

Package: kuzco (via r-universe)

October 13, 2026

Title Computer Vision with Large Language Models

Version 0.1.0.9000

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

Roxygen list(markdown = TRUE)

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>

Config/Needs/website brand.yml

Config/pak/sysreqs cmake libfftw3-dev libglpk-dev make libmagick++-dev gsfonts libicu-dev libjpeg-dev libpng-dev libtiff-dev libuv1-dev libxml2-dev libssl-dev libnode-dev libx11-dev

Repository <https://frankiethull.r-universe.dev>

Date/Publication 2026-10-13 03:25:26 UTC

RemoteUrl <https://github.com/frankiethull/kuzco>

RemoteRef HEAD

RemoteSha f27c0c59fbe28ca1cb0a77d1f974896d3493e87c

Contents

chat_ellmer	2
edit_prompt	3
kuzco-tools	3
kuzco_app	5
list_prompts	5
llm_image_alt_text	6
llm_image_classification	7
llm_image_custom	8
llm_image_extract_text	9
llm_image_recognition	10
llm_image_sentiment	11
view_image	13
view_llm_results	13
Index	14

chat_ellmer	<i>chat ellmer helper (predates ellmer::chat)</i>
-------------	---

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	<i>edit prompt</i>
-------------	--------------------

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-tools	<i>ellmer tools for kuzco computer vision functions</i>
-------------	---

Description

Lite-weight wrappers that expose each kuzco computer vision function as an `ellmer::tool()` so they can be registered on a chatbot, e.g.

```
chat <- ellmer::chat_ollama(model = "qwen2.5v1")
chat$register_tool(tool_llm_image_classification())
chat$chat("What is in this image?", ellmer::content_image_file("img/photo.jpg"))
```

Each constructor fixes the model and provider at registration time and exposes only the arguments an LLM should control (image, additional_prompt, language, plus function-specific ones). The tools always use the "ellmer" backend, which enforces structured outputs. Results (single-row data frames) are collapsed to compact JSON text so the model gets a clean, readable result.

Usage

```
tool_llm_image_alt_text(  
    llm_model = "qwen2.5vl",  
    provider = "ollama",  
    name = "kuzco_image_alt_text"  
)  
  
tool_llm_image_classification(  
    llm_model = "qwen2.5vl",  
    provider = "ollama",  
    name = "kuzco_image_classification"  
)  
  
tool_llm_image_custom(  
    llm_model = "qwen2.5vl",  
    provider = "ollama",  
    name = "kuzco_image_custom"  
)  
  
tool_llm_image_extract_text(  
    llm_model = "qwen2.5vl",  
    provider = "ollama",  
    name = "kuzco_image_extract_text"  
)  
  
tool_llm_image_recognition(  
    llm_model = "qwen2.5vl",  
    provider = "ollama",  
    name = "kuzco_image_recognition"  
)  
  
tool_llm_image_sentiment(  
    llm_model = "qwen2.5vl",  
    provider = "ollama",  
    name = "kuzco_image_sentiment"  
)
```

Arguments

llm_model	a local LLM model either pulled from ollama or hosted
provider	for backend = "ellmer", provider refers to the ellmer::chat_* providers, e.g. "ollama", "openai", "claude"
name	tool name registered with the chatbot; defaults are "kuzco_image_<task>"

Functions

- tool_llm_image_alt_text(): Tool: generate alt text for an image
- tool_llm_image_classification(): Tool: classify an image

- `tool_llm_image_custom()`: Tool: run a custom image prompt
- `tool_llm_image_extract_text()`: Tool: extract text from an image (OCR)
- `tool_llm_image_recognition()`: Tool: recognize objects in an image
- `tool_llm_image_sentiment()`: Tool: analyze image sentiment

`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	<i>Image Alt Text using LLMs</i>
--------------------	----------------------------------

Description

Image Alt Text using LLMs

Usage

```
llm_image_alt_text(  
  llm_model = "qwen2.5v1",  
  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 = "qwen2.5v1",  
  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 = "qwen2.5v1",
  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 = "qwen2.5v1",  
  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 = "qwen2.5v1",  
  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, <i>please note that the LLM should be told to return as JSON while kuzco will handle the conversions to and from JSON</i>
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 <i>note this will be converted to JSON for the LLM.</i>
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 = "qwen2.5v1",
  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 = "qwen2.5v1",
  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 = "qwen2.5v1",
  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 = "qwen2.5v1",
  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 = "qwen2.5v1",  
  image = system.file("img/test_img.jpg", package = "kuzco"),  
  recognize_object = "nose",  
  backend = 'ellmer',  
  additional_prompt = "",  
  provider = "ollama",  
  language = "English"  
)
```

llm_image_sentiment	<i>Image Sentiment using LLMs</i>
---------------------	-----------------------------------

Description

Image Sentiment using LLMs

Usage

```
llm_image_sentiment(
  llm_model = "qwen2.5v1",
  image = system.file("img/test_img.jpg", package = "kuzco"),
  backend = "ellmer",
  additional_prompt = "",
  provider = "ollama",
  language = "English",
  ...
)
```

Arguments

<code>llm_model</code>	a local LLM model either pulled from ollama or hosted
<code>image</code>	a local image path that has a jpeg, jpg, or png
<code>backend</code>	either 'ollamar' or 'ellmer', note that 'ollamar' suggests structured outputs while 'ellmer' enforces structured outputs
<code>additional_prompt</code>	text to append to the image prompt
<code>provider</code>	for backend = 'ollamar', provider is ignored. for backend = 'ellmer', provider refers to the <code>ellmer::chat_*</code> providers and can be used to switch from "ollama" to other providers such as "perplexity"
<code>language</code>	a language to guide the LLM model outputs
<code>...</code>	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 = "qwen2.5v1",
  image = system.file("img/test_img.jpg", package = "kuzco"),
  backend = 'ellmer',
  additional_prompt = "",
  provider = "ollama",
  language = "English"
)
```

view_image	<i>View Images quickly and easily</i>
------------	---------------------------------------

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	<i>view llm results as a tidy great table</i>
------------------	---

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

`chat_ellmer`, [2](#)

`edit_prompt`, [3](#)

`ellmer::tool()`, [3](#)

`kuzco-tools`, [3](#)

`kuzco_app`, [5](#)

`list_prompts`, [5](#)

`llm_image_alt_text`, [6](#)

`llm_image_classification`, [7](#)

`llm_image_custom`, [8](#)

`llm_image_extract_text`, [9](#)

`llm_image_recognition`, [10](#)

`llm_image_sentiment`, [11](#)

`tool_llm_image_alt_text (kuzco-tools)`, [3](#)

`tool_llm_image_classification`
 (`kuzco-tools`), [3](#)

`tool_llm_image_custom (kuzco-tools)`, [3](#)

`tool_llm_image_extract_text`
 (`kuzco-tools`), [3](#)

`tool_llm_image_recognition`
 (`kuzco-tools`), [3](#)

`tool_llm_image_sentiment (kuzco-tools)`,
 [3](#)

`view_image`, [13](#)

`view_llm_results`, [13](#)