Package 'odiffr'

December 1, 2025

Title Fast Pixel-by-Pixel Image Comparison Using 'odiff'

Version 0.1.0

Description R bindings to 'odiff', a blazing-fast pixel-by-pixel image comparison tool <https://github.com/dmtrKovalenko/odiff>. Supports PNG, JPEG, WEBP, and TIFF with configurable thresholds, antialiasing detection, and region ignoring. Requires system installation of 'odiff'. Ideal for visual regression testing in automated workflows.

SystemRequirements odiff (>= 3.0.0) - https://github.com/dmtrKovalenko/odiff

License MIT + file LICENSE

URL https://github.com/BenWolst/odiffr

BugReports https://github.com/BenWolst/odiffr/issues

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

Imports tools

Suggests knitr, magick, png, rmarkdown, testthat (>= 3.0.0), tibble, withr

Config/testthat/edition 3

VignetteBuilder knitr

NeedsCompilation no

Author Ben Wolstenholme [aut, cre]

Maintainer Ben Wolstenholme <odiffr@benwolst.dev>

Repository CRAN

Date/Publication 2025-12-01 15:00:02 UTC

Contents

`compare_images` *Compare Two Images*

Description

High-level function for comparing images with convenient output. Returns a tibble if the tibble package is available, otherwise a data.frame. Accepts file paths or magick-image objects.

Usage

```
compare_images(
  img1,
  img2,
  diff_output = NULL,
  threshold = 0.1,
  antialiasing = FALSE,
  fail_on_layout = FALSE,
  ignore_regions = NULL,
  ...
)
```

Arguments

`img1`	Path to the first image, or a magick-image object.
`img2`	Path to the second image, or a magick-image object.
`diff_output`	Path for the diff output image (PNG only). Use `NULL` for no diff output, or `TRUE` to auto-generate a temporary file path.
`threshold`	Numeric; color difference threshold between 0.0 and 1.0. Default is 0.1.
`antialiasing`	Logical; if `TRUE`, ignore antialiased pixels. Default is `FALSE`.
`fail_on_layout`	Logical; if `TRUE`, fail if images have different dimensions. Default is `FALSE`.

`ignore_regions` List of regions to ignore during comparison. Use `ignore_region()` to create regions, or pass a data.frame with columns `x1`, `y1`, `x2`, `y2`.

`...` Additional arguments passed to `odiff_run()`.

Value

A tibble (if available) or data.frame with columns:

match Logical; `TRUE` if images match.

reason Character; comparison result reason.

diff_count Integer; number of different pixels.

diff_percentage Numeric; percentage of different pixels.

diff_output Character; path to diff image, or `NA`.

img1 Character; path to first image.

img2 Character; path to second image.

See Also

`odiff_run()` for the low-level interface, `ignore_region()` for creating ignore regions.

Examples

```
## Not run:
# Compare two image files
result <- compare_images("baseline.png", "current.png")
result$match

# With diff output
result <- compare_images("baseline.png", "current.png", diff_output = TRUE)
result$diff_output

# Compare magick-image objects (requires magick package)
library(magick)
img1 <- image_read("baseline.png")
img2 <- image_read("current.png")
result <- compare_images(img1, img2)

# Ignore specific regions
result <- compare_images("baseline.png", "current.png",
                         ignore_regions = list(
                           ignore_region(0, 0, 100, 50),    # Header
                           ignore_region(0, 500, 800, 600)  # Footer
                         ))

## End(Not run)
```

`compare_images_batch` *Compare Multiple Image Pairs*

Description

Compare multiple pairs of images in batch. Useful for visual regression testing across many screenshots.

Usage

```
compare_images_batch(pairs, diff_dir = NULL, ...)
```

Arguments

`pairs`	A data.frame with columns `img1` and `img2` containing file paths, or a list of named lists with `img1` and `img2` elements.
`diff_dir`	Directory to save diff images. If `NULL`, no diff images are created. If provided, diff images are named based on the input file names.
`...`	Additional arguments passed to `compare_images()`.

Value

A tibble (if available) or data.frame with one row per comparison, containing all columns from `compare_images()` plus a `pair_id` column.

Examples

```
## Not run:
# Create a data frame of image pairs
pairs <- data.frame(
  img1 = c("baseline/page1.png", "baseline/page2.png"),
  img2 = c("current/page1.png", "current/page2.png")
)

# Compare all pairs
results <- compare_images_batch(pairs, diff_dir = "diffs/")

# Check which comparisons failed
results[!results$match, ]

## End(Not run)
```

find_odiff *Find the odiff Binary*

Description

Locates the odiff executable using a priority-based search:

1. User-specified path via `options(odiffr.path = "...")`
2. System PATH (`Sys.which("odiff")`)
3. Cached binary from `odiffr_update()`

Usage

```
find_odiff()
```

Value

Character string with the absolute path to the odiff executable.

Examples

```
## Not run:
find_odiff()

## End(Not run)
```

ignore_region *Create an Ignore Region*

Description

Helper function to create a region specification for use with `odiff_run()` and `compare_images()`.

Usage

```
ignore_region(x1, y1, x2, y2)
```

Arguments

x1	Integer; x-coordinate of the top-left corner.
y1	Integer; y-coordinate of the top-left corner.
x2	Integer; x-coordinate of the bottom-right corner.
y2	Integer; y-coordinate of the bottom-right corner.

Value

A list with components x1, y1, x2, y2.

Examples

```
# Create a region to ignore
region <- ignore_region(10, 10, 100, 50)

# Use with odiff_run
## Not run:
result <- odiff_run("img1.png", "img2.png",
                    ignore_regions = list(region))

## End(Not run)
```

---

odiffr_cache_path *Get Cache Directory Path*

Description

Returns the path to the odiffr cache directory where downloaded binaries are stored.

