

Package ‘PaddleOCR’

August 30, 2026

Title R Client for the 'PaddleOCR' Cloud API

Version 0.2.1

Description An R client for the 'PaddleOCR' cloud service API

<https://www.paddleocr.ai/latest/en/version3.x/inference_deployment/serving/paddleocr_official_api/overview.html>.

Submit images, PDFs, or URLs for OCR processing using models like 'PaddleOCR-VL-1.6'.

Supports job submission, polling, result retrieval, automatic image download, and streaming PDF-to-markdown conversion with batch processing.

License MIT + file LICENSE

URL <https://github.com/xiaoluolorn/PaddleOCR>

BugReports <https://github.com/xiaoluolorn/PaddleOCR/issues>

Depends R (>= 4.0.0)

Imports httr (>= 1.4.0), jsonlite (>= 1.7.0), tools, utils

Suggests pdftools (>= 3.0.0), testthat (>= 3.0.0), withr

Config/testthat/edition 3

Encoding UTF-8

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Chaoyang Luo [aut, cre]

Maintainer Chaoyang Luo <xiaoluolorn@gmail.com>

Repository CRAN

Date/Publication 2026-08-30 09:50:02 UTC

Contents

batch_pdf_to_markdown_with_paddle	2
image_to_markdown	3
paddle_ocr	4
pdf_page_to_image	6

pdf_to_images	7
pdf_to_markdown_with_paddle	7
poll_paddle_job	9
process_paddle_jsonl_result	10
submit_paddle_job	11
Index	13

batch_pdf_to_markdown_with_paddle
<i>Batch convert PDFs to Markdown via PaddleOCR</i>

Description

Processes all PDF files in a directory, converting each to Markdown using [pdf_to_markdown_with_paddle](#). Requires the **pdftools** package.

Usage

```
batch_pdf_to_markdown_with_paddle(  
    pdf_dir = ".",  
    output_root = file.path(pdf_dir, "paddle_output"),  
    dpi = 300,  
    batch_trigger = 3,  
    workers = 1,  
    resume = TRUE,  
    token = "",  
    job_url = "",  
    model = "",  
    poll_interval = 5,  
    max_wait_seconds = 1800,  
    timeout = 600  
)
```

Arguments

pdf_dir	Directory containing PDF files (default: current directory).
output_root	Root directory for output (default: "<pdf_dir>/paddle_output").
dpi	Image resolution for rendering (default: 300).
batch_trigger	Number of pages to render before starting OCR (default: 3).
workers	Maximum number of OCR jobs submitted concurrently.
resume	Logical; reuse completed work from interrupted runs.
token	PaddleOCR API token.
job_url	PaddleOCR API endpoint.
model	Model name.

poll_interval Polling interval in seconds.
max_wait_seconds Maximum wait time per job.
timeout HTTP timeout.

Value

A named list of results (one per PDF). Failed conversions return a list with an error element.

Examples

```
## Not run:  
results <- batch_pdf_to_markdown_with_paddle(  
  pdf_dir = "papers",  
  output_root = "papers/paddle_output"  
)  
  
## End(Not run)
```

image_to_markdown	<i>OCR a single image file via PaddleOCR</i>
-------------------	--

Description

Submits an image for OCR, polls for completion, and saves Markdown output.

Usage

```
image_to_markdown(  
  image_path,  
  output_dir,  
  page_index,  
  token = "",  
  job_url = "",  
  model = "",  
  use_doc_orientation_classify = FALSE,  
  use_doc_unwarping = FALSE,  
  use_chart_recognition = FALSE,  
  poll_interval = 5,  
  max_wait_seconds = 1800,  
  timeout = 600  
)
```

Arguments

image_path	Path to the image file.
output_dir	Directory for OCR output.
page_index	Page index for naming output files (1-indexed).
token	PaddleOCR API token.
job_url	PaddleOCR API endpoint.
model	Model name.
use_doc_orientation_classify	Logical; enable orientation classification.
use_doc_unwarping	Logical; enable document unwarping.
use_chart_recognition	Logical; enable chart recognition.
poll_interval	Polling interval in seconds.
max_wait_seconds	Maximum wait time.
timeout	HTTP timeout.

Value

A list with markdown_paths, markdown_texts, and job_id.

Examples

```
## Not run:
result <- image_to_markdown("page_1.png", output_dir = "output", page_index = 1)

## End(Not run)
```

paddle_ocr

Perform OCR on an image, PDF, or URL using PaddleOCR

Description

This is the main entry point for the PaddleOCR cloud API. Submit a local file (image or document) or a URL, and the function will handle job submission, polling, result retrieval, and file saving automatically.

