layout manager according to the template layout information, handling complex situations such as cross-page, cross-column, and element avoidance, to obtain a document recognition sample. During this process, the system synchronously generates corresponding document recognition labels based on the template style information and template layout information. The label content covers information such as element categories, bounding boxes, line-level/word-level/character-level boxes, reading order, table structures, cell spans, writing mode.

[0062] For example, when generating a document with mixed Arabic (horizontal right-to-left) and Chinese (horizontal left-to-right), the element renderer invokes corresponding fonts and arrangement rules to render characters according to the writing mode control algorithm, while the layout manager ensures correct adjacency relationships between

the two language paragraphs according to the reading layout algorithm, and outputs fine-grained labels containing character coordinates and language attributes immediately upon completion of rendering. Through the collaborative work of the element renderer and the layout manager, full-link controllable generation from macroscopic layout to microscopic character level is achieved, thereby ensuring high consistency between samples and labels, thus significantly reducing manual annotation costs.

[0063] In step S108, a document recognition sample pair is constructed based on the document recognition sample and the document recognition label, and the document recognition sample pair is stored into a sample library.

[0064] Specifically, a document recognition sample pair is an associated data unit composed of a document recognition sample (image or file) and its corresponding document recognition label (structured annotation data). The construction process specifically includes: keeping element positions in the document recognition sample unchanged, and adding position-invariant enhancement information (such as blur, noise, brightness adjustment, contrast change, local shadows) to obtain a first enhanced document recognition sample; and/or, adjusting element positions in the document recognition sample, adding position-variant enhancement information (such as rotation, affine transformation, warping, projective transformation), and performing synchronous geometric transformation on the labels while transforming the image, to obtain a second enhanced document recognition sample; and constructing a first document recognition sample pair based on the first enhanced document recognition sample and its label, and constructing a second document recognition sample pair based on the second enhanced document recognition sample and its label, where the first document recognition sample pair and the second document recognition sample pair are collectively referred to as document recognition sample pairs. The storing the document recognition sample pairs into a sample library may refer to writing the constructed sample pairs into a persistent storage medium, forming a versionable and traceable dataset for subsequent training, evaluation, and active learning strategies of the multimodal document recognition model.

[0065] For example, after generating a clear contract sample, the system automatically generates a version with added Gaussian noise (position-invariant) and a version slightly rotated by 5 degrees (position-variant), and respectively match labels that have undergone

corresponding transformations, forming two enhanced sample pairs stored in the sample library. By packaging samples with labels and performing diverse enhancements, the distribution of training data is enriched, and robustness of the model to real imaging disturbances and layout deformations is improved, laying a solid foundation for constructing high-quality effective datasets.

[0066] Referring to the schematic diagram shown in Figure 2a, with the sample construction method according to the embodiment, after acquiring raw data through an input acquisition module, format classification may be performed to determine documents and texts. For documents, after PDF conversion and image conversion, image cleaning is performed, and then high-quality images are subjected to document analysis to determine the base corpus and document images. For texts, paragraph splitting and paragraph classification are performed to determine the corresponding base corpus. Thereafter, element style information and page layout information are extracted from the document images, and enhanced style template APIs are generated through template management for subsequent template construction. The base corpus undergoes corpus management to generate text corpora. After fusion rendering and template processing through a synthesis engine, samples are obtained. During this process, to ensure sufficient corpus, corpus expansion may also be performed to obtain more corpora and generate more types of samples. Data augmentation may also be performed on the generated samples to increase sample quantity and comprehensiveness. Based on this, model training and inference are performed. After model training, AL strategy evaluation may be used to determine the distribution of samples on model training effects, and subsequently, constraints may be constructed based on this to readjust template generation, ensuring the generation of high-quality samples for training models and improving model training effects.

[0067] In summary, a document file and a text corpus are acquired through the dual-source input mechanism and are separately parsed into high-quality document images, document base corpus, and text base corpus; then element style information and page layout information are precisely extracted from the document images to construct reusable document templates. On this basis, the document template is deeply fused with the screened and enhanced target base corpus by an element renderer working collaboratively with a layout manager in combination with a preset writing mode and a font resource library, which not only realizes automatic generation of document recognition samples with multiple elements, multiple

layouts and multiple writing modes, but also synchronously derives accurate document recognition labels during the rendering process, and finally constructs and stores document recognition sample pairs containing samples and labels. In this way, high accuracy of the generated samples in visual presentation, structural logic, and label alignment can be effectively ensured, and the authenticity and diversity of training data can be significantly improved, which can provide high-quality data support with strong supervision signals for multimodal document recognition models, thereby accelerating model training convergence and improving generalization performance, thus realizing a closed-loop iteration from data construction to model optimization.

[0068] In one or more implementations of this embodiment, the acquiring a document file and a text corpus, and determining multiple document images and a document base corpus according to the document file, includes the following steps 21 to 24.

[0069] In step 21, an original dataset is acquired, original data in the original dataset is classified, and a document file and a text corpus are determined according to a classification result.

[0070] The original dataset may refer to an initial data collection used for constructing training samples, with sources including but not limited to business historical sample libraries, public datasets, manually uploaded data, or web-crawled data. The original data is typically heterogeneous, containing multiple file formats. The classifying the original data may refer to dividing the original data into document file classes and text corpus classes according to file extensions, file header information, or metadata characteristics. Document files refer to files carrying layout structure information, such as DOC, DOCX, PPT, XLS, PDF, HTML; text corpora refer to corpus files containing only plain text content or simple markup languages, such as TXT, JSONL, MD, LaTeX, XML.

[0071] Through classification processing, document data with visual typographic attributes and text data with pure semantic content may be separated, providing accurate input sources for subsequent image processing and paragraph structuring processing respectively. For example, if an original dataset contains both .pdf files and .txt files, the system recognizes the suffix .pdf and classifies such file as a document file, and recognizes the suffix .txt and classifies such file as a text corpus, thereby ensuring that in the subsequent process, document files are converted into images to extract layout and style, while text corpora are directly split

to supplement semantic content. This step aims to solve the problem of conflicting processing logic caused by mixed original data formats, ensuring correct separation of dual-source inputs.

[0072] In step 22, the document file is converted into a preset document format, and the document file in the preset document format is converted into multiple initial document images, where each initial document image corresponds to each document page in the document file.

[0073] The preset document format may refer to a unified intermediate exchange format, preferably Portable Document Format (PDF). The converting the document file into a preset document format may refer to uniformly render or convert non-image document files (such as DOCX, PPT, etc.) into PDF files by a format conversion engine, to eliminate differences in layout engines among different source file formats and achieve format normalization. The converting the document file in the preset document format into multiple initial document images may refer to rendering each page of the PDF file into independent bitmap images (such as JPG, PNG format) at a preset resolution (e.g., 300 DPI) and color mode (e.g., RGB). Each initial document image has a strict one-to-one correspondence with a physical page in the document file.

[0074] For example, if the inputted document file is a Word document containing 10 pages, the system first converts the document file into a PDF file, and then renders the PDF file page by page to generate 10 initial document images, each completely reproducing the visual content of a single page of the original document. In this step, through format normalization and page-granularity slicing, standard consistency of subsequent processing objects in the image dimension is ensured, laying the foundation for extracting accurate page layout information.

[0075] In step 23, image detection is performed on the multiple initial document images, and initial document images that meet a sample construction condition is selected according to a detection result as the multiple document images.

[0076] Image detection may refer to a process of evaluating the quality attribute of the initial document image using an image processing algorithm or a pre-trained visual model. The sample construction condition may refer to a preset image quality threshold standard for screening high-quality effective samples. Specific dimensions of image detection include

sharpness detection (e.g., based on variance calculation using Laplacian operator), rotation angle detection (determining whether the image has abnormal angle tilt), distortion detection (determining whether there is perspective deformation or scanning creases), and noise detection. The selecting according to the detection result may refer to eliminating low-quality images that do not meet the sample construction condition, and retaining only the qualified images as the subsequent plurality of document images.

[0077] For example, in a case of a sharpness threshold of 50 and a rotation angle tolerance of ± 2 degrees, if the sharpness score of an initial document image is 30 or the rotation angle is 15 degrees, it is determined that the initial document image does not meet the sample construction condition and is filtered out; conversely, if the image is clear and upright, it is selected for the next stage. This step effectively prevents low-quality images that are blurred, tilted, or severely distorted from entering the subsequent parsing process, preventing distortion of extracted style information or layout errors caused by input noise, significantly improving the robustness of the document base corpus.

[0078] In step 24, the multiple document images are parsed, document paragraph texts and document paragraph categories corresponding to the document paragraph texts are determined according to a parsing result, and the document base corpus is formed from the document paragraph texts and the document paragraph categories.

[0079] The parsing the multiple document images may refer to perform content recognition and structure restoration on the selected document images by a multimodal document understanding model or OCR combined with a layout analysis algorithm. The document paragraph text refers to a text content block with semantic continuity recognized from the images; the document paragraph category refers to a functional type of the text content block in the document structure, including but not limited to titles, subtitles, body paragraphs, list items, figure captions, table titles, table contents, headers, footers, formulas, or code blocks. The forming the document base corpus may refer to packaging the recognized text content with its corresponding category label as a binary tuple structure (Text, Category), to form a structured base corpus collection.

[0080] For example, when parsing a document image of a paper page, the system recognizes the centered large-font content at the top as "Abstract", with the category determined as title; recognizes the multi-column small-font content in the middle as specific

discussion, with the category determined as body text; recognizes the text below the chart at the bottom as "Figure 1 System Architecture Diagram", with the category determined as figure caption. The final document base corpus generated is the set of {(Abstract, title), (This paper proposes..., body text), (Figure 1 System Architecture Diagram, figure caption)}. This step realizes the transformation from unstructured images to structured semantic data, so that the extracted corpus not only contains textual information but also carries clear layout functional attributes, facilitating precise mapping and alignment with element styles in the template in subsequent steps.

[0081] In summary, the above processing not only improves the accuracy and consistency of input data, but also provides a reliable data foundation for subsequently extracting real element styles and page layouts, ensuring that the generated document templates can truly reflect document features in actual business scenarios.

[0082] In one or more implementations of this embodiment, the determining a text base corpus according to the text corpus includes the following steps 31 to 33.

[0083] In step 31, the text corpus is split into paragraphs to obtain multiple candidate text paragraphs.

[0084] The text corpus may refer to a collection of raw text data in non-image form, derived from real business documents, public knowledge bases, large language model generated content, multilingual text resources, code fragment libraries, mathematical formula libraries, etc. Paragraph splitting may refer to cutting a continuous long text stream into text units with independent semantics or structural features according to preset granularity control rules.

[0085] Specifically, paragraph splitting can be achieved by identifying explicit delimiters in the text, for example, using consecutive blank lines, specific title marker symbols (e.g. #), HTML tags (e.g. <p>, <div>, <h1>), LaTeX environment identifiers, or fixed indentation amounts as cutting boundaries. Through this coarse-grained splitting operation, unstructured raw text may be transformed into discrete candidate text paragraphs, providing standardized processing objects for subsequent fine-grained classification. For example, if the inputted text corpus is a source file of an academic paper containing abstract, main text, references, and code blocks, the system first splits the text corpus into tens of candidate text paragraphs based on blank lines and chapter title markers, each paragraph possibly corresponding to a complete natural paragraph, a formula block, or a piece of program code. Through this splitting

approach, the complexity of subsequent processing can be effectively reduced, and the relative semantic integrity of each processing unit can be ensured.

[0086] In step 32, the multiple candidate text paragraphs are classified, and text paragraph texts and text paragraph categories corresponding to the text paragraph texts are determined according to a classification result.

[0087] Classification may refer to performing semantic analysis and structure recognition on the candidate text paragraphs obtained above by a natural language processing model, a rule matching engine, or a multimodal classifier, to determine their functional types. Text paragraph categories include at least one or more of titles, subtitles, body paragraphs, quoted paragraphs, figure captions, table captions, mathematical formulas, code blocks, list items, headers, and footers. The classification process aims to solve the problem that categories are implicit in pure text corpora and are difficult to directly use for template filling. Specifically, the classifier analyzes textual features of the paragraph (such as keyword distribution, special symbols, length), format features (such as whether it contains LaTeX syntax, indentation level), and contextual semantics, thereby outputting accurate category labels. The text paragraph text is the cleaned and standardized paragraph content itself.

[0088] For example, for a candidate paragraph containing *def calculate_area(r): return 3.14 * r * r*, the classification module recognizes that the candidate paragraph contains programming language keywords and function definition structure, and classifies the candidate paragraph as a code block category; for a paragraph containing $\sum_{i=1}^n x_i$, it recognizes the paragraph as a mathematical formula category. Through this fine-grained classification mechanism, the coverage of corpus sources can be greatly expanded, especially supplementing special format contents such as formulas and codes that are difficult to accurately obtain from document image parsing, ensuring the richness and semantic precision of corpus categories, and achieving precise alignment between corpora and subsequent template element categories.

[0089] In step 33, the text base corpus is formed based on the text paragraph texts and the text paragraph categories.