Usage

```
odiffr_cache_path()
```

Value

Character string with the path to the cache directory.

Examples

```
odiffr_cache_path()
```

---

odiffr_clear_cache *Clear the odiffr Cache*

Description

Removes all cached binaries downloaded by `odiffr_update()`.

Usage

```
odiffr_clear_cache()
```

Value

Invisibly returns TRUE if successful, FALSE otherwise.

Examples

```
## Not run:
odiffr_clear_cache()

## End(Not run)
```

---

odiffr_update *Download Latest odiff Binary*

Description

Downloads the odiff binary from GitHub releases to the user's cache directory. The downloaded binary will be used by `find_odiff()` if no system-wide installation or user-specified path is found.

Usage

```
odiffr_update(version = "latest", force = FALSE)
```

Arguments

version	Character string specifying the version to download. Use `"latest"` (default) to download the most recent release, or specify a version tag like `"v4.1.2"`.
force	Logical; if TRUE, re-download even if the binary already exists in the cache. Default is FALSE.

Value

Character string with the path to the downloaded binary.

Examples

```
## Not run:
# Download latest version
odiffr_update()

# Download specific version
odiffr_update(version = "v4.1.2")

# Force re-download
odiffr_update(force = TRUE)

## End(Not run)
```

odiff_available *Check if odiff is Available*

Description

Check if odiff is Available

Usage

```
odiff_available()
```

Value

Logical `TRUE` if odiff is found and executable, `FALSE` otherwise.

Examples

```
odiff_available()
```

odiff_info *Display odiff Configuration Information*

Description

Display odiff Configuration Information

Usage

```
odiff_info()
```

Value

A list with components:

os Operating system (darwin, linux, windows)

arch Architecture (arm64, x64)

path Path to the odiff binary

version odiff version string

source Source of the binary (option, system, cached)

Examples

```
## Not run:
odiff_info()

## End(Not run)
```

odiff_run *Run odiff Command (Low-Level)*

Description

Direct wrapper around the odiff CLI with zero external dependencies. Returns a structured list with comparison results.

Usage

```
odiff_run(
  img1,
  img2,
  diff_output = NULL,
  threshold = 0.1,
  antialiasing = FALSE,
  fail_on_layout = FALSE,
  diff_mask = FALSE,
  diff_overlay = NULL,
  diff_color = NULL,
  diff_lines = FALSE,
  reduce_ram = FALSE,
  ignore_regions = NULL,
  timeout = 60
)
```

Arguments

img1	Character; path to the first (baseline) image file.
img2	Character; path to the second (comparison) image file.
diff_output	Character or NULL; optional path for the diff output image. Must have .png extension. If NULL, no diff image is created.
threshold	Numeric; color difference threshold between 0.0 and 1.0. Lower values are more precise. Default is 0.1.
antialiasing	Logical; if TRUE, ignore antialiased pixels. Default is FALSE.
fail_on_layout	Logical; if TRUE, fail immediately if images have different dimensions. Default is FALSE.
diff_mask	Logical; if TRUE, output only the changed pixels in the diff image. Default is FALSE.
diff_overlay	Logical or numeric; if TRUE or a number between 0 and 1, add a white shaded overlay to the diff image for easier reading. Default is NULL (no overlay).
diff_color	Character; hex color for highlighting differences (e.g., "#FF0000"). Default is NULL (uses odiff default, red).
diff_lines	Logical; if TRUE, include line numbers containing different pixels in the output. Default is FALSE.

`reduce_ram`	Logical; if `TRUE`, use less memory but run slower. Useful for very large images. Default is `FALSE`.
`ignore_regions`	A list of regions to ignore during comparison. Each region should be a list with `x1`, `y1`, `x2`, `y2` components, or use `ignore_region()` to create them. Can also be a data.frame with these columns.
`timeout`	Numeric; timeout in seconds for the odiff process. Default is 60.

Value

A list with the following components:

match Logical; `TRUE` if images match, `FALSE` otherwise.

reason Character; one of `"match"`, `"pixel-diff"`, `"layout-diff"`, or `"error"`.

diff_count Integer; number of different pixels, or `NA`.

diff_percentage Numeric; percentage of different pixels, or `NA`.

diff_lines Integer vector of line numbers with differences, or `NULL`.

exit_code Integer; odiff exit code (0 = match, 21 = layout diff, 22 = pixel diff).

stdout Character; raw stdout output.

stderr Character; raw stderr output.

img1 Character; path to first image.

img2 Character; path to second image.

diff_output Character or `NULL`; path to diff image if created.

duration Numeric; time elapsed in seconds.

See Also

`compare_images()` for a higher-level interface, `ignore_region()` for creating ignore regions.

Examples

```
## Not run:
# Basic comparison
result <- odiff_run("baseline.png", "current.png")
result$match

# With diff output
result <- odiff_run("baseline.png", "current.png", "diff.png")

# With threshold and antialiasing
result <- odiff_run("baseline.png", "current.png",
                    threshold = 0.05, antialiasing = TRUE)

# Ignoring specific regions
result <- odiff_run("baseline.png", "current.png",
                    ignore_regions = list(
                      ignore_region(10, 10, 100, 50),
                      ignore_region(200, 200, 300, 300)
```

```
                    ))

## End(Not run)
```

`odiff_version` *Get odiff Version*

Description

Get odiff Version

Usage

```
odiff_version()
```

Value

Character string with the odiff version, or `NA_character_` if unavailable.

Examples

```
## Not run:
odiff_version()

## End(Not run)
```

Index