Usage

```
paddle_ocr(
  file_path,
  output_dir = "output",
  token = "",
  job_url = "",
  model = "",
  use_doc_orientation_classify = FALSE,
  use_doc_unwarping = FALSE,
  use_chart_recognition = FALSE,
  poll_interval = 5,
  max_wait_seconds = 3600,
  timeout = 600
)
```

Arguments

<code>file_path</code>	A local file path or a URL (<code>http://</code> or <code>https://</code>).
<code>output_dir</code>	Directory to save output Markdown and images (default: "output").
<code>token</code>	PaddleOCR API bearer token. If empty, reads from the <code>PADDLE_OCR_TOKEN</code> environment variable.
<code>job_url</code>	PaddleOCR API endpoint. Defaults to the official cloud URL.
<code>model</code>	Model name to use. Defaults to "PaddleOCR-VL-1.6".
<code>use_doc_orientation_classify</code>	Logical; enable document orientation classification.
<code>use_doc_unwarping</code>	Logical; enable document unwarping.
<code>use_chart_recognition</code>	Logical; enable chart/table recognition.
<code>poll_interval</code>	Seconds between status checks (default: 5).
<code>max_wait_seconds</code>	Maximum wait time for job completion (default: 3600).
<code>timeout</code>	HTTP request timeout in seconds (default: 600).

Value

A list with elements:

- file_path** Input file path or URL.
- job_id** The submitted job ID.
- output_dir** Output directory path.
- markdown_files** Paths to saved Markdown files.
- page_count** Number of pages processed.

Examples

```
## Not run:
# OCR a local image
result <- paddle_ocr("document.png")

# OCR a URL
result <- paddle_ocr("https://example.com/document.jpg")

# With custom options
result <- paddle_ocr(
  file_path = "table.png",
  use_chart_recognition = TRUE,
  output_dir = "my_output"
)

## End(Not run)
```

pdf_page_to_image	<i>Convert a single PDF page to an image</i>
-------------------	--

Description

Renders one page of a PDF to a PNG file. Uses a temporary directory internally to avoid **pdftools** filename template issues.

Usage

```
pdf_page_to_image(pdf_path, page_index, image_dir, dpi = 300, overwrite = TRUE)
```

Arguments

pdf_path	Path to the PDF file.
page_index	Page number to render (1-indexed).
image_dir	Directory to save the PNG image.
dpi	Resolution in dots per inch (default: 300).
overwrite	Logical; overwrite an existing rendered page. Set to FALSE to reuse it (default: TRUE).

Value

Path to the saved PNG image.

Examples

```
## Not run:
img <- pdf_page_to_image("document.pdf", page_index = 1, image_dir = "pages")

## End(Not run)
```

`pdf_to_images`*Convert all pages of a PDF to images*

Description

Renders every page of a PDF file to PNG images at the specified DPI. Requires the **pdftools** package.

Usage

```
pdf_to_images(pdf_path, image_dir, dpi = 300)
```

Arguments

<code>pdf_path</code>	Path to the PDF file.
<code>image_dir</code>	Directory to save PNG images into.
<code>dpi</code>	Resolution in dots per inch (default: 300).

Value

A character vector of image file paths.

Examples

```
## Not run:  
images <- pdf_to_images("document.pdf", "images", dpi = 300)  
  
## End(Not run)
```

`pdf_to_markdown_with_paddle`*Convert a PDF to Markdown via PaddleOCR*

Description

Renders PDF pages to images one by one, then submits them for OCR using the PaddleOCR cloud API. Pages are processed in streaming batches: once `batch_trigger` pages are rendered, OCR begins while rendering continues.

Usage

```
pdf_to_markdown_with_paddle(
    pdf_path,
    output_dir = NULL,
    combined_markdown = TRUE,
    dpi = 300,
    batch_trigger = 3,
    workers = 1,
    resume = TRUE,
    token = "",
    job_url = "",
    model = "",
    use_doc_orientation_classify = FALSE,
    use_doc_unwarping = FALSE,
    use_chart_recognition = FALSE,
    poll_interval = 5,
    max_wait_seconds = 1800,
    timeout = 600,
    ...
)
```

Arguments

pdf_path	Path to the PDF file.
output_dir	Output directory. If NULL, defaults to "<pdf_name>_paddle_output" next to the PDF.
combined_markdown	Logical; if TRUE (default), combine all page Markdown into a single file.
dpi	Image resolution for rendering (default: 300).
batch_trigger	Number of pages to render before starting OCR (default: 3).
workers	Maximum number of OCR jobs submitted concurrently (default: 1). OCR runs concurrently on the PaddleOCR service; result polling remains local and sequential.
resume	Logical; reuse rendered page images, completed Markdown files, and submitted job IDs from an interrupted run (default: TRUE).
token	PaddleOCR API token.
job_url	PaddleOCR API endpoint.
model	Model name.
use_doc_orientation_classify	Logical; enable orientation classification.
use_doc_unwarping	Logical; enable document unwarping.
use_chart_recognition	Logical; enable chart recognition.
poll_interval	Polling interval in seconds.

max_wait_seconds	Maximum wait time per job.
timeout	HTTP timeout.
...	Ignored (for future compatibility).