[0090] The text base corpus may refer to a structured data collection composed of text content - category label binary tuples, with a data structure isomorphic to the document base corpus obtained from document image parsing. This step is the process of encapsulating and assembling the processing results of the preceding steps, aiming to form standard corpus units

that can be directly referenced by document templates.

[0091] Specifically, the system binds the text paragraph texts determined above with their corresponding text paragraph categories to form standard data objects such as {content: text content, category: category label}. These objects are then stored in a corpus library as input
5 sources for subsequent construction of the target base corpus. Through this structured assembly, unstructured raw text corpora are transformed into assets with clear semantic attributes, so that the layout manager can, according to the element types defined in the template (e.g., a code block needs to be filled here), directly retrieve and call matching corpus units from the text base corpus in the subsequent document generation process. This
10 significantly improves the language diversity and professionalism of the generated samples, solving the technical problem in traditional solutions where the generated content is monotonous or does not conform to real scene distributions due to the lack of category-specific corpora.

[0092] In summary, a complete processing chain of coarse-grained splitting →
15 fine-grained classification → structured assembly is constructed. By splitting the text corpus into paragraphs, effective control of corpus granularity is achieved, laying the foundation for refined processing; by classifying candidate paragraphs and determining text paragraph texts and categories, precise alignment between corpus content and template element semantics is achieved, which is the core means to solve the problem of unstructured text corpora. The
20 synergistic effect of the two ensures that the generated text base corpus can be seamlessly integrated into the fusion process of document templates, not only compensating for the deficiencies of corpora obtained solely from document image parsing in terms of content breadth and special formats (such as formulas and codes), but also greatly enhancing the linguistic diversity and professional scenario coverage of the finally generated document
25 recognition samples, thereby providing high-quality, highly effective training data support for the multimodal document recognition model.

[0093] In one or more implementations of this embodiment, a method for determining element style information and page layout information corresponding to any one of the multiple document images is further provided according to this embodiment. This method
30 aims to solve the technical problems of fragmented style in single images, implicit layout, and inconsistent extraction due to large differences in document types. Through a paradigm of

type guidance → rule inference → structure verification → information extraction, it ensures accurate and robust extraction of reusable and interpretable asset information from a single document image. The determining the element style information and page layout information corresponding to any one of the multiple document images includes the following steps 41 to 43.

[0094] In step 41, sub-element style information corresponding to multiple element style types are extracted from the document image according to a preset element style extraction rule, and the element style information is formed from the sub-element style information corresponding to the multiple element style types.

[0095] The preset element style extraction rule may refer to a predefined logical set for parsing visual attributes from visual data or document source files, whose function is to convert unstructured image pixel information into structured style parameters. The element style type may refer to classification of visual features corresponding to different categories of elements in a document, such as title class, body text class, table class, image class, with different element types focusing on different style aspects. The sub-element style information may refer to specific visual attribute values extracted for a particular element type, including font name, font size, font weight (e.g., bold, regular), italicization, font color, background color, alignment (left, center, right, justified), line spacing, paragraph spacing, indentation, border style, shadow effect, and transparency.

[0096] Specifically, the system first performs preliminary analysis on the document image through Optical Character Recognition (OCR) technology or a multimodal document understanding model, to locate various text blocks, image regions, and table regions. Then, for each located region, corresponding extraction strategy is invoked according to its element style type. For example, for a text-type element, the extraction rule focuses on analyzing character geometric features to infer font size and font weight, analyzing pixel color distribution to determine font color and background color, and calculating character baseline distances to determine line spacing; for a table-type element, the extraction rule focuses on detecting grid line styles, cell border thickness, and shading fill conditions.

[0097] For example, in a case of processing a document image of an academic paper, the system recognizes the region containing the text "Abstract" as a title-type element. According to a preset rule, the system extracts the sub-element style information of this region as font of

Times New Roman, font size of 14pt, font weight of bold, color of black, alignment of centered, and paragraph spacing after of 12pt. In addition, for body paragraphs, the system extracts font as Times New Roman, font size as 10.5pt, line spacing as 1.5 times, and first-line indent as 2 characters. The system aggregates the sub-element style information extracted for different element types such as titles, body text, and figure captions to form complete element style information. Through this type-specific and aggregation approach, fine-grained capture of document visual styles can be achieved, avoiding style loss or confusion caused by general rules, thus providing accurate style constraints for subsequent construction of high-fidelity document templates.

[0098] In step 42, the document type corresponding to the document image is determined, and inferred layout information corresponding to the document image is determined according to the document type.

[0099] The document type may refer to a category divided based on the content structure, application scenario, and typographic standard of a document, such as identity document, financial invoice, academic journal paper, technical report, book chapter, or legal document. Different document types generally follow specific industry typographic standards or fixed layout logic. Inferred layout information may refer to an expected page structure rule derived based on prior knowledge of the document type, including expected region division (such as header, footer, body region, sidebar region), column number setting (single-column, double-column, multi-column), reading order logic, and topological relationship constraint among elements.

[0100] Specifically, the system performs global analysis on the document image through a classification model or rule matching mechanism to identify its document type. Once the document type is determined, the system immediately loads the layout inference rule base bound to that type. These rule bases contain structural assumptions specific to that type of document. For example, if the document type is determined as an academic journal paper, the inferred layout information by default includes double-column layout, top header containing the journal name, bottom footer containing page numbers, charts generally spanning or embedded within a single column, etc.; if the document type is determined as an identity card, the inferred layout information locks fixed field positions, such as name, ID number, address regions having strict coordinate ranges and relative positional relationships.

[0101] For example, if an image of the first page of a conference paper is inputted, the system recognizes the conference paper as a double-column academic paper type. Accordingly, the generated inferred layout information includes: the page is divided into left and right columns by a vertical centerline, reading order is left column from top to bottom first, then right column from top to bottom; the title region spans both columns at the top; the abstract region follows the title and may span columns; the reference region is at the bottom of the page. This inferred layout information based on document type essentially injects prior knowledge of domain experts into the extraction process, so that the system no longer blindly segments the image but goes to verify and extract with expected structures, significantly improving the accuracy and generalization capability of layout understanding, solving the problem that general layout algorithms tend to fail when processing specially formatted documents.

[0102] In step 43, the document image is verified according to the inferred layout information, and page hierarchy structure information and region relationship information are extracted from the document image if the verification passes, to form the page layout information.

[0103] The verification may refer to a process of comparing the actual detected structure of the document image with the inferred layout information generated above, which is used to determine whether the actual image conforms to the standard structural logic of that document type. The page hierarchy structure information may refer to a parent-child membership relationship and hierarchical division of document content in logic, for example, a page consists of header, body, footer; the body consists of several paragraphs, chart blocks; a paragraph contains lines, words, characters, etc. The region relationship information may refer to spatial adjacency, inclusion, spanning, and semantic association relationships among functional regions on a page, such as an adjacency relationship between charts and their captions, a table cross-page continuation relationship, a reading flow connection relationship between columns.

[0104] Specifically, the system uses the inferred layout information as a verification template to check whether key features in the document image match. For example, checking whether expected header and footer regions exist, whether column width ratios of a double-column document are within an allowable error range, whether a key field of a

fixed-format document appear within a predetermined coordinate interval. If verification fails (e.g., the image is severely rotated, key regions are missing, or layout is chaotic), the image may be marked as an abnormal sample for manual review or discarded; if the verification passes, the deep extraction process is initiated to officially parse the hierarchy and region relationship of the page. In this case, the system not only records the coordinate boxes of each region, but also constructs a graph structure or tree structure describing a logical connection among them.

[0105] For example, in a double-column paper image that passes verification, the extracted page hierarchy structure information includes: the root node indicates the page, child nodes indicate the header, the left-column body, the right-column body, and the footer; where the left-column body has multiple paragraph nodes and chart nodes underneath. In addition, the extracted region relationship information includes: the region of paper image 1 and the below explanation region of the paper image 1 have a strong adjacency relationship, and are determined as the same semantic unit; Table 2 spans the bottom of the left column and the top of the right column, and is marked as a cross-column table, and the reading order is recorded as jumping continuously from the bottom of the left column to the top of the right column. Through this mechanism of verifying first and then extracting, it is ensured that the finally generated page layout information not only conforms to the common specifications of that type of document, but also faithfully reflects the specific content of the current image, effectively avoiding noise interference and laying a solid foundation for subsequent construction of structurally reasonable and logically consistent document templates.

[0106] As shown in the schematic diagram in Figure 2b, in a case of extracting page layout information, in order to ensure the accuracy of the extracted information, inferred layout may be performed first based on the document type, where different types correspond to different layout inference strategies. Then, verification rules are used for verification based on the inference result; if verification fails, manual adjustment may be introduced; if verification passes, information such as header regions, footer regions, body regions, sidebar regions, single-column layout, double-column layout, multi-column layout, grid layout, floating regions, page-table-caption association relationships, and image-caption association relationships may be determined. Subsequently, this part of layout information may be stored in the page layout library for use when generating templates.

[0107] In summary, through the synergistic effect of the above technical features, efficient conversion from a single document image to high-value layout asset is achieved. Specifically, by extracting sub-style information of multiple element types according to a preset rule, fine-grained capture of document visual details is achieved, solving the problem of fragmented styles; by first determining the document type and then generating inferred layout information, domain prior knowledge is introduced into the extraction process, solving the problem of layout ambiguity causing understanding bias; and by using the inferred layout information for verification and then extracting hierarchy and region relationships after passing, a dual-guarantee mechanism is constructed, which not only filters out abnormal data but also ensures that the extracted page hierarchy structure information and region relationship information are authentic and logically self-consistent. These three links are closely interconnected, so that the extracted element style information and page layout information are not only highly accurate, but also highly interpretable and reusable, and can directly support the automated construction of high-quality document templates, thereby improving the authenticity and diversity of synthetic training data, and ultimately serving the performance improvement of multimodal document recognition models.

[0108] In one or more implementations of this embodiment, the constructing a document template according to the element style information and the page layout information includes the following steps 51 to 54.

[0109] In step 51, the element style information is stored into an element style library, and the page layout information is stored into a page layout library.

[0110] The element style library is a structured database for centrally managing visual attribute data extracted from real documents; element style information may refer to a collection of visual attributes of various elements in a document, specifically including parameters such as font name, font size, font weight, italicization, font color, background color, alignment, line spacing, paragraph spacing, indentation, border style, shadow effect, and transparency. The page layout library is an asset library for storing page hierarchy structure and region relationship definitions; page layout information may refer to a spatial distribution rule of various elements on a page, including division of header regions, footer regions, body regions, sidebar regions, and layout modes such as single-column, double-column, multi-column, grid, as well as floating regions, table-caption association

relationships, image-caption association relationships. By storing discrete style and layout information in classified form, asset deposition and version management of documents are achieved, providing a data foundation for subsequent flexible combination of templates.

[0111] For example, when the system parses a PDF file of an academic paper containing complex tables, the extracted title style of Song Typeface, small fourth size, bold is stored in the element style library and marked as academic title style A, while its corresponding double-column layout with charts spanning columns is stored in the page layout library and marked as academic double-column layout B. This asset-based storage approach allows high-quality features from different source documents to be independently invoked and reused, avoiding the inefficient process of re-parsing original documents each time samples are generated.

[0112] In step 52, element style information to be combined is selected from the element style library according to a preset template construction rule, and page layout information to be combined is selected from the page layout library.

[0113] The preset template construction rules are a set of logical constraints for guiding legal and effective matching between styles and layouts, aiming to ensure consistency in visual logic and typographic norms of the combined document. Element style information to be combined is a specific style subset selected from the element style library according to the rules; page layout information to be combined is a specific layout subset selected from the page layout library according to the rules. The selection process is not random, but based on style compatibility checks and layout constraint satisfaction for intelligent matching.

[0114] For example, if the preset rule defines generating a vertical ancient book style sample, the template construction rule will forcibly select style information with Kai Typeface and vertical writing attributes from the element style library, and simultaneously select layout information with no header/footer, single-column full-width, and right-to-left column advancement from the page layout library. Conversely, if a modern financial report is to be generated, the rule will guide selection of style information with Dengxian typeface and clear table lines, and layout information with grid layout and fixed header. Through this rule-driven selection mechanism, restrictions of original documents can be flexibly transcended, allowing high-quality styles and layouts from different document sources to be recombined, thereby constructing mixed-feature samples that are rare in nature but urgently needed for model

training.

[0115] In step 53, a combined document template according to the element style information to be combined and the page layout information to be combined, and determine a preset reference document template.

5 **[0116]** The combined document template is a new structure definition file generated by fusing the selected element style information to be combined and the page layout information to be combined, containing complete page region division constraints, element type constraints, and element style constraints. The preset reference document template may refer to a pre-defined benchmark template in the system or an original template directly derived
10 from a single real document without combination changes, which is used as a baseline object for enhancement or comparison. The process of constructing a combined document template essentially instantiates abstract style parameters into a specific layout framework, to form an intermediate description file that can be directly read by the rendering engine.

