Package 'kuzco'

January 26, 2026

Title Computer Vision with Large Language Models

Version 0.1.0

Description Make computer vision tasks approachable in R by leveraging Large Language Models. Providing fine-tuned prompts, boilerplate functions, and input/output helpers for common computer vision workflows, such as classifying and describing images.
Functions are designed to take images as input and return structured data, helping users build practical applications with minimal code.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 4.1.0)

Imports dplyr, ellmer, graphics, gt, gtExtras, imager, jsonlite, magick, ollamar, rlang, stringr, tidyr, utils

Suggests knitr, quarto, shiny, bslib, rmarkdown, purrr, mirai, tibble, tictoc, httr

VignetteBuilder quarto, knitr

URL https://frankiethull.github.io/kuzco/, https://github.com/frankiethull/kuzco

BugReports https://github.com/frankiethull/kuzco/issues

NeedsCompilation no

Author Frank Hull [aut, cre, cph], Johannes Breuer [ctb] (ORCID: <https://orcid.org/0000-0001-5906-7873>), Jordi Rosell [ctb] (ORCID: <https://orcid.org/0000-0002-4349-1458>)

Maintainer Frank Hull <frankiethull@gmail.com>

Repository CRAN

Date/Publication 2026-01-26 16:50:02 UTC

Contents

`chat_ellmer` *chat ellmer helper (predates ellmer::chat)*

Description

a minimal wrapper function to switch which provider is used for each llm_image* function when ellmer backend is selected, ollamar only supports ollama

Usage

```
chat_ellmer(provider = "ollama")
```

Arguments

`provider`	a provider, such as "ollama", or "claude", or "github"

Value

which ellmer function (provider) to use for kuzco llm_image_* when backend is ellmer

edit_prompt *edit prompt*

Description

edit a listed prompt installed with kuzco

Usage

```
edit_prompt(prompt)
```

Arguments

prompt	a prompt from list_prompts()

Value

a prompt markdown file to edit

Examples

```
## Not run:
edit_prompt("system-prompt-alt-text.md")

## End(Not run)
```

kuzco_app *shiny kuzco app*

Description

a simple wrapper of kuzco to make computer vision for everyone. few-shot via frank hull and shiny assistant (https://gallery.shinyapps.io/assistant/)

Usage

```
kuzco_app()
```

Value

a shiny app instance as a playground for local llms

Examples

```
## Not run:
kuzco_app()

## End(Not run)
```

list_prompts *list prompts*

Description

list prompts installed with kuzco

Usage

```
list_prompts()
```

Value

a list of prompts stored within kuzco

Examples

```
list_prompts()
```

llm_image_alt_text *Image Alt Text using LLMs*

Description

Image Alt Text using LLMs

Usage

```
llm_image_alt_text(
  llm_model = "qwen2.5vl",
  image = system.file("img/test_img.jpg", package = "kuzco"),
  backend = "ellmer",
  additional_prompt = "",
  provider = "ollama",
  language = "English",
  ...
)
```

Arguments

`llm_model`	a local LLM model either pulled from ollama or hosted
`image`	a local image path that has a jpeg, jpg, or png
`backend`	either 'ellmer' or 'ollamar', note that 'ollamar' suggests structured outputs while 'ellmer' enforces structured outputs
`additional_prompt`	text to append to the image prompt
`provider`	for `backend = 'ollamar'`, `provider` is ignored. for `backend = 'ellmer'`, `provider` refers to the `ellmer::chat_*` providers and can be used to switch from "ollama" to other providers such as "perplexity"
`language`	a language to guide the LLM model outputs
`...`	a pass through for other generate args and model args like temperature. set the temperature to 0 for more deterministic output

Value

a df with text

Examples

```
llm_image_alt_text(
 llm_model = "qwen2.5vl",
 image = system.file("img/test_img.jpg", package = "kuzco"),
 backend = 'ellmer',
 additional_prompt = "",
provider = "ollama",
language = "English"
)
```