Details

Requires the **pdftools** package.

Value

A list with PDF path, image paths, Markdown file paths, combined Markdown path and text, and output directory.

Examples

```
## Not run:
result <- pdf_to_markdown_with_paddle("document.pdf")
cat(result$combined_markdown_text)

## End(Not run)
```

poll_paddle_job	<i>Poll a PaddleOCR job until completion</i>
-----------------	--

Description

Repeatedly queries the job status endpoint until the job reaches "done" or "failed" state, or the maximum wait time is exceeded.

Usage

```
poll_paddle_job(
  job_id,
  token = "",
  job_url = "",
  poll_interval = 5,
  max_wait_seconds = 3600
)
```

Arguments

job_id	The job ID returned by submit_paddle_job .
token	PaddleOCR API bearer token. If empty, reads from the PADDLE_OCR_TOKEN environment variable.
job_url	PaddleOCR API endpoint. Defaults to the official cloud URL.
poll_interval	Seconds between status checks (default: 5).
max_wait_seconds	Maximum total wait time in seconds (default: 3600).

Value

A character string containing the JSONL result URL.

Examples

```
## Not run:
job_id <- submit_paddle_job("document.png", token = "your_token")
jsonl_url <- poll_paddle_job(job_id, token = "your_token")

## End(Not run)
```

process_paddle_jsonl_result

Download and parse a PaddleOCR JSONL result

Description

Fetches the JSONL result from the URL, parses each line, and saves layout-parsed Markdown documents and their associated images.

Usage

```
process_paddle_jsonl_result(jsonl_url, output_dir, starting_doc_index = 0L)
```

Arguments

jsonl_url	URL to the JSONL result file.
output_dir	Directory to save output files.
starting_doc_index	Integer offset for naming output files (default: 0).

Value

A list with elements markdown_files (file paths), markdown_texts (text content), and doc_count (number of documents saved).

Examples

```
## Not run:
result <- process_paddle_jsonl_result(
  jsonl_url = "https://example.com/result.jsonl",
  output_dir = "output"
)

## End(Not run)
```

submit_paddle_job	<i>Submit an OCR job to the PaddleOCR cloud API</i>
-------------------	---

Description

Submits a local file or a URL for OCR processing. The function detects whether the input is a URL or a local file path and uses the appropriate API call method (JSON body vs. multipart upload).

Usage

```
submit_paddle_job(
  file_path,
  token = "",
  job_url = "",
  model = "",
  use_doc_orientation_classify = FALSE,
  use_doc_unwarping = FALSE,
  use_chart_recognition = FALSE,
  timeout = 600
)
```

Arguments

file_path	A local file path or a URL (http:// or https://).
token	PaddleOCR API bearer token. If empty, reads from the PADDLE_OCR_TOKEN environment variable.
job_url	PaddleOCR API endpoint. Defaults to the official cloud URL.
model	Model name to use. Defaults to "PaddleOCR-VL-1.6".
use_doc_orientation_classify	Logical; enable document orientation classification.
use_doc_unwarping	Logical; enable document unwarping.
use_chart_recognition	Logical; enable chart/table recognition.
timeout	HTTP request timeout in seconds (default: 600).

Value

A character string containing the job ID.

Examples

```
## Not run:
# Submit a local image
job_id <- submit_paddle_job(
  file_path = "document.png",
```

```
    token = "your_token_here"
  )

  # Submit a URL
  job_id <- submit_paddle_job(
    file_path = "https://example.com/document.jpg",
    token = "your_token_here"
  )

  ## End(Not run)
```

Index

`batch_pdf_to_markdown_with_paddle`, [2](#)

`image_to_markdown`, [3](#)

`paddle_ocr`, [4](#)

`pdf_page_to_image`, [6](#)

`pdf_to_images`, [7](#)

`pdf_to_markdown_with_paddle`, [2](#), [7](#)

`poll_paddle_job`, [9](#)

`process_paddle_jsonl_result`, [10](#)

`submit_paddle_job`, [9](#), [11](#)