[0117] For example, the system applies the selected tech-style color scheme and font style
15 set to a three-column news layout set, to generate a brand-new combined document template that defines the specific style performance of title, body text, and image in each of the three columns. In addition, the system may retain an uncombined original document template as a reference document template for subsequent differentiated enhancement or as a baseline for fidelity control. This step realizes the transformation from asset components to a complete
20 blueprint, providing a rich template pool for generating diverse samples.

[0118] In step 54, enhancement processing is performed on the reference document template and/or the combined document template, and a document template is determined according to an enhancement processing result.

[0119] The enhancement processing may refer to applying controlled perturbations or
25 mutation operations to style parameters or layout parameters in the template, aiming to simulate diversity variations of real-world documents and non-ideal factors in the imaging process, thereby improving the robustness of generated samples. Specific enhancement methods include but are not limited to style perturbation (such as fine-tuning font size, changing font weight, adjusting line spacing, introducing slight color deviations, simulating
30 ink diffusion or print blur effects) and layout fine-tuning (such as slightly shifting text block positions, changing column width ratios, introducing random page margin variations,

simulating binding occlusion regions, etc.). By performing the above enhancement processing on the reference document template and/or the combined document template, the system can derive a large number of variant templates with minor differences from one base template.

[0120] For example, for a standard contract document template, enhancement processing may generate a series of variants: in some variants, the signature block position is shifted down by 5 pixels; in others, the line spacing of clause texts is compressed by 10%; in still others, the transparency of the seal region is adjusted to semi-transparent to simulate a photocopy effect. These variants obtained through enhancement processing are determined as the document templates ultimately used for sample generation. This mechanism not only solves the problem of rigid single templates, but more importantly establishes a bridge connecting templates with hard sample generation, enabling the system to directionally generate hard samples that are difficult to collect in real scenes but crucial for model training (such as extreme layouts, blurred handwriting, complex mixed arrangements, etc.), thereby significantly improving the generalization ability of the multimodal document recognition model.

[0121] In specific implementations, referring to the schematic diagram shown in Figure 2c, when constructing document templates, to ensure the richness of document templates, the triples (document paragraph position, content, element type) obtained from the document image parsing stage may be used to aggregate document content by category. After determining the cluster corresponding to each category, categories may be traversed, and based on the traversal result, paragraph images of any category can be cropped from document images, styles can be recognized through a model, and styles passing rule verification can be selected as category styles; simultaneously, specific document templates may be determined in combination with page layout, and combined document templates may be determined based on category styles and page layouts in the page layout library; the specific document templates and the combined document templates may be fused to obtain an enhanced style template API, which can be subsequently invoked to generate high-quality and rich sample pairs.

[0122] In summary, through the above processing, the generated document templates not only possess structural rationality of real documents, but also have diversity and complexity beyond real distribution, effectively solving the technical problems in existing solutions such

as single template sources, lack of flexibility, and difficulty in covering hard scenarios, laying a solid foundation for subsequently generating high-quality, highly effective training samples for multimodal document recognition.

[0123] In one or more implementations of this embodiment, the determining a target base corpus according to the document base corpus and the text base corpus includes the following steps 61 to 64.

[0124] In step 61, a candidate base corpus is constructed from the document base corpus and the text base corpus.

[0125] The candidate base corpus may refer to a corpus collection formed by preliminarily fusing the document base corpus obtained from parsing real documents and the text base corpus obtained from pure text resources. The document base corpus typically contains paragraph texts with clear layout structure features and their category labels, such as titles, body text, table captions, reflecting the typographic habits of real documents; the text base corpus is obtained from knowledge bases, large language model generated content, or public corpus libraries, providing rich semantic content and diverse expressions. The process of constructing the candidate base corpus is to standardize these two heterogeneous data sources into a preset data format, such as a unified JSON structure, so that each corpus unit contains text content, paragraph category, and source identifier. By fusing dual-source corpora, both the structural constraint information of real documents and the semantic diversity of general texts can be retained, laying the content foundation for subsequently generating multi-element, multi-scenario document recognition samples.

[0126] For example, a balance sheet paragraph parsed from an annual financial report (document base corpus) and an asset liquidity analysis paragraph captured from a financial news library (text base corpus) are merged and stored in the same candidate pool. Thus, an initial corpus library with both structural authenticity and content richness is constructed, solving the problem of insufficient scenario coverage caused by a single data source.

[0127] In step 62, deduplication is performed on the candidate base corpus, and the deduplicated candidate base corpus is divided into specific document corpus and non-specific document corpus.

[0128] The deduplication operation aims to eliminate redundant information in the

candidate base corpus, preventing the model from overfitting to repeated samples during training. The deduplication strategy may include exact hash matching-based complete deduplication, and semantic vector similarity-based fuzzy deduplication. If the semantic similarity of two text pieces exceeds a preset threshold (e.g., 0.95), two text pieces are considered duplicates and only one is retained. After deduplication, the remaining corpus is divided into specific document corpus and non-specific document corpus according to the source attributes or business scenario features of the corpus. Specific document corpus may refer to corpora corresponding to specific business fields, specific document types (such as ID cards, driver's licenses, medical records, legal contracts), or having special layout structures. Such corpora often account for a small proportion in real data but are difficult to recognize. The non-specific document corpus may refer to general, scenario-unrestricted common text corpora. This division mechanism is a prerequisite for implementing differentiated corpus enhancement strategies.

[0129] For example, the system marks paragraphs such as the tax number and the amount parsed from a VAT invoice template as specific document corpus, and marks common descriptive text from general encyclopedias as non-specific document corpus. Through this step, refined stratification of the corpus library is achieved, clarifying the direction of hard samples that need to be enhanced in the subsequent process.

[0130] In step 63, similar corpus expansion processing is performed on the specific document corpus, and/or identical corpus enhancement processing is performed on the specific document corpus, and a target specific document corpus is determined according to a processing result.

[0131] The similar corpus expansion processing may refer to finding or generating new corpora that are semantically similar but differently expressed compared to the original specific document corpus by using semantic retrieval or generation models, so as to expand the coverage breadth of specific scenarios. For example, for the specific corpus "patient chief complaint", variants such as "patient self-reported symptoms", "reasons for visit description" that are semantically equivalent but worded differently are expanded. Identical corpus enhancement processing may refer to performing operations such as synonym rewriting, translation back-translation, sentence pattern transformation, or noise injection on the original specific document corpus, increasing diversity of expression forms without changing the core

semantics, to deepen the robust understanding of the model on such corpora. These two processing methods may be performed independently or in combination, aiming to solve the problem of scarcity of domain-specific hard samples. Through similar expansion, the system can cover more potential expression variants. Through identical enhancement, the system can simulate writing differences or recognition interferences that may occur in real scenes. For example, for a specific legal provision corpus, both similar expansion through retrieval of similar case libraries and generation of multiple versions through paraphrasing using a large language model can be performed. After the above processing, the original scarce specific document corpus is transformed into a rich, diverse, and comprehensive target specific document corpus, significantly improving generalization capability of the model in key business scenarios.

[0132] In step 64, the target specific document corpus and the non-specific document corpus are used as the target base corpus.

[0133] The target base corpus is the high-quality corpus collection ultimately used to drive the document synthesis engine. This step reintegrates the target specific document corpus that has undergone directional enhancement processing with the non-specific document corpus that remains unchanged or has undergone only basic cleaning. During integration, the weights of the two types of corpora may be adjusted according to a preset distribution ratio, ensuring that the generated training data contains both sufficient general samples to maintain the basic capability of the model and have high density of hard samples to break through performance bottlenecks of the model. The target base corpus not only retains the structural attribute of the document base corpus and the semantic attribute of the text base corpus, but also compensates for the shortcomings of specific scenario data through targeted enhancement processing.

[0134] For example, in a case of constructing a training set for medical document recognition, the target base corpus may include a large number of expanded diagnostic reports and prescription specific corpora, as well as an appropriate amount of general medical knowledge corpora. The target base corpus thus formed directly serves the subsequent document template fusion and sample rendering stages, ensuring that the generated document recognition samples are more balanced in content distribution and more targeted in hard scenarios, thereby effectively supporting the efficient training of multimodal document recognition models.

[0135] Referring to the schematic diagram in Figure 2d, during corpus processing, deduplication may be performed first on the base corpus to obtain corpus with unique semantic information; then through content classification and content warehousing, specific document corpus and pan-industry corpus may be obtained, so that sample construction can be performed subsequently in combination with these corpora. Further, as shown in the schematic diagram in Figure 2e, for specific corpora, similar corpus expansion and/or identical corpus enhancement may also be combined to complete corpus augmentation processing, so that subsequently batch corpora can be used to construct a larger number of higher-quality sample pairs, thereby improving the model training effect.

[0136] In summary, by fusing the document base corpus and the text base corpus, the silo effect of data sources is broken, and a content-rich candidate pool is constructed. In addition, through deduplication and the classification mechanism, specific document corpora with high value are precisely identified, providing a basis for differentiated processing. Further, with the help of similar corpus expansion and identical corpus enhancement technologies, the problems of scarcity and insufficient diversity of samples in specific scenarios are directionally solved, significantly improving the supply capacity of hard samples. Finally, the enhanced specific corpora and non-specific corpora are organically integrated to form a balanced and comprehensive target base corpus. This process not only eliminates redundant noise and ensures corpus quality, but more importantly establishes a corpus governance closed loop oriented to model weaknesses, so that the generated corpora can directly drive the directional generation of complex document samples in subsequent stages, thereby significantly improving the generalization performance and robustness of the multimodal document recognition model in real business scenarios.

[0137] In one or more implementations of this embodiment, the fusing the document template and the target base corpus by an element renderer and a layout manager according to a preset writing mode and a font resource library, to obtain a document recognition sample and a document recognition label, includes the following steps 71 to 73.

[0138] In step 71, the target base corpus is rendered by the element renderer according to the preset writing mode and the font resource library in combination with template style information in the document template, to obtain paragraph elements.

[0139] The element renderer may refer to a computing module for converting text corpora

into graphical elements with specific visual attributes. The template style information includes visual attributes such as font name, font size, font weight, color, italicization, alignment, line spacing, paragraph spacing, border style, and shading, derived from the data extracted from real documents and stored in the element style library. The writing mode may refer to the organization and expansion mode of text on a page, specifically including one or more of horizontal left-to-right, horizontal right-to-left, vertical top-to-bottom with columns right-to-left, and vertical top-to-bottom with columns left-to-right, and even supporting mixed writing modes within the same page. The font resource library contains character sets and glyph data covering multiple global languages and ancient/modern text forms.

[0140] Specifically, the element renderer first matches corresponding template style information from the document template according to the paragraph category of each paragraph in the target base corpus. Then, the renderer invokes font files from the font resource library that match the target language or text form, and converts plain text content into paragraph elements with precise geometric coordinates and visual styles. During this process, the element renderer executes a pre-rendering algorithm to obtain dimensional information of the element before full layout, and determines character arrangement direction and line breaking logic through a writing mode control algorithm. For example, when processing Arabic texts, the element renderer automatically invokes the horizontal right-to-left writing mode rule and corresponding Arabic font resources, ensuring correct character ligature form and reading order compliance. Similarly, when processing vertically written ancient texts, the renderer precisely calculates the baseline position and column advancement distance of each character according to the vertical top-to-bottom with columns right-to-left rule.

[0141] For example, the input end receives the target base corpus and template style information, and outputs paragraph elements with complete style attributes after internal state management and glyph mapping processing. Through this rendering mechanism, fine control over individual characters, single lines of text, and entire paragraphs can be achieved, ensuring that the generated paragraph elements highly restore the typographic features of real documents in visual appearance. This step aims to solve the problems of single style, missing fonts, and incorrect writing directions in traditional synthesis methods, providing high-fidelity basic components for subsequent layout.

[0142] In step 72, the paragraph element is rendered into the document template by the layout manager according to template layout information in the document template, to obtain a document recognition sample.

[0143] The layout manager may refer to a scheduling module responsible for coordinating the relative position and hierarchical relationship of multiple paragraph elements in page space. The template layout information includes page hierarchy structure, region division (such as header, body, sidebar, footer), adjacency relationship among regions, and reading order constraint, derived from the page layout library obtained from the aforementioned parsing.

[0144] Specifically, the layout manager receives the generated paragraph elements and arranges them according to the region constraint defined in the template layout information. The layout manager internally runs a region occupation algorithm to determine whether the current region can accommodate the paragraph element to be rendered. If space is insufficient, a cross-column cutting algorithm or cross-page cutting algorithm is triggered to intelligently split the paragraph element and flow it to the next column or next page, while maintaining semantic continuity. In addition, the layout manager executes a conflict rollback algorithm, when large-size elements such as images or tables overlap with text, the conflict rollback algorithm automatically adjusts the occupation state of preceding elements to avoid key elements. For example, when generating a double-column academic paper sample, the layout manager strictly follows the reading order of left column first, then right column, and when the remaining space at the bottom of the left column is insufficient to accommodate a complete formula block, the layout manager automatically moves the entire formula block to the top of the right column, leaving appropriate blank space or filling with other short texts in the original position, thereby avoiding structural damage caused by forced truncation.

