

# Package ‘pixels’

December 4, 2020

**Type** Package

**Title** Tools for Working with Image Pixels

**Version** 0.1.1

**Description** Provides tools to show and draw image pixels using 'HTML' widgets and 'Shiny' applications. It can be used to visualize the 'MNIST' dataset for handwritten digit recognition or to create new image recognition datasets.

**License** MIT + file LICENSE

**URL** <https://github.com/javierluraschi/pixels>

**BugReports** <https://github.com/javierluraschi/pixels/issues>

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 3.1.2)

**Imports** htmlwidgets, shiny, miniUI

**Suggests** testthat

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Author** Daniel Falbel [aut, cre],  
Javier Luraschi [aut],  
RStudio [cph]

**Maintainer** Daniel Falbel <daniel@rstudio.com>

**Repository** CRAN

**Date/Publication** 2020-12-04 13:50:06 UTC

## R topics documented:

pixels-package . . . . .	2
get_pixels . . . . .	2
shiny_pixels_output . . . . .	3
shiny_render_pixels . . . . .	4
show_pixels . . . . .	5

---

pixels-package	<i>Tools for Working with Image Pixels</i>
----------------	--

---

### Description

This package provides tools to show and draw image pixels using 'HTML' widgets and 'Shiny' applications.

### Details

This package can be used to: Visualize the 'MNIST' dataset for handwritten digit recognition, create simple test cases for image classification, or to create new datasets by building a 'Shiny' applicaiton that enables many users to capture images for supervised learning.

### Author(s)

**Maintainer:** Javier Luraschi <javier@rstudio.com>

Other contributors:

- RStudio [copyright holder]

### See Also

Useful links:

- <https://github.com/javierluraschi/pixels>
- Report bugs at <https://github.com/javierluraschi/pixels/issues>

---

get_pixels	<i>Gets Pixels</i>
------------	--------------------

---

### Description

Creates an ShinyGadget to retrieve pixels.

### Usage

```
get_pixels(pixels = NULL, grid = c(28, 28), size = c(250, 250),  
brush = matrix(c(0, 0.5, 0.8, 0.5, 0, 0.5, 1, 1, 1, 0.5, 0.8, 1, 1, 1, 0.8,  
0.5, 1, 1, 1, 0.5, 0, 0.5, 0.8, 0.5, 0), 5, 5), params = list(fill =  
list(color = "#555555"), grid = list(color = "#EEEEEE")))
```

**Arguments**

pixels	The pixels to render as a 1-dimensional vector, row-first order expected.
grid	The grid dimensions specified as a vector.
size	The canvas dimensions specified as a vector.
brush	The brush specified as a matrix.
params	A set of parameters to customize the visual appearance. #’ @examples library(pixels) if (interactive()) get_pixels()

---

shiny\_pixels\_output    *Shiny Widget Output*

---

**Description**

Provides a Shiny Widget for Output.

**Usage**

```
shiny_pixels_output(outputId, width = "100%", height = "400px")
```

**Arguments**

outputId	The identifier for this widget.
width	The width for this widget.
height	The height for this widget.

**Examples**

```
library(shiny)

ui <- fluidPage(
  tags$head(
    tags$style(HTML("
      #pixels {
        height: 270px !important;
        margin-top: 10px;
      }
    "))
  ),
  titlePanel("Digit Capture Application"),
  textOutput("prompt"),
  shiny_pixels_output("pixels"),
  actionButton("captureDigit", "Capture")
)

server <- function(input, output) {
```

```
output$pixels <- shiny_render_pixels(  
  show_pixels()  
)  
  
digit <- reactiveVal(floor(runif(1, 1, 10)))  
output$prompt <- renderText(paste0("Please draw number ", digit(), ":"))  
  
observeEvent(input$captureDigit, {  
  digit_path <- file.path("digits", digit())  
  if (!dir.exists(digit_path)) dir.create(digit_path, recursive = TRUE)  
  saveRDS(input$pixels, paste0(digit_path, "/", as.numeric(Sys.time()), ".rds"))  
  
  digit(floor(runif(1, 1, 10)))  
  output$pixels <- shiny_render_pixels(  
    show_pixels()  
  )  
})  
}  
  
if (interactive()) {  
  shinyApp(ui = ui, server = server)  
}
```

---

shiny\_render\_pixels    *Shiny Widget Render*

---

## Description

Renders the Shiny Widget.

## Usage

```
shiny_render_pixels(expr, env = parent.frame(), quoted = FALSE)
```

## Arguments

expr	The expr for shinyRenderWidget.
env	The env for shinyRenderWidget.
quoted	The quoted for shinyRenderWidget.

## See Also

[shiny\_pixels\_output()] for an example of using this function within a 'Shiny' application.

---

`show_pixels`*Show Pixels*

---

**Description**

Creates an HTMLWidget to show pixels.

**Usage**

```
show_pixels(pixels = NULL, grid = c(28, 28), size = c(250, 250),
  brush = matrix(c(0, 0.5, 0.8, 0.5, 0, 0.5, 1, 1, 1, 0.5, 0.8, 1, 1, 1, 0.8,
  0.5, 1, 1, 1, 0.5, 0, 0.5, 0.8, 0.5, 0), 5, 5), params = list(fill =
  list(color = "#555555"), grid = list(color = "#EEEEEE")))
```

**Arguments**

<code>pixels</code>	The pixels to render as a 1-dimensional vector, row-first order expected.
<code>grid</code>	The grid dimensions specified as a <code>c(width,height)</code> vector.
<code>size</code>	The canvas dimensions specified as a <code>c(width,height)</code> vector.
<code>brush</code>	The brush specified as a matrix.
<code>params</code>	A set of parameters to customize the visual appearance.

**Examples**

```
library(pixels)
show_pixels(
  round(runif(400, 0, 1)),
  grid = c(40, 10),
  size = c(800, 200),
  params = list(fill = list(color = "#FF3388"))
)
```

# Index

`get_pixels`, [2](#)

`pixels` (`pixels-package`), [2](#)

`pixels-package`, [2](#)

`shiny_pixels_output`, [3](#)

`shiny_render_pixels`, [4](#)

`show_pixels`, [5](#)