

# Package ‘imgrec’

May 8, 2026

**Type** Package

**Title** An Interface for Image Recognition

**Version** 0.1.4

**Date** 2024-06-21

**URL** <https://github.com/cschwem2er/imgrec>

**BugReports** <https://github.com/cschwem2er/imgrec/issues>

**Description** Provides an interface for image recognition using the 'Google Vision API' <<https://cloud.google.com/vision/>> . Converts API data for features such as object detection and optical character recognition to data frames. The package also includes functions for analyzing image annotations.

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.2.3

**Imports** knitr (>= 1.4.7), base64enc (>= 0.1-0), dplyr (>= 1.1.0), htr (>= 1.4.0), jsonlite (>= 1.8.0), rlang (>= 1.1.0), methods

**Suggests** magick (>= 2.8.0), ggplot2 (>= 3.5.0), usethis (>= 2.2.0), pillar (>= 1.6.0), rmarkdown (>= 2.2)

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Carsten Schwemmer [aut, cre] (ORCID: <<https://orcid.org/0000-0001-9084-946X>>)

**Maintainer** Carsten Schwemmer <c.schwem2er@gmail.com>

**Repository** CRAN

**Date/Publication** 2024-06-21 12:20:05 UTC

## Contents

get_annotations . . . . .	2
gvision_init . . . . .	3
parse_annotations . . . . .	4
save_json . . . . .	5

---

get_annotatons	<i>get image annotations</i>
----------------	------------------------------

---

### Description

Calls the 'Google Vision' API to return annotations. The function automatically creates batches

### Usage

```
get_annotatons(images, features, max_res, mode)
```

### Arguments

images	A character vector for images to be annotated. Can either be url strings or local images, as specified with mode.
features	A character vector for the features to be returned. Accepts 'all' or any combination of the following inputs: 'label', 'web', 'text', 'face', 'landmark', 'logo', 'safe_search', 'object', 'properties'
max_res	An integer specifying the maximum number of results to be returned for each feature.
mode	Accepts 'url' for image urls and 'local' for file paths to local images.

### Value

An response object of class 'gvision\_annotatons'.

### See Also

Google Vision [features](#) and [quotas](#).

### Examples

```
## Not run:

gvision_init()

# one image url
sw_image <- 'https://upload.wikimedia.org/wikipedia/en/4/40/Star_Wars_Phantom_Menace_poster.jpg'
results <- get_annotatons(images = sw_image, # image character vector
                          features = 'all', # request all available features
                          max_res = 10, # maximum number of results per feature
                          mode = 'url') # maximum number of results per feature

# multiple image urls
finn_image <- 'https://upload.wikimedia.org/wikipedia/en/2/2a/Finn-Force_Awakens_%282015%29.png'
padme_image <- 'https://upload.wikimedia.org/wikipedia/en/e/ee/Amidala.png'
```

```
input_imgs <- c(sw_image, finn_image, padme_image)
results <- get_annotatations(images = input_imgs,
                             features = c('label', 'face'), max_res = 5, mode = 'url')

# one local image
temp_img_path <- tempfile(fileext = '.png')
download.file(finn_image, temp_img_path, mode = 'wb', quiet = TRUE)

results <- get_annotatations(images = temp_img_path,
                             features = c('label', 'face'), max_res = 5, mode = 'local')

## End(Not run)
```

---

gvision\_init

*authorization for Google Vision*

---

## Description

Initializes the authorization credentials for the 'Google Vision' API. Needs to be called before using any other functions of `imgrec` and requires `gvision_key` as environment variable.

## Usage

```
gvision_init()
```

## Value

nothing.

## Examples

```
## Not run:
Sys.setenv(gvision_key = "Your Google Vision API key")

gvision_init()

## End(Not run)
```

---

parse\_annotations      *parse image annotations*

---

### Description

Parses the annotations and converts most of the features to data frames. Also stores the corresponding image identifiers for each feature as 'img\_id'

### Usage

```
parse_annotations(annotations)
```

### Arguments

annotations      An annotation object created with [get\\_annotations](#).

### Value

A list containing data frames for each feature:

**labels** label annotations  
**web\_labels** web label annotations  
**web\_similar** similar web images  
**web\_match\_partial** partial matching web images  
**web\_match\_full** full matching web images  
**web\_match\_pages** matching web pages  
**faces** face annotations  
**objects** object annotations  
**logos** logo annotations  
**landmarks** landmark annotations  
**full\_text** full text annotation  
**safe\_serarch** safe search annotation  
**colors** dominant color annotations  
**crop\_hints** crop hints for ratios 0.8/1.0/1.2

### Examples

```
## Not run:
# initialize api credentials
gvision_init()

# annotate images
finn_image <- 'https://upload.wikimedia.org/wikipedia/en/2/2a/Finn-Force_Awakens_%282015%29.png'
sw_image <- 'https://upload.wikimedia.org/wikipedia/en/8/82/Leiadeathstar.jpg'
```

```
padme_image <- 'https://upload.wikimedia.org/wikipedia/en/e/ee/Amidala.png'

results <- get_annotatons(images = c(finn_image, sw_image, padme_image),
                           features = 'all', max_res = 10, mode = 'url')

# parse annotations
img_data <- parse_annotatons(results)

# available feature data frames
names(img_data)

## End(Not run)
```

---

save_json	<i>save annotation data as JSON</i>
-----------	-------------------------------------

---

## Description

Writes raw JSON data as returned by the Google Vision API to a UTF-8 encoded local file.

## Usage

```
save_json(annotations, file)
```

## Arguments

annotations	An annotation object created with <a href="#">get_annotatons</a> .
file	Local path where the JSON data should be stored.

## Value

nothing.

## Examples

```
## Not run:
gvision_init()

finn_image <- 'https://upload.wikimedia.org/wikipedia/en/2/2a/Finn-Force_Awakens_%282015%29.png'
results <- get_annotatons(images = finn_image, features = 'all',
                          max_res = 10, mode = 'url')
temp_file_path <- tempfile(fileext = '.json')
save_json(results, temp_file_path)

## End(Not run)
```

# Index

`get_annotations`, [2](#), [4](#), [5](#)

`gvision_init`, [3](#)

`parse_annotations`, [4](#)

`save_json`, [5](#)