[0145] For example, after receiving paragraph elements, the layout manager calculates the available space of each management region, dynamically adjusts the rendering coordinates of elements, and finally integrates all elements into a complete page image, i.e., the document recognition sample. This step, through strict layout constraint and intelligent conflict handling mechanism, ensures the rigor of the generated samples in structural logic, effectively avoiding element misalignment, overlap anomalies, or reading order confusion, so that the generated document recognition sample possess a complex and realistic layout structure.

[0146] In step 73, a document recognition label corresponding to the document recognition sample is determined based on the template style information and the template layout information.

[0147] The document recognition label may refer to annotation data synchronously generated while generating the document recognition sample, used to describe the category, position, content, and structural relationship of each element in the sample. Since the label is derived directly from the template style information and template layout information used in the generation process, rather than through post-hoc manual or model annotation, the label has extremely high accuracy and consistency.

[0148] Specifically, during the process in which the element renderer generates paragraph elements, the system has already recorded the precise coordinate, font attribute, and belonging paragraph relationship of each character and each line of text; during the layout arrangement process, the layout manager further records the page region, hierarchical index, and global reading order of each element. Based on these native data, the system directly constructs a multi-dimensional label system including element bounding boxes, category labels, text content, reading order sequences, paragraph hierarchy trees, and writing mode identifiers. For example, for a generated table sample, the label not only contains the overall bounding box of the table, but also precisely records the row-column span of each cell, the character-level coordinate of the internal text, and adjacency relationship between cells. For multi-page continuous document, the label also includes inter-page continuity identifier, indicating that a certain paragraph is continued from the previous page to the current page.

[0149] This step achieves the technical effect of "simultaneous generation and annotation", completely eliminating label noise caused by human errors or model recognition deviations in traditional post-annotation methods. By directly mapping template style information and layout information to label data, perfect alignment between labels and image pixels is ensured, providing a strong supervision signal for training the multimodal document recognition model, significantly improving the learning effectiveness of the model in fine-grained element recognition, complex layout analysis, and reading order recovery tasks.

[0150] It should be noted that since the target base corpus records the text content and type in the corpus, and the document template records element style information and page layout information, during fusion processing, corresponding titles, texts, abstracts, etc. may be

selected from the corpus according to the element types in the template and filled into the template, thereby obtaining high-quality samples. In addition, since the element positions, layout, etc. have already been determined during the parsing stage of documents and texts, this information may be used as labels to obtain sample pairs for training the model.

5 **[0151]** In summary, the element renderer uses the preset writing mode and font resource library to ensure visual authenticity and multilingual adaptability of the target base corpus when converted into paragraph elements, and the layout manager then precisely arranges these high-fidelity elements in complex page structures according to template layout information, solving structural challenges such as multi-element mixed arrangement,
10 cross-page continuity, and conflict avoidance. On this basis, the system directly uses the style and layout parameters inherent in the generation process to synchronously derive document recognition labels, fundamentally ensuring zero-error and high-precision labels. The three work closely together, making the generated document recognition samples not only visually realistic and structurally reasonable, but also pixel-level aligned in annotation, thereby
15 effectively supporting the training needs of multimodal document recognition models for hard samples (such as complex tables, mixed writing modes, ancient/modern texts, etc.), significantly reducing manual annotation costs and improving model iteration efficiency.

[0152] In one or more implementations of this embodiment, the element renderer includes at least one of the following algorithms (1) to (5).

20 **[0153]** (1) A rollback algorithm, configured to roll back a local paragraph element or a global paragraph element.

[0154] The rollback algorithm may refer to a processing mechanism that, when rendering conflicts, element overlaps, or layout constraint violations are detected during the rendering process, may revoke the rendering state back to a previous stable node. The algorithm
25 maintains a rendering state stack, recording the position coordinate, style attribute, and region occupation information of each paragraph element before rendering. When a subsequent element cannot be placed in the current region according to a preset rule, the rollback algorithm is triggered, deciding whether to roll back only the current local paragraph element or a global sequence of paragraph elements including the current one, depending on the
30 conflict type. For example, in a multi-column layout, if a long table attempts to cross the column end but the remaining space is insufficient to accommodate its header, the rollback

algorithm rolls the entire table back to the starting position of the next column or page, releasing the temporarily occupied space in the current column. In a mixed writing mode page, if a vertically written paragraph and a horizontally written image encroach on each other's regions, the algorithm can roll back the vertical paragraph to a backup region for rearrangement. Through this state rollback mechanism, the structural integrity of document generation can be ensured, avoiding page layout collapse caused by failure of a single element rendering, providing a clean initial state for subsequent truncation or rearrangement operations.

[0155] (2) A truncation algorithm, configured to decide and manage, when a global paragraph element cannot be rendered, the state of a paragraph element to be rendered.

[0156] The truncation algorithm may refer to processing logic that, during pre-rendering or formal rendering, when it is determined that the currently available region cannot fully accommodate the entire paragraph element, intelligently calculates the optimal truncation point and manages the state of the remaining content. The algorithm analyzes line-breaking rules, word boundaries, and semantic coherence of the paragraph content based on character-level, line-level, or semantic-level granularity, thereby determining the optimal truncation position. After truncation, the algorithm generates a pending state marker, and records the text content not yet rendered, style inheritance relationships, and expected continuation positions, so that rendering can continue on the next page or next region. For example, for a long body paragraph that spans pages, when the remaining height at the bottom of the page is only sufficient for three lines of text, the truncation algorithm cuts at the end of the third line at a complete word boundary, marks the remaining text as pending, and automatically resumes rendering at the top of the next page with the indentation and font attributes restored of the paragraph. In vertically written ancient document generation, if space at the column end is insufficient, the algorithm preferentially cuts at an appropriate sentence break based on punctuation principle. Through precise truncation decision and state management, seamless continuation of long content across multiple pages or regions is achieved, ensuring reading logic coherence.

[0157] (3) A pre-rendering algorithm, configured to perform pre-rendering processing on a paragraph element to be rendered.

[0158] The pre-rendering algorithm may refer to a predictive mechanism that, before

formally writing a paragraph element into the page layout, performs lightweight rendering on a virtual canvas or in memory to obtain the geometric dimension and attribute information of the paragraph element. The algorithm invokes the font resource library and style parameters, simulates the real rendering process, and calculates the precise width, height, baseline position, and line-breaking condition of the element to be rendered under a specific writing mode, without actually generating final image pixels. These prior data are passed to the layout manager for evaluating whether the current region can accommodate the element, thereby guiding layout decisions. For example, when processing a complex table cell, the pre-rendering algorithm first calculates the minimum cell height required for text containing a specific number of characters and font size; if it is found to exceed the preset row height, the pre-rendering algorithm notifies the layout manager in advance to adjust row height or trigger the truncation process. In mixed-lingual scenarios, the pre-rendering algorithm can quickly estimate the total width required for mixed paragraphs of Arabic (right-to-left) and Chinese (left-to-right), preventing layout overflow due to directional differences. Through the precise spatial prediction provided by pre-rendering, the accuracy of layout planning and the success rate of one-time rendering are significantly improved, reducing trial-and-error costs.

[0159] (4) A writing mode control algorithm, configured to determine a sub-writing mode corresponding to a paragraph element to be rendered.

[0160] The writing mode control algorithm may refer to a logic unit that, according to the document template definition or paragraph attributes, dynamically identifies and assigns the specific writing direction to be followed by each paragraph element to be rendered. The algorithm supports coexistence of multiple writing modes on the same page or even within the same region, including horizontal left-to-right, horizontal right-to-left, vertical top-to-bottom with columns right-to-left, and vertical top-to-bottom with columns left-to-right. During the rendering process, the algorithm switches character arrangement engines, line-breaking logic, and cursor movement rules in real time, ensuring that texts of different languages are correctly presented according to their inherent writing habits. For example, on a document page containing Chinese body text and Arabic quotations, the writing mode control algorithm automatically identifies the Chinese paragraph to adopt horizontal left-to-right mode, and immediately switches to horizontal right-to-left mode when an Arabic paragraph is identified, and accordingly adjusts punctuation positions and number arrangements. When generating Japanese ancient document samples, the algorithm can set the main text to vertical

top-to-bottom with columns right-to-left mode, while embedding English annotations in horizontal left-to-right mode within it. Through fine-grained sub-writing mode control, unified modeling and high-quality generation of global multilingual and ancient/modern text forms are achieved.

5 **[0161]** (5) A first information extraction algorithm, configured to extract an element to be rendered according to a preset information extraction granularity.

[0162] The first information extraction algorithm may refer to a processing module that, during or after element rendering, simultaneously collects and outputs multi-level structured data of the element. The algorithm supports setting extraction granularity on demand,
10 covering character-level, word-level, line-level, paragraph-level, and even cell-level, and can precisely obtain geometric coordinate (such as bounding box, baseline, rotation angle), style attribute (such as font, font size, color, character spacing), and topological relationship (such as belonging line, belonging paragraph, adjacency relationship) of each unit. The extracted information directly serves as part of the document recognition label, strictly corresponding to
15 the generated document image. For example, when rendering a mathematical formula element, the first information extraction algorithm not only records the overall bounding box of the formula, but also extracts the character-level coordinate and font style of each symbol, as well as superscript/subscript relationships between symbols. When generating a complex table, the algorithm extracts row-column index, row-span/column-span attribute of each cell, and
20 line-level bounding box information of text within the cell. Through such high-precision information extraction, it is ensured that the generated samples possess character-level and structure-level supervision signals, providing a solid data foundation for training high-precision multimodal document recognition models.

[0163] In summary, the pre-rendering algorithm, as an intelligent front-end step, provides
25 precise spatial priors for layout planning, avoiding resource waste caused by blind rendering. The writing mode control algorithm, as a globalization key, dynamically adapts typographic rules of different languages, ensuring correct presentation of multilingual content. The rollback algorithm and truncation algorithm constitute the fault tolerance foundation, the former flexibly revoking states in case of conflicts, the latter intelligently splitting content
30 when space is limited. The two together solve rendering anomalies under long paragraphs and complex layouts, ensuring the stability of the generation process. The first information

extraction algorithm, as the source of label synchronization, captures fine-grained features during rendering, achieving high consistency between data and labels. On this basis, with the close coordination among algorithms, the system can upgrade from linear rendering to closed-loop controlled intelligent generation, not only significantly improving the success rate of document sample generation and visual authenticity, but also ensuring the high quality and traceability of output data in structure, style, and semantics, thereby effectively supporting the training needs of multimodal document recognition models in complex scenarios.

[0164] In one or more implementations of this embodiment, the layout manager includes at least one of the following algorithms (1) to (6).

[0165] (1) A writing mode algorithm, configured to manage the writing result of a paragraph element according to the writing mode.

[0166] The writing mode algorithm may refer to a logic module in the layout manager for parsing and executing specific text arrangement rules. The algorithm functions to dynamically calculate the occupying position and advancement direction of paragraph elements on the page according to a preset writing mode (including one or more of horizontal left-to-right, horizontal right-to-left, vertical top-to-bottom with columns right-to-left, vertical top-to-bottom with columns left-to-right). Specifically, when it is detected that the style information of the current paragraph element specifies a vertical writing mode, the writing mode algorithm controls characters to stack vertically and controls column advancement direction from right to left or left to right, ensuring that the generated document conforms to the typographic habits of specific languages or historical documents. For example, when generating ancient document text, the writing mode algorithm sets the baseline of text lines to be vertically arranged and forces subsequent text columns to extend to the left. If the current column space is insufficient, it automatically rolls back to the previous column or jumps to the corresponding starting position of the next page. Through coordination between the writing mode algorithm and layout constraints in the document template, it is ensured that in multilingual mixed or special writing scenarios, the visual presentation and reading logic of paragraph elements are strictly consistent, avoiding semantic confusion caused by directional errors.

[0167] (2) A reading layout algorithm, configured to manage the layout order of paragraph elements.

[0168] The reading layout algorithm may refer to sorting logic for maintaining the semantic coherence of document content. The algorithm functions to strictly maintain the relative order of text elements (such as titles, body text, lists) during rendering, as well as the adjacency relationships between images, tables, and their captions. Specifically, the reading layout algorithm constructs a priority queue based on reading flow, ensuring that in complex image-text mixed scenarios, even if element positions are dynamically adjusted, the logical reading order still conforms to human cognitive habits. For example, when there is a combination of image - image caption - body paragraph on a page, the reading layout algorithm forces the relative order of these three to be locked, preventing the body paragraph from being accidentally inserted between the image and its caption; in double-column layout, the algorithm ensures that the end of the first paragraph and the beginning of the second paragraph are logically continuous, rather than jumping. Through coordination between the reading layout algorithm and the aforementioned writing mode algorithm, not only orderly arrangement of elements within a single page is ensured, but also a foundation for cross-page reading order of multi-page continuous documents is provided.

[0169] (3) A region occupation algorithm, configured to manage the region occupation state in the document template according to region occupation information of paragraph elements.