---

llm_image_classification

Image Classification using LLMs

Description

Image Classification using LLMs

Usage

```
llm_image_classification(
  llm_model = "qwen2.5vl",
  image = system.file("img/test_img.jpg", package = "kuzco"),
  backend = "ellmer",
  additional_prompt = "",
  provider = "ollama",
```

```
  language = "English",
  ...
)
```

Arguments

llm_model	a local LLM model either pulled from ollama or hosted
image	a local image path that has a jpeg, jpg, or png
backend	either 'ollamar' or 'ellmer', note that 'ollamar' suggests structured outputs while 'ellmer' enforces structured outputs
additional_prompt	text to append to the image prompt
provider	for backend = 'ollamar', provider is ignored. for backend = 'ellmer', provider refers to the ellmer::chat_* providers and can be used to switch from "ollama" to other providers such as "perplexity"
language	a language to guide the LLM model outputs
...	a pass through for other generate args and model args like temperature

Value

a df with image_classification, primary_object, secondary_object, image_description, image_colors, image_proba_names, image_proba_values

Examples

```
llm_image_classification(
 llm_model = "qwen2.5vl",
 image = system.file("img/test_img.jpg", package = "kuzco"),
 backend = 'ellmer',
 additional_prompt = "",
provider = "ollama",
language = "English"
)
```

---

llm_image_custom *Customized Vision using LLMs*

Description

Customized Vision using LLMs

Usage

```
llm_image_custom(
  llm_model = "qwen2.5vl",
  image = system.file("img/test_img.jpg", package = "kuzco"),
  backend = "ellmer",
  system_prompt = "You are a terse assistant in computer vision sentiment.",
  image_prompt = "return JSON describing image, do not include json or backticks",
  example_df = NULL,
  provider = "ollama",
  ...
)
```

Arguments

`llm_model`	a local LLM model either pulled from ollama or hosted
`image`	a local image path that has a jpeg, jpg, or png
`backend`	either 'ollamar' or 'ellmer'
`system_prompt`	overarching assistant description, *please note that the LLM should be told to return as JSON while kuzco will handle the conversions to and from JSON*
`image_prompt`	anything you want to really remind the llm about.
`example_df`	an example data.frame to show the llm what you want returned *note this will be converted to JSON for the LLM*.
`provider`	for `backend = 'ollamar'`, `provider` is ignored. for `backend = 'ellmer'`, `provider` refers to the `ellmer::chat_*` providers and can be used to switch from "ollama" to other providers such as "perplexity"
`...`	a pass through for other generate args and model args like temperature

Value

a customized return based on example_df for custom control

Examples

```
llm_image_custom(
 llm_model = "qwen2.5vl",
 image = system.file("img/test_img.jpg", package = "kuzco"),
 backend = "ellmer",
 system_prompt = "You are a terse assistant in computer vision sentiment.",
 image_prompt = "return JSON describing image, do not include json or backticks",
 example_df = NULL,
 provider = "ollama"
)
```

`llm_image_extract_text` *Image OCR for Text Extraction using LLMs*

Description

Image OCR for Text Extraction using LLMs

Usage

```
llm_image_extract_text(
  llm_model = "qwen2.5vl",
  image = system.file("img/text_img.jpg", package = "kuzco"),
  backend = "ellmer",
  additional_prompt = "",
  provider = "ollama",
  language = "English",
  ...
)
```

Arguments

`llm_model`	a local LLM model either pulled from ollama or hosted
`image`	a local image path that has a jpeg, jpg, or png
`backend`	either 'ellmer' or 'ollamar', note that 'ollamar' suggests structured outputs while 'ellmer' enforces structured outputs
`additional_prompt`	text to append to the image prompt
`provider`	for `backend = 'ollamar'`, `provider` is ignored. for `backend = 'ellmer'`, `provider` refers to the `ellmer::chat_*` providers and can be used to switch from "ollama" to other providers such as "perplexity"
`language`	a language to guide the LLM model outputs
`...`	a pass through for other generate args and model args like temperature. set the temperature to 0 for more deterministic output

