

```
library(boundingBox)
```

Intro

When we create bounding boxes in images, particularly if we intend to feed them into predictive models for object localization and classification, we are often:

- Processing many images
- Classifying the contents of the bounding boxes
- Standardizing the size of the images
- Capturing the coordinates

The **boundingbox** package v1.0.1 which was just recently released on CRAN supports these activities in efficient and effective ways:

- It allows you to stream your images from a directory for processing, in batches if desired.
- It allows you to set a common classifier for all images, or select a classifier per bounding box.
- It allows you to resize all of the output images to be the same.
- It outputs a dataframe of the coordinates along with other metadata points.

There are two primary functions **boxer** and **boxer2** to generate bounding boxes, and the auxiliary function **outBox** to output the images.

Examples

The **boxer** function allows you to stream through images and select between 2 and 26 points around which a bounding box is generated in each image. You can set a common classifier for all of the images.

```
box_coords <-  
  boxer(  
    names = c("S2.jpg", "W1.jpg"),  
    file_path_input = system.file("extdata",      package = "boundingbox"),  
    file_path_output = tempdir(),