[0170] The region occupation algorithm may refer to a dynamic management mechanism for real-time tracking and updating of available page space. The algorithm functions to determine, based on the geometric dimension (width, height) and shape of the paragraph element to be rendered, whether it can be placed in various regions (such as single-column areas, double-column areas, grid areas) of the current document template, and mark occupied states. Specifically, the region occupation algorithm divides the page into several logical management units; whenever an element is successfully rendered, the algorithm immediately updates the occupation flag of the corresponding unit and calculates the remaining available space. For example, in a template containing a header, double-column body, and footer, after a title element occupies the top full-width region, the region occupation algorithm immediately reduces the available height limit of the left and right columns below, preventing subsequent body text from overflowing. If a paragraph element needs to span two columns, the algorithm simultaneously locks the continuous space of both columns. Through coordination between the region occupation algorithm and the reading layout algorithm, efficient use of page space

is achieved, ensuring compactness and normalization of element layout.

[0171] (4) A conflict rollback algorithm, configured to roll back to a state before a rendering node in case of a rendering conflict.

[0172] The conflict rollback algorithm may refer to a retry mechanism for handling rendering exceptions and restoring the scene. The algorithm functions to, when attempting to place a large element (such as a large image, complex table) and finding insufficient space in the current region or overlap with other elements, cancel some previous rendering operations and roll back the layout state to a safe node before the conflict occurred. Specifically, when the region occupation algorithm determines that the current remaining space cannot accommodate the element to be rendered, or the pre-rendering result indicates that occlusion will occur, the conflict rollback algorithm is triggered. The algorithm moves the layout pointer back to the end of the previous complete paragraph or the start of the page, and re-plans the flow of subsequent elements. For example, when the header of a cross-page table is exactly at the page end and the table body content is too large to be completely displayed on the current page, the conflict rollback algorithm migrates the entire table (including the header) to the top of the next page, rather than forcibly truncating it and causing layout fragmentation. Through coordination between the conflict rollback algorithm and the region occupation algorithm, layout disorder is effectively avoided, improving the robustness and aesthetics of document generation.

[0173] (5) A cutting algorithm, configured to cut a paragraph element that does not meet the display requirement.

[0174] The cutting algorithm may refer to a splitting logic for achieving cross-column or cross-page continuous display of long content. The algorithm functions to semantically and reasonably split paragraph elements whose length exceeds the current available region limit (such as long text paragraphs, multi-row tables), enabling them to continue to the next column or next page. Specifically, the cutting algorithm analyzes the internal structure of the element, finds the optimal breakpoint (such as sentence terminators, natural paragraph pauses, table row boundaries), ensuring that the two parts after cutting are still semantically coherent. For example, when a body text extends to the column end, the cutting algorithm does not cut in the middle of a word or formula, but traces back to the previous complete word or sentence boundary for cutting, and automatically flows the remaining content to the start of the next

column. For table elements, the algorithm splits data rows by page while keeping the header repeated. Through coordination between the cutting algorithm and the conflict rollback algorithm, the space limitation problem is solved while ensuring the completeness and readability of document content, particularly suitable for multi-page continuous document generation scenarios.

[0175] (6) A second information extraction algorithm, configured to extract a rendered element, page information and region information according to a preset information extraction granularity.

[0176] The second information extraction algorithm may refer to an output interface for synchronously collecting and recording metadata of rendering results. The algorithm functions to extract detailed geometric positions, style attributes, and hierarchical relationship information from documents that have been laid out by the layout manager according to a preset granularity (such as character-level, line-level, paragraph-level, cell-level, page-level, region-level), and convert them into document recognition labels. Specifically, after the layout manager completes the arrangement of all elements, the second information extraction algorithm traverses the rendering tree, recording the coordinate box of each character, the baseline position of each line of text, the row-column span of each cell, the boundary range of each region, and adjacency relationships among elements. For example, for a rendered table, the algorithm can not only extract the circumscribed rectangle of the entire table, but also precisely extract the upper-left coordinates, width, height, merge status of each cell, and character-level position information of text within cells. For pages with mixed writing modes, the algorithm can also record the writing mode labels corresponding to different regions. Through coordination between the second information extraction algorithm and the aforementioned five scheduling algorithms, synchronous generation from visual rendering to structured labels is achieved, providing precise data support for subsequently constructing high-quality document recognition sample pairs.

[0177] In specific implementation, referring to the schematic diagram shown in Figure 2f, after obtaining the page template (document template) containing layout information and style information, the corpus text and page template may be processed using the layout manager and element renderer in combination with the writing mode and font libraries for sample pair construction. During this process, different elements may correspond to different element

renderers, such as text renderer, image renderer, formula renderer, table renderer, chart renderer, code renderer, list renderer, seal renderer, icon renderer, combined with the rollback algorithm, truncation algorithm, writing mode control algorithm, and information extraction algorithm involved in the renderers to complete the rendering operation of paragraph elements.

5 The layout manager may handle layouts such as single-column, double-column, grid, mainly involving the writing mode algorithm, reading layout algorithm, region occupation algorithm, cutting algorithms (cross-column cutting and cross-page cutting), and information extraction algorithm, to achieve multi-page image and image annotation operations, and then through the above processing determine samples and labels for subsequent model training.

10 **[0178]** In summary, the writing mode algorithm establishes the baseline text flow direction; the reading layout algorithm locks the semantic logical order on this basis; the region occupation algorithm monitors spatial resource allocation in real time; the conflict rollback algorithm provides flexible correction capability for space insufficiency; the cutting algorithm solves the problem of continuous display of long content; and the second information
15 extraction algorithm converts the final layout result into supervised structured data. This synergistic effect enables the layout manager not only to adapt to multiple complex global writing modes such as horizontal and vertical, but also to maintain layout logical correctness and visual regularity when handling hard scenarios such as image-text mixed arrangement, cross-page tables, and dense element arrangement, thereby significantly improving the
20 authenticity and diversity of generated samples, effectively solving the technical problems that traditional layout engines tend to make errors and have difficulty coordinating when facing complex document constraints.

[0179] In one or more implementations of this embodiment, the constructing a document recognition sample pair based on the document recognition sample and the document
25 recognition label includes the following steps 81 to 84.

[0180] In step 81, element positions in the document recognition sample are kept unchanged, and position-invariant enhancement information is added to the document recognition sample to obtain a first enhanced document recognition sample.

[0181] Position-invariant enhancement may refer to perturbing the visual attribute of the
30 document recognition sample while maintaining the geometric coordinates, relative positional relationships, and topological structures of elements (such as characters, text lines, paragraphs,

table cells, images, etc.) unchanged. Position-invariant enhancement information includes but is not limited to Gaussian noise, salt-and-pepper noise, motion blur, defocus blur, resolution reduction, brightness adjustment, contrast adjustment, local shadow occlusion, color dithering, or compression artifacts. By introducing the above position-invariant enhancement information, image quality degradation caused by imaging device sensor noise, illumination changes, transmission compression loss, or lens stains in real scenes can be simulated, without destroying the original layout structure and reading order.

[0182] For example, when the document recognition sample contains dense table data, adding Gaussian noise can simulate shooting noise in low-light environments. In addition, the bounding box coordinates of table cells remain unchanged, but the pixel values within the cells fluctuate randomly. Similarly, for a document sample containing a handwritten signature, adding local shadow occlusion simulates the effect of a finger blocking the lens, the geometric position of the signature region does not shift, but the grayscale values of some strokes are reduced.

[0183] As shown in the schematic diagram in Figure 2g, an original sample generated only by combining corpus and document template without interference effects is included. In order to simulate real document presentation effects, noise can be added to the original sample to achieve effects such as simulating real shadows, occlusions, etc., to generate more realistic samples, so that training the multimodal document recognition model with these samples enables the model to learn to overcome the effects of shadows, occlusions, etc., improving document recognition accuracy. For example, the resolution of the original sample can be reduced to obtain a blurred sample; interference lines can be added to the original sample to obtain an interference sample; a crease effect can be added to the original sample to simulate photographed or book text, to obtain a crease sample; a shadow can be added to the original sample to simulate shadow occlusion effects, to obtain a shadow sample; and so on, samples covering different enhancement effects can be obtained, so that subsequently these high-quality samples can be used to train the model, improving the recognition effect of the model on real data.

[0184] With this processing method, it is ensured that while enhancing the diversity of image quality, the originally generated character-level, line-level, and cell-level labels can be directly reused without coordinate transformation, ensuring strict alignment between labels

and enhanced image content. This step aims to improve the robustness of feature extraction of the multimodal document recognition model in complex imaging environments by simulating non-geometric imaging interference.

[0185] In step 82, element positions in the document recognition sample are adjusted, and position-variant enhancement information is added to the document recognition sample to obtain a second enhanced document recognition sample.

[0186] Position-variant enhancement may refer to applying geometric transformations to elements in the document recognition sample, changing their absolute coordinates or relative arrangement relationships on the page, and simultaneously applying corresponding visual perturbations. Operations corresponding to position-variant enhancement information include rotation (e.g., arbitrary angle rotation), affine transformation, perspective transformation, elastic distortion, random cropping and filling, random page margin adjustment, or local region displacement. When performing position adjustment, the system calculates the transformation matrix in real time and applies the matrix synchronously to the document recognition label bound to the sample (such as bounding box coordinates, keypoint positions, polygon vertices), thereby ensuring that the enhanced image content and label information remain precisely matched in space. For example, for a document recognition sample written in horizontal left-to-right mode, applying a clockwise 5-degree rotation and adding slight motion blur changes the tilt angle of all text lines; the system then uses the same rotation matrix to update the four vertex coordinates of all text line labels, so that the label boxes tightly enclose the rotated text lines. For a sample containing a cross-page table, applying elastic distortion to simulate paper bending deformation changes the shapes of table cells nonlinearly, and the corresponding cell-level labels also undergo nonlinear coordinate mapping. Through this position-variant enhancement, real physical scenarios such as user shooting at an angle, paper creases, scanner feed skew, or uneven document placement can be simulated. This step effectively solves the problem of poor model generalization due to lack of geometric diversity in traditional fixed-layout training data, significantly enhancing the model's adaptability to various geometric distortions.

[0187] In step 83, a first document recognition sample pair is constructed based on the first enhanced document recognition sample and the document recognition label, and a second document recognition sample pair is constructed based on the second enhanced document

recognition sample and the document recognition label.

[0188] Constructing a sample pair is a process of structurally binding document images that have undergone different enhancement strategies with their corresponding standard labels. For the first enhanced document recognition sample, since element positions have not changed, the original document recognition label generated in the previous step can be directly reused to form the first document recognition sample pair, which focuses on training content recognition stability of the model under noise interference. For the second enhanced document recognition sample, the label that has undergone geometric transformation correction (i.e., the label with coordinates updated according to the position-variant enhancement information) needs to be bound with this sample to form the second document recognition sample pair, which focuses on training the localization and recognition capabilities of the model under deformation interference. During this process, the content of the label (such as character sequences, category attributes, reading order indices) remains unchanged, and only its spatial description information (such as coordinate values) is updated depending on the enhancement type.

[0189] For example, in the first document recognition sample pair, although the image has added noise, the title-body-table hierarchical structure label inside it is completely consistent with that before enhancement; while in the second document recognition sample pair, after perspective transformation, the originally rectangular table region becomes trapezoidal, and the region coordinates in the corresponding table structure label are accordingly adjusted to trapezoidal vertex coordinates, but the internal row-column logical relationship labels of the table remain unchanged. By separately constructing these two types of sample pairs, comprehensive data coverage from noise resistance to deformation resistance is achieved, ensuring the completeness of training data distribution.

[0190] In step 84, the first document recognition sample pair and the second document recognition sample pair are used as the document recognition sample pair.

[0191] The above two types of sample pairs are collectively referred to as document recognition sample pairs, which means that the final sample library simultaneously contains diversified data in two dimensions: position-invariant enhancement and position-variant enhancement. This integration process is not a simple physical stacking, but constructs a training dataset with dual robustness characteristics. By jointly incorporating the first

enhanced document recognition sample pairs (focusing on texture and pixel-level robustness) and the second enhanced document recognition sample pairs (focusing on geometric and structural-level robustness) into the sample library, the subsequent multimodal document recognition model can simultaneously learn the capability to resist imaging noise and adapt to various shooting angles and paper deformations during training.

[0192] For example, in a training batch, the model may simultaneously receive a clear layout sample with Gaussian noise and a bent layout sample with perspective distortion, forcing the model encoder to extract deep semantic features that are neither disturbed by noise nor affected by geometric deformation. The document recognition sample pairs finally formed through this step not only retain the high-precision label advantages of the original generated data, but also greatly expand the distribution range of data through targeted enhancement strategies, effectively avoiding model overfitting to idealized synthetic data, thereby showing higher recognition accuracy and stability when facing complex and diverse real document scenes in practice.