Value

a df with text and a confidence score

Examples

```
llm_image_extract_text(
 llm_model = "qwen2.5vl",
 image = system.file("img/test_img.jpg", package = "kuzco"),
 backend = 'ellmer',
 additional_prompt = "",
```

```
provider = "ollama",
language = "English"
)
```

`llm_image_recognition` *Image Recognition using LLMs*

Description

Image Recognition using LLMs

Usage

```
llm_image_recognition(
  llm_model = "qwen2.5vl",
  image = system.file("img/test_img.jpg", package = "kuzco"),
  recognize_object = "face",
  backend = "ellmer",
  additional_prompt = "",
  provider = "ollama",
  language = "English",
  ...
)
```

Arguments

`llm_model`	a local LLM model either pulled from ollama or hosted
`image`	a local image path that has a jpeg, jpg, or png
`recognize_object`	an item you want to LLM to look for
`backend`	either 'ollamar' or 'ellmer', note that 'ollamar' suggests structured outputs while 'ellmer' enforces structured outputs
`additional_prompt`	text to append to the image prompt
`provider`	for `backend = 'ollamar'`, `provider` is ignored. for `backend = 'ellmer'`, `provider` refers to the `ellmer::chat_*` providers and can be used to switch from "ollama" to other providers such as "perplexity"
`language`	a language to guide the LLM model outputs
`...`	a pass through for other generate args and model args like temperature. set the temperature to 0 for more deterministic output

Value

a df with object_recognized, object_count, object_description, object_location

Examples

```
llm_image_recognition(
 llm_model = "qwen2.5vl",
 image = system.file("img/test_img.jpg", package = "kuzco"),
recognize_object = "nose",
 backend = 'ellmer',
 additional_prompt = "",
provider = "ollama",
language = "English"
)
```

`llm_image_sentiment` *Image Sentiment using LLMs*

Description

Image Sentiment using LLMs

Usage

```
llm_image_sentiment(
  llm_model = "qwen2.5vl",
  image = system.file("img/test_img.jpg", package = "kuzco"),
  backend = "ellmer",
  additional_prompt = "",
  provider = "ollama",
  language = "English",
  ...
)
```

Arguments

`llm_model`	a local LLM model either pulled from ollama or hosted
`image`	a local image path that has a jpeg, jpg, or png
`backend`	either 'ollamar' or 'ellmer', note that 'ollamar' suggests structured outputs while 'ellmer' enforces structured outputs
`additional_prompt`	text to append to the image prompt
`provider`	for `backend = 'ollamar'`, `provider` is ignored. for `backend = 'ellmer'`, `provider` refers to the `ellmer::chat_*` providers and can be used to switch from "ollama" to other providers such as "perplexity"
`language`	a language to guide the LLM model outputs
`...`	a pass through for other generate args and model args like temperature. set the temperature to 0 for more deterministic output

Value

a df with image_sentiment, image_score, sentiment_description, image_keywords

Examples

```
llm_image_sentiment(
 llm_model = "qwen2.5vl",
 image = system.file("img/test_img.jpg", package = "kuzco"),
 backend = 'ellmer',
 additional_prompt = "",
provider = "ollama",
language = "English"
)
```

`view_image` *View Images quickly and easily*

Description

View Images quickly and easily

Usage

```
view_image(image = system.file("img/test_img.jpg", package = "kuzco"))
```

Arguments

`image`	an image to view

Value

a plot of the image in a Plots pane

Examples

```
view_image(image = system.file("img/test_img.jpg", package = "kuzco"))
```

`view_llm_results` *view llm results as a tidy great table*

Description

view llm results as a tidy great table

Usage

```
view_llm_results(llm_results)
```

Arguments

`llm_results`	results from one of the llm_image_* functions

Value

a great table to view the results neatly

Examples

```
## Not run:
view_llm_results(llm_image_alt_text())

## End(Not run)
```

Index