[0193] In summary, through the synergistic effect of position-invariant enhancement and position-variant enhancement, a multi-level data augmentation system is constructed. By keeping element positions unchanged and adding position-invariant enhancement information such as noise and blur, random interference during imaging is simulated, ensuring that the model can still stably extract text and structural features when facing low-quality images. In addition, by adjusting element positions and applying position-variant enhancement information such as rotation and warping, and synchronously updating label coordinates, geometric distortion in real acquisition scenarios is simulated, endowing the model with strong spatial transformation adaptability. These two enhancement methods complement each other; the former ensures absolute label consistency under pixel perturbation, while the latter verifies dynamic label following under geometric transformation. Together, they transform a single synthetic sample into training data with high generalization value. On this basis, the two enhancement results are respectively constructed as first and second document recognition sample pairs and uniformly stored, forming a complete data closed loop covering static anti-noise and dynamic anti-deformation, significantly improving the robustness and accuracy of the multimodal document recognition model in complex real-world scenarios.

[0194] In one or more implementations of this embodiment, after the step of storing the

document recognition sample pair into a sample library, the method further includes the following steps 91 to 93.

[0195] In step 91, a target document recognition sample pair is extracted from the sample library.

5 **[0196]** The target document recognition sample pair may refer to a sample set selected from the massive stored document recognition sample pairs according to a preset screening strategy for the current training cycle. The source of target document recognition sample pairs may be full random sampling, or directional mining of hard samples based on the active learning strategy. Specifically, the system may, according to the performance of the multimodal
10 document recognition model in the previous training round, statistically analyze error distributions under different templates, style combinations, or writing modes, locate sample regions where the model has low confidence or inconsistent predictions, and extract the corresponding target document recognition sample pair from the sample library.

[0197] For example, when it is detected that the accuracy of the model significantly drops
15 on ancient documents with vertical top-to-bottom with columns right-to-left writing mode, the extraction strategy preferentially screens target document recognition sample pairs containing that writing mode feature from the sample library. Through this extraction mechanism based on feedback on model weaknesses, it is ensured that the data input into the training process has the highest optimization value, avoiding repeated training on simple samples that the
20 model has already mastered, thereby significantly improving data utilization efficiency.

[0198] In step 92, the target document recognition sample in the target document recognition sample pair is inputted into a multimodal document recognition model for processing, to obtain predicted document recognition information.

[0199] The multimodal document recognition model is a neural network model to be
25 trained or currently being iterated, which typically includes a visual encoder, a text decoder, and a cross-modal fusion module, and is used for joint processing of document images, layout structures, and text semantics. The target document recognition sample is the document image with rich annotation information generated in the previous steps. The predicted document recognition information may refer to the output results after the model performs forward
30 inference on the input sample, including but not limited to character recognition sequences, element category labels, bounding box coordinates, reading order sequences, table structure

trees, and key-value relationship pairs.

[0200] Specifically, the processor inputs the target document recognition sample into the model, the model internally extracts image features through convolutional layers or Transformer layers, combines predefined writing mode constraints and font resource prior knowledge, and decodes layer by layer to generate prediction content. For example, for a target document recognition sample containing a complex table, the predicted document recognition information output by the model not only includes the text content within cells, but also includes row-column span information of cells and topological relationships of adjacent cells. This step aims to transform static sample data into dynamic prediction results through the model's actual inference behavior, providing an error calculation basis for subsequent parameter updates.

[0201] In step 93, the multimodal document recognition model is trained according to the predicted document recognition information and the target document recognition label in the target document recognition sample pair, until a target multimodal document recognition model meeting a training stop condition is obtained.

[0202] The target document recognition label is the ground-truth data synchronously generated with the target document recognition sample, with granularity covering character-level, line-level, paragraph-level, cell-level, and page-level, and has extremely high accuracy and consistency, serving as supervision signals for guiding model optimization. The training process includes calculating a loss function value between the predicted document recognition information and the target document recognition label, and updating model parameters using a back propagation algorithm. The loss function construction can comprehensively consider text recognition errors (such as CTC loss or cross-entropy loss), position regression errors (such as Smooth L1 loss), and structure matching errors (such as Hungarian matching loss). Training stop conditions may refer to preset convergence criteria, such as the comprehensive evaluation metrics on a validation set (e.g., Hmean, F1-score) no longer improving, the loss function value falling below a set threshold, or reaching the maximum number of training epochs.

[0203] Specifically, the system continuously monitors various metrics during the training process, when the model performs stably on a validation set containing multiple writing modes and complex layouts and meets the stop condition, the current multimodal document

recognition model is determined as the target multimodal document recognition model. For example, if the stop condition is set such that the validation set accuracy fluctuation is less than 0.1% for 5 consecutive epochs, once this condition is met, training automatically terminates and the model weight is saved. Thus, through high-precision synchronous labels and directionally extracted hard samples, model convergence can be effectively accelerated, and it is ensured that the finally obtained target multimodal document recognition model has excellent generalization capability in real complex scenarios.

[0204] In summary, extracting the target document recognition sample pair from the sample library ensures the relevance and effectiveness of training data, solving the problem of low training efficiency caused by blind accumulation of samples in traditional methods. In addition, using the synchronously generated high-precision target document recognition labels as supervision signals eliminates the interference of manual annotation noise on model training, enabling the model to learn more refined character-level localization and complex layout structure features. Finally, driving model iteration based on the difference between prediction information and ground-truth labels until the stop condition is met achieves an automated training process oriented to model performance improvement. This mechanism not only verifies the practical application value of the document recognition sample pairs generated above, but also guides the update and expansion of the sample library in reverse through active learning feedback, ensuring continuous production of effective data that can address model weaknesses, thereby significantly improving the comprehensive performance of the multimodal document recognition model in tasks such as global writing mode, multilingual mixed arrangement, and complex table parsing, while reducing labor costs.

[0205] In one or more implementations of this embodiment, the method further includes the following steps 1 to 2.

[0206] In step 1, evaluation information corresponding to each document recognition sample pair in the sample library is determined according to a training result of the target multimodal document recognition model.

[0207] The evaluation information may refer to metric data used to quantitatively characterize the contribution or difficulty level of a document recognition sample pair during model training. This evaluation information is obtained by analyzing the difference between the prediction output of the model and the ground-truth label when processing each document

recognition sample pair in the sample library. Specifically, evaluation information may include dimensions of the model for a specific sample pair such as the confidence score, prediction error rate, category confusion degree, bounding box regression deviation, and structure parsing failure rate. For example, for a document recognition sample pair containing a complex cross-page table, if the model consistently fails to correctly recognize cell merge relationships across multiple training iterations, resulting in a structure parsing failure rate above a preset threshold, the sample pair is marked as high difficulty, and its evaluation information will record the specific error type (such as cross-page row breakage or cell ownership error) and error frequency. Furthermore, evaluation information may be refined to the template level, statistically analyzing the overall performance of sample pair derived from the same document template or similar element style combination, thereby identifying which template generate samples that are currently the weak points of the model. Through this fine-grained evaluation mechanism, the system can precisely determine the capability shortcomings of the model under specific layout structure, specific writing mode, or specific element combination, providing data support for subsequent constraint condition construction.

[0208] In step 2, a template constraint condition is constructed according to the evaluation information, where the template constraint condition is used to constrain a generated document template in a next sample construction cycle.

[0209] The template constraint condition may refer to a set of parameters used to limit, guide, or weight the document template generating rule in the next round of sample construction. This template constraint condition is derived inversely from the evaluation information determined above, and its function is to convert model-identified weak points into specific generation control instruction. Specifically, when the evaluation information shows a high error rate for a certain type of page layout (such as double-column mixed with floating images), the constructed template constraint condition will increase the selection probability weight of that page layout type in the next cycle, or forcibly require that document templates in the next cycle must include such layout features. If the evaluation information shows that a certain element style combination (such as extremely small font size with low-contrast background) leads to recognition failure, the template constraint condition may specify that templates with such style features are to be generated with emphasis in the next cycle.

[0210] For example, if the evaluation information indicates that the model performs poorly on ancient text recognition under vertical top-to-bottom with columns right-to-left writing mode, the constructed template constraint condition will specify that in the next sample construction cycle, at least 30% of the document templates must adopt that specific writing mode, paired with content from an ancient corpus for rendering. In addition, the template constraint condition may also include specific numerical range limits for element density, cross-page continuity complexity, and strategy for adding interference items (such as noise, watermarks). In this way, the system realizes a transformation from passive random generation to active directional supplementation, ensuring that newly generated document recognition sample pairs can target the current blind spots of the model.

[0211] In summary, the above processing not only significantly improves the relevance and value density of training data, but also endows the sample construction system with self-evolution capability, dynamically adjusting generation strategy as model capability improve, continuously mining deeper hard samples, thereby maximizing the acceleration of performance convergence and generalization improvement of the multimodal document recognition model while reducing manual intervention cost.

[0212] In addition, during the training of the multimodal document recognition model, considering that the training data of the multimodal document recognition model comes from the sample library, and there may be issues such as uneven distribution of samples in the sample library or unstable sample extraction rules during training, resulting in high recognition accuracy for documents of one dimension but low accuracy for documents of another dimension. For example, there are many paper samples but few receipt samples, so the recognition accuracy of the model for receipt documents will be poor, and after deployment the model will not meet the recognition requirement for receipt documents. To address this problem, most existing techniques use resampling of samples to fine-tune the model, but the resampling stage requires a significant amount of time and cost to collect targeted samples for model tuning, greatly affecting model training speed. In this regard, a technical solution for automatically generating effective samples to fine-tune the model is provided in this embodiment, which can determine the weak directions of the model in combination with strategies, and then directionally generate high-quality samples for continued model optimization. In this embodiment, the method further includes:

determining, according to the training result of the target multimodal document recognition model, recognition error statistics information corresponding to the target multimodal document recognition model; determining a reference effective sample corresponding to the recognition error statistics information, and constructing an effective sample set according to the reference effective sample; and optimizing the target multimodal document recognition model using the effective sample set, until a business multimodal document recognition model meeting a deployment condition is obtained.

[0213] Specifically, recognition error statistics information specifically refers to statistical information of recognition errors for different samples during the training phase of the target multimodal document recognition model. Correspondingly, a reference effective sample specifically refers to a sample of a certain type for which the recognition accuracy of the model is below a threshold. Based on this reference effective sample, new samples of similar types can be generated to form an effective sample set to optimize the model. It can be understood that the reference effective sample reflects that the recognition accuracy of the model for a certain type of document is low, so the model needs to be continually trained on that type of document to adapt to recognition of that type of document. Correspondingly, the deployment condition specifically refers to the condition set for deploying the multimodal document recognition model, such as the recognition accuracy of the model for a specific type of document reaching 90%, or the number of model optimization iterations, etc., which is not limited in this embodiment.

[0214] Based on this, during the model training phase, in order to enable the multimodal document recognition model to cover a wider range of fields and achieve higher recognition effects for different types of documents, after a certain stage of training, recognition error statistics information corresponding to the target multimodal document recognition model can be determined according to the training result of the target multimodal document recognition model. The recognition error statistics information can accurately reflect the recognition accuracy of the model on different types of documents, and then the reference effective sample corresponding to the recognition error statistics information can be determined. Through the reference effective sample, the document types with low recognition accuracy at the current stage can be determined, and then an effective sample set can be constructed according to the reference effective sample. Subsequently, the effective sample set is used to optimize the target multimodal document recognition model until a business multimodal

document recognition model meeting the deployment condition is obtained.

[0215] In specific implementation, when determining the reference effective sample, an active learning strategy may be used to integrate multiple results stability, single prediction results, and annotation result comparison strategies to record effective strategies, effectively recording error information on different samples, error statistics on different style combinations, error statistics on different element types, error statistics on different layout structures, and active learning based on the error information to precisely determine the reference effective sample for which the current recognition of the model is weak, thereby enabling subsequent batch construction of similar samples for model training, reducing the screening time of effective samples.

[0216] Further, when constructing the effective sample set according to the reference effective sample, reference may be made to the sample construction process described above, which is not repeated in this embodiment. It should be noted that since the reference effective sample corresponds to a fixed document template, when constructing effective samples in the effective sample set, the document template may be directly reused, and then new text corpora may be combined with the document template to batch generate effective samples for model training. The construction of new text corpora may be achieved through similar corpus expansion or identical corpus enhancement, the descriptions of which can be found in the same or corresponding descriptions in the above embodiments, and will not be repeated in this embodiment.

[0217] In addition, when constructing the effective sample set, considering that there may be multiple reasons for the low recognition accuracy of the model on that type of document, in order to solve all problems in one training session, data augmentation may also be performed on the constructed effective samples to simulate real document presentation effect, thereby constructing high-quality effective sample set, which can improve model robustness more effectively. The description of data augmentation may be found in the description of sample enhancement in the above embodiments, and will not be repeated in this embodiment.

[0218] For example, after the multimodal document recognition model is trained with samples from the sample library, it is determined, through statistical analysis, that the model has low recognition accuracy for receipt documents. In this case, targeted training may be performed for the model. Since the weak document type of the model has been determined,

effective samples may be constructed specifically for receipt samples, such as furniture order receipt templates. New corpora may be selected and fused with the furniture order receipt template to generate a large number of new furniture order receipt samples. On this basis, data augmentation may also be performed on the furniture order receipt samples, such as adding
5 crease effects, shadow effects, occlusion effects, etc., to form effective samples. Finally, the samples in the set may be used to optimize the multimodal document recognition model. After optimization, re-evaluation may be performed, and the above process may be iterated until the model meets a business deployment requirement, after which training may be stopped to obtain a multimodal document recognition model with higher recognition accuracy and wider
10 document type coverage.

[0219] In specific implementation, for the difficulty of collecting scarce samples, the above processing may also quickly and efficiently generate related effective samples. It can be understood that in some business scenarios, due to the limited public availability of scarce samples, it is difficult for the model to achieve high recognition accuracy for such documents.
15 In this case, through the above processing method, only one document corresponding to the scarce data is needed, and a large number of effective samples of the same type can be quickly generated from that one document, thereby training the model to achieve high recognition accuracy for scarce data documents as well.

[0220] In summary, by training the multimodal document recognition model in a targeted
20 manner, the robustness of the model can be effectively improved, and rapid and efficient generation of similar samples is supported when the number of reference effective samples is extremely small, effectively overcoming the difficulty of sample collection. Training the model in this way enables the model to cover a wider range of document types, so that the model can be deployed in different business scenarios and meet the usage requirements of
25 those business scenarios.

[0221] Referring to Figure 3, Figure 3 shows a flowchart of a model training method according to an embodiment of the present disclosure, which specifically includes the following steps S302 to S308.

[0222] In step S302, a document file and a text corpus are acquired, multiple document
30 images and a document base corpus are determined according to the document file, and a text base corpus is determined according to the text corpus.

[0223] In step S304, element style information and page layout information corresponding to each document image are extracted, a document template is constructed according to the element style information and the page layout information, and a target base corpus is determined according to the document base corpus and the text base corpus.

5 [0224] In step S306, the document template and the target base corpus are fused by an element renderer and a layout manager according to a preset writing mode and a font resource library, to obtain a document recognition sample and a document recognition label.

[0225] In step S308, a multimodal document recognition model is trained based on the document recognition sample and the document recognition label, until a target multimodal
10 document recognition model meeting a training stop condition is obtained.

[0226] The model training method according to this embodiment corresponds to the sample construction method described above, where details not described in detail may be found in the same or corresponding description in the above embodiments, which will not be repeated here in this embodiment.

15 [0227] Corresponding to the above method embodiments, a sample construction device is further provided according to the embodiments of the present disclosure. Figure 4 shows a structural diagram of a sample construction device according to an embodiment of the present disclosure. As shown in Figure 4, the device includes an acquisition module 402, an extraction module 404, a fusion module 406 and a storage module 408. The acquisition
20 module 402 is configured to acquire a document file and a text corpus, determine multiple document images and a document base corpus according to the document file, and determine a text base corpus according to the text corpus. The extraction module 404 is configured to extract element style information and page layout information corresponding to each of the document images, construct a document template according to the element style information
25 and the page layout information, and determine a target base corpus according to the document base corpus and the text base corpus. The fusion module 406 is configured to fuse the document template and the target base corpus by an element renderer and a layout manager according to a preset writing mode and a font resource library, to obtain a document recognition sample and a document recognition label. The storage module 408 is configured
30 to construct a document recognition sample pair based on the document recognition sample and the document recognition label, and store the document recognition sample pair into a

sample library.

[0228] In an embodiment, the acquiring a document file and a text corpus, and determining multiple document images and a document base corpus according to the document file, includes:

5 acquiring an original dataset, classifying original data in the original dataset, and determining a document file and a text corpus according to a classification result;

 converting the document file into a preset document format, and converting the document file in the preset document format into multiple initial document images, where each initial document image corresponds to each document page in the document file;

10 performing image detection on the multiple initial document images, and selecting, according to a detection result, initial document images that meet a sample construction condition as the multiple document images; and

 parsing the multiple document images, determining a document paragraph text and a document paragraph category corresponding to the document paragraph text according to a parsing result, and forming the document base corpus from the document paragraph text and the document paragraph category.

[0229] In an embodiment, the determining a text base corpus according to the text corpus includes:

 splitting the text corpus into paragraphs to obtain multiple candidate text paragraphs;

20 classifying the multiple candidate text paragraphs, and determining a text paragraph text and a text paragraph category corresponding to the text paragraph text according to a classification result;

 forming the text base corpus based on the text paragraph text and the text paragraph category.

25 **[0230]** In an embodiment, the determining the element style information and the page layout information corresponding to each of the multiple document images includes:

 extracting, according to a preset element style extraction rule, sub-element style information corresponding to each of the multiple element style types from the document image, and forming the element style information from the sub-element style information

corresponding to each of the multiple element style types;

determining a document type corresponding to the document image, and determining inferred layout information corresponding to the document image according to the document type; and

- 5 verifying the document image according to the inferred layout information, and extracting, if verification passes, page hierarchy structure information and region relationship information from the document image to form the page layout information.

[0231] In an embodiment, the constructing a document template according to the element style information and the page layout information includes:

- 10 storing the element style information into an element style library, and storing the page layout information into a page layout library;

selecting element style information to be combined from the element style library according to a preset template construction rule, and selecting page layout information to be combined from the page layout library;

- 15 constructing a combined document template according to the element style information to be combined and the page layout information to be combined, and determining a preset reference document template;

- performing enhancement processing on the reference document template and/or the combined document template, and determining a document template according to an enhancement processing result.
- 20

[0232] In an embodiment, the determining a target base corpus according to the document base corpus and the text base corpus includes:

constructing a candidate base corpus from the document base corpus and the text base corpus;

- 25 performing deduplication on the candidate base corpus, and dividing the deduplicated candidate base corpus into specific document corpus and non-specific document corpus;

performing similar corpus expansion on the specific document corpus, and/or performing identical corpus enhancement on the specific document corpus, and determining a target specific document corpus according to a processing result; and

using the target specific document corpus and the non-specific document corpus as the target base corpus.

[0233] In an embodiment, the fusing the document template and the target base corpus by an element renderer and a layout manager according to a preset writing mode and a font resource library, to obtain a document recognition sample and a document recognition label, includes:

rendering the target base corpus by the element renderer by fusing the template style information in the document template according to the preset writing mode and the font resource library, to obtain a paragraph element;

rendering the paragraph elements into the document template by the layout manager according to template layout information in the document template, to obtain a document recognition sample; and

determining a document recognition label corresponding to the document recognition sample based on the template style information and the template layout information.

[0234] In an embodiment, the element renderer includes at least one of the following algorithms:

a rollback algorithm, configured to roll back a local paragraph element or a global paragraph element;

a truncation algorithm, configured to decide and manage, if a global paragraph element cannot be rendered, a state of a paragraph element to be rendered;

a pre-rendering algorithm, configured to perform pre-rendering processing on a paragraph element to be rendered;

a writing mode control algorithm, configured to determine a sub-writing mode corresponding to a paragraph element to be rendered; and

a first information extraction algorithm, configured to extract an element to be rendered according to a preset information extraction granularity.

[0235] In an embodiment, the layout manager includes at least one of the following algorithms:

a writing mode algorithm, configured to manage a writing result of a paragraph element

according to the writing mode;

a reading layout algorithm, configured to manage a layout order of a paragraph element;

a region occupation algorithm, configured to manage a region occupation state in the document template according to region occupation information of the paragraph element;

5 a conflict rollback algorithm, configured to roll back to a state before a rendering node in case of a rendering conflict;

a cutting algorithm, configured to cut a paragraph element that does not meet a display requirement; and

10 a second information extraction algorithm, configured to extract a rendered element, page information and region information according to a preset information extraction granularity.

[0236] In an embodiment, the constructing a document recognition sample pair based on the document recognition sample and the document recognition label includes:

15 keeping element positions in the document recognition sample unchanged, and adding position-invariant enhancement information to the document recognition sample to obtain a first enhanced document recognition sample; and/or,

adjusting element positions in the document recognition sample, and adding position-variant enhancement information to the document recognition sample to obtain a second enhanced document recognition sample;

20 constructing a first document recognition sample pair based on the first enhanced document recognition sample and the document recognition label, and constructing a second document recognition sample pair based on the second enhanced document recognition sample and the document recognition label; and

25 using the first document recognition sample pair and the second document recognition sample pair as the document recognition sample pair.

[0237] In an embodiment, after the step of storing the document recognition sample pair into a sample library, the method further includes:

extracting a target document recognition sample pair from the sample library;

inputting a target document recognition sample in the target document recognition sample pair into a multimodal document recognition model for processing, to obtain predicted document recognition information; and

5 training the multimodal document recognition model according to the predicted document recognition information and a target document recognition label in the target document recognition sample pair, until a target multimodal document recognition model meeting a training stop condition is obtained.

[0238] In an embodiment, the method further includes:

10 determining, according to a training result of the target multimodal document recognition model, evaluation information corresponding to each document recognition sample pair in the sample library; and

constructing a template constraint condition according to the evaluation information, where the template constraint condition is used to constrain a generated document template in a next sample construction cycle.

15 **[0239]** In an embodiment, the method further includes:

determining, according to the training result of the target multimodal document recognition model, recognition error statistics information corresponding to the target multimodal document recognition model;

20 determining a reference effective sample corresponding to the recognition error statistics information, and constructing an effective sample set according to the reference effective sample; and

optimizing the target multimodal document recognition model using the effective sample set, until a business multimodal document recognition model meeting a deployment condition is obtained.

25 **[0240]** The above is a schematic solution of the sample construction device of this embodiment. It should be noted that the technical solution of this sample construction device belongs to the same concept as the technical solution of the sample construction method described above, and details not described in detail in the technical solution of the sample construction device may be found in the description of the technical solution of the sample construction method.

30

[0241] Corresponding to the above method embodiments, a model training device is further provided according to the embodiment of the present disclosure. Figure 5 shows a structural diagram of a model training device according to an embodiment of the present disclosure. As shown in Figure 5, the device includes a corpus acquisition module 502, an information
5 extraction module 504, a corpus fusion module 506 and a model training module 508. The corpus acquisition module 502 is configured to acquire a document file and a text corpus, determine multiple document images and a document base corpus according to the document file, and determine a text base corpus according to the text corpus. The information extraction
10 module 504 is configured to extract element style information and page layout information corresponding to each of the document images, construct a document template according to the element style information and the page layout information, and determine a target base corpus according to the document base corpus and the text base corpus. The corpus fusion
15 module 506 is configured to fuse the document template and the target base corpus by an element renderer and a layout manager according to a preset writing mode and a font resource library, to obtain a document recognition sample and a document recognition label. The model training module 508 is configured to train a multimodal document recognition model based on the document recognition sample and the document recognition label, until a target multimodal document recognition model meeting a training stop condition is obtained.

[0242] The above is a schematic solution of the model training device of this embodiment.
20 It should be noted that the technical solution of this model training device belongs to the same concept as the technical solution of the model training method described above, and details not described in detail in the technical solution of the model training device can be found in the description of the technical solution of the model training method.

[0243] Figure 6 illustrates a structural block diagram of a computing device 600 according
25 to an embodiment of the present disclosure. The components of the computing device 600 include, but are not limited to, a memory 610 and a processor 620. The processor 620 is connected to the memory 610 via a bus 630, and a database 650 is used to store data.

[0244] The computing device 600 further includes an access device 640, the access device
540 enables the computing device 600 to communicate via one or more networks 660.
30 Examples of these networks include a public switched telephone network (PSTN), a local area network (LAN), a wide area network (WAN), a personal area network (PAN), or a

combination of communication networks such as the Internet. The access device 640 may include one or more of any types of wired or wireless network interfaces (e.g., a network interface controller (NIC), such as an IEEE802.11 Wireless Local Area Network (WLAN) wireless interface, Worldwide Interoperability for Microwave Access (Wi-MAX) interface, Ethernet interface, Universal Serial Bus (USB) interface, cellular network interface, Bluetooth interface, Near Field Communication (NFC).

[0245] In an embodiment of the present disclosure, the above components of the computing device 600, as well as other components not shown in Figure 6, may also be connected to each other, such as via a bus. It should be understood that the block diagram of the structure of the computing device shown in Figure 6 is for exemplary purposes only and is not a limitation on the scope of the present disclosure. Those skilled in the art may add or replace other components as desired.

[0246] The computing device 600 may be any type of stationary or mobile computing device, including a mobile computer or a mobile computing device (e.g., a tablet computer, a personal digital assistant, a laptop computer, a notebook computer, a netbook, etc.), a cell phone (e.g., a smartphone), a wearable computing device (e.g., a smartwatch, smartglasses, etc.), or other types of mobile devices, or a stationary computing device such as desktop computer or a personal computer (PC). The computing device 600 may also be a mobile or stationary server.

[0247] The processor 620 is used to execute computer executable instructions, which, when executed by the processor, perform the steps of the above-described document processing method.

[0248] A schematic solution of the computing device of the embodiment is described above. It should be noted that the technical solution of the computing device belong to the same concept as the technical solution of the sample construction method or the model training method described above, and the details not described in detail in the technical solution of the document processing device may be found in the description of the technical solution of the sample construction method or the model training method described above.

[0249] A computer-readable storage medium is further provided according to an embodiment of the present disclosure, which stores computer-executable instructions, the computer-executable instructions, when executed by a processor, perform the steps of the

above-described sample construction method or the model training method.

[0250] A schematic solution of the computer-readable storage medium of the embodiment is described above. It should be noted that the technical solution of the computer-readable storage medium belong to the same concept as the technical solution of the sample construction method or the model training method described above, and the details not described in detail in the technical solution of the storage medium may be referred to the description of the technical solution of the sample construction method or the model training method described above.

[0251] A computer program product is further provided according to an embodiment of the present disclosure, which includes computer programs or instructions, the computer programs or instructions, when executed by a processor, perform the steps of the above-described sample construction method or the model training method.

[0252] A schematic solution of the computer program product of the embodiment is described above. It should be noted that the technical solution of the computer program product belong to the same concept as the technical solution of the sample construction method or the model training method described above, and the details not described in detail in the technical solution of the computer program product may be referred to the description of the technical solution of the sample construction method or the model training method described above.

[0253] Preferred embodiments of the present disclosure are described above. Other embodiments are within the scope of the claims. In some cases, the actions or steps described in the claims may be performed in a different order than in the embodiments and still achieve the desired results. In addition, the processes depicted in the drawings do not necessarily require the particular order or consecutive order shown to achieve the desired results. In some embodiments, multi-tasking and parallel processing is also possible or may be advantageous.

[0254] The computer instruction includes computer program codes, the computer program code may be in the form of source code, in the form of object code, in the form of an executable file or in some intermediate form or the like. The computer readable medium may include: any entity or device capable of carrying the computer program code, a recording medium, a USB disk, a removable hard disk, a diskette, an optical disk, a computer memory, a Read-Only Memory (ROM), a Random Access Memory (RAM), an electric carrier signal, a

telecommunications signal, and software distribution medium, etc. It should be noted that the content of the computer-readable medium may be increased or decreased according to the requirements of the patent practice, for example, in some regions, according to the patent practice, the computer-readable medium does not include carrier signals and telecommunication signals.

[0255] It should be noted that, for each of the foregoing method embodiments, for the sake of simplicity of description, they are all expressed as a series of combinations of actions, but those skilled in the art should know that the embodiments of the present disclosure are not limited by the order of the described actions, because according to the embodiments of the present disclosure, certain steps may be performed in other orders or simultaneously. Secondly, those skilled in the art should also be aware that the embodiments described in the disclosure are preferred embodiments, and the actions and modules involved are not necessarily necessary for the embodiments of the present disclosure.

[0256] In the above-described embodiments, the description of each embodiment has its own focus, and the part not described in detail in a certain embodiment can be referred to the relevant description of other embodiments.

[0257] The above preferred embodiments of the present disclosure are intended only to aid in the description of the present disclosure. The preferred embodiments are not an exhaustive recitation of all details, nor do they limit the invention to the specific embodiments described only. Apparently, many modifications and variations may be made in accordance with the contents of the embodiments of the present disclosure. These embodiments are selected and specifically described in the present disclosure to better explain the principles and practical applications of the embodiments of the present disclosure so that those skilled in the art to which they belong will be able to well understand and utilize the present disclosure.

CLAIMS

Claim 1. A sample construction method, comprising:

acquiring a document file and a text corpus, determining a plurality of document images
5 and a document base corpus according to the document file, and determining a text base
corpus according to the text corpus;

extracting element style information and page layout information corresponding to each
of the document images, constructing a document template according to the element style
information and the page layout information, and determining a target base corpus according
10 to the document base corpus and the text base corpus;

fusing the document template and the target base corpus by an element renderer and a
layout manager according to a preset writing mode and a font resource library, to obtain a
document recognition sample and a document recognition label; and

constructing a document recognition sample pair based on the document recognition
15 sample and the document recognition label, and storing the document recognition sample pair
into a sample library.

Claim 2. The sample construction method according to claim 1, wherein the acquiring a
document file and a text corpus, and determining a plurality of document images and a
document base corpus according to the document file, comprises:

20 acquiring an original dataset, classifying original data in the original dataset, and
determining a document file and a text corpus according to a classification result;

converting the document file into a preset document format, and converting the
document file in the preset document format into a plurality of initial document images,
wherein each of the initial document images corresponds to a document page in the document
25 file;

performing image detection on the plurality of initial document images, and selecting,
according to a detection result, initial document images that meet a sample construction
condition as the plurality of document images; and

parsing the plurality of document images, determining a document paragraph text and a

document paragraph category corresponding to the document paragraph text according to a parsing result, and forming the document base corpus from the document paragraph text and the document paragraph category.

Claim 3. The sample construction method according to claim 1, wherein the determining a
5 text base corpus according to the text corpus comprises:

splitting the text corpus into paragraphs to obtain a plurality of candidate text paragraphs;

classifying the plurality of candidate text paragraphs, and determining a text paragraph text and a text paragraph category corresponding to the text paragraph text according to a
10 classification result; and

forming the text base corpus based on the text paragraph text and the text paragraph category.

Claim 4. The sample construction method according to claim 1, wherein the determining the element style information and the page layout information corresponding to each of the
15 plurality of document images comprises:

extracting, according to a preset element style extraction rule, sub-element style information corresponding to each of a plurality of element style types from the document image, and forming the element style information from the sub-element style information corresponding to each of the plurality of element style types;

20 determining a document type corresponding to the document image, and determining inferred layout information corresponding to the document image according to the document type; and

verifying the document image according to the inferred layout information, and extracting, if verification passes, page hierarchy structure information and region relationship
25 information from the document image to form the page layout information.

Claim 5. The sample construction method according to claim 1, wherein the constructing a document template according to the element style information and the page layout information comprises:

storing the element style information into an element style library, and storing the page

layout information into a page layout library;

selecting element style information to be combined from the element style library according to a preset template construction rule, and selecting page layout information to be combined from the page layout library;

5 constructing a combined document template according to the element style information to be combined and the page layout information to be combined, and determining a preset reference document template; and

performing enhancement processing on the reference document template and/or the combined document template, and determining a document template according to an
10 enhancement processing result.

Claim 6. The sample construction method according to claim 1, wherein the determining a target base corpus according to the document base corpus and the text base corpus comprises:

constructing a candidate base corpus from the document base corpus and the text base corpus;

15 performing deduplication on the candidate base corpus, and dividing the deduplicated candidate base corpus into specific document corpus and non-specific document corpus;

performing similar corpus expansion on the specific document corpus, and/or performing identical corpus enhancement processing on the specific document corpus, and determining a target specific document corpus according to a processing result;

20 using the target specific document corpus and the non-specific document corpus as the target base corpus.

Claim 7. The sample construction method according to claim 1, wherein the fusing the document template and the target base corpus by an element renderer and a layout manager according to a preset writing mode and a font resource library, to obtain a document
25 recognition sample and a document recognition label, comprises:

rendering the target base corpus by the element renderer by fusing template style information in the document template according to the preset writing mode and the font resource library, to obtain a paragraph element;

rendering the paragraph elements into the document template by the layout manager

according to template layout information in the document template, to obtain a document recognition sample; and

determining a document recognition label corresponding to the document recognition sample based on the template style information and the template layout information.

5 Claim 8. The sample construction method according to claim 7, wherein the element renderer comprises at least one of the following algorithms:

a rollback algorithm, configured to roll back a local paragraph element or a global paragraph element;

10 a truncation algorithm, configured to decide and manage, if a global paragraph element cannot be rendered, a state of a paragraph element to be rendered;

a pre-rendering algorithm, configured to perform pre-rendering processing on a paragraph element to be rendered;

a writing mode control algorithm, configured to determine a sub-writing mode corresponding to a paragraph element to be rendered; and

15 a first information extraction algorithm, configured to extract an element to be rendered according to a preset information extraction granularity.

Claim 9. The sample construction method according to claim 7, wherein the layout manager comprises at least one of the following algorithms:

20 a writing mode algorithm, configured to manage a writing result of a paragraph element according to the writing mode;

a reading layout algorithm, configured to manage a layout order of a paragraph element;

a region occupation algorithm, configured to manage a region occupation state in the document template according to region occupation information of the paragraph element;

25 a conflict rollback algorithm, configured to roll back to a state before a rendering node in case of a rendering conflict;

a cutting algorithm, configured to cut a paragraph element that does not meet a display requirement; and

a second information extraction algorithm, configured to extract a rendered element, page information and region information according to a preset information extraction

granularity.

Claim 10. The sample construction method according to any one of claims 1 to 9, wherein the constructing a document recognition sample pair based on the document recognition sample and the document recognition label comprises:

5 keeping element positions in the document recognition sample unchanged, and adding position-invariant enhancement information to the document recognition sample to obtain a first enhanced document recognition sample; and/or,

 adjusting element positions in the document recognition sample, and adding position-variant enhancement information to the document recognition sample to obtain a
10 second enhanced document recognition sample;

 constructing a first document recognition sample pair based on the first enhanced document recognition sample and the document recognition label, and constructing a second document recognition sample pair based on the second enhanced document recognition sample and the document recognition label; and

15 using the first document recognition sample pair and the second document recognition sample pair as the document recognition sample pair.

Claim 11. The sample construction method according to any one of claims 1 to 9, wherein after the step of storing the document recognition sample pair into a sample library, the sample construction method further comprises:

20 extracting a target document recognition sample pair from the sample library;

 inputting a target document recognition sample in the target document recognition sample pair into a multimodal document recognition model for processing, to obtain predicted document recognition information; and

 training the multimodal document recognition model according to the predicted
25 document recognition information and a target document recognition label in the target document recognition sample pair, until a target multimodal document recognition model meeting a training stop condition is obtained.

Claim 12. The sample construction method according to claim 11, further comprising:

 determining, according to a training result of the target multimodal document recognition

model, evaluation information corresponding to each document recognition sample pair in the sample library; and

constructing a template constraint condition according to the evaluation information, wherein the template constraint condition is used to constrain a generated document template in a next sample construction cycle.

Claim 13. The sample construction method according to claim 11, further comprising:

determining, according to the training result of the target multimodal document recognition model, recognition error statistics information corresponding to the target multimodal document recognition model;

determining a reference effective sample corresponding to the recognition error statistics information, and constructing an effective sample set according to the reference effective sample; and

optimizing the target multimodal document recognition model by using the effective sample set, until a business multimodal document recognition model meeting a deployment condition is obtained.

Claim 14. A model training method, comprising:

acquiring a document file and a text corpus, determining a plurality of document images and a document base corpus according to the document file, and determining a text base corpus according to the text corpus;

extracting element style information and page layout information corresponding to each of the document images, constructing a document template according to the element style information and the page layout information, and determining a target base corpus according to the document base corpus and the text base corpus;

fusing the document template and the target base corpus by an element renderer and a layout manager according to a preset writing mode and a font resource library, to obtain a document recognition sample and a document recognition label; and

training a multimodal document recognition model based on the document recognition sample and the document recognition label, until a target multimodal document recognition model meeting a training stop condition is obtained.

Claim 15. A sample construction device, comprising:

an acquisition module, configured to acquire a document file and a text corpus, determine a plurality of document images and a document base corpus according to the document file, and determine a text base corpus according to the text corpus;

5 an extraction module, configured to extract element style information and page layout information corresponding to each of the document images, construct a document template according to the element style information and the page layout information, and determine a target base corpus according to the document base corpus and the text base corpus;

10 a fusion module, configured to fuse the document template and the target base corpus by an element renderer and a layout manager according to a preset writing mode and a font resource library, to obtain a document recognition sample and a document recognition label; and

15 a storage module, configured to construct a document recognition sample pair based on the document recognition sample and the document recognition label, and store the document recognition sample pair into a sample library.

Claim 16. A model training device, comprising:

a corpus acquisition module, configured to acquire a document file and a text corpus, determine a plurality of document images and a document base corpus according to the document file, and determine a text base corpus according to the text corpus;

20 an information extraction module, configured to extract element style information and page layout information corresponding to each of the document images, construct a document template according to the element style information and the page layout information, and determine a target base corpus according to the document base corpus and the text base corpus;

25 a corpus fusion module, configured to fuse the document template and the target base corpus by an element renderer and a layout manager according to a preset writing mode and a font resource library, to obtain a document recognition sample and a document recognition label; and

a model training module, configured to train a multimodal document recognition model

based on the document recognition sample and the document recognition label, until a target multimodal document recognition model meeting a training stop condition is obtained.

Claim 17. A computing device, comprising:

a memory, configured to store computer executable instructions; and

5 a processor, configured to execute the computer executable instructions, wherein the computer executable instructions, when executed by the processor, perform the steps of the method according to any one of claims 1 to 14.

Claim 18. A computer-readable storage medium, comprising computer executable instructions stored thereon, wherein the computer executable instructions, when executed by a processor,

10 perform the steps of the method according to any one of claims 1 to 14.

Claim 19. A computer program product, comprising computer programs or instructions, wherein the computer programs or instructions, when executed by a processor, perform the steps of the method according to any one of claims 1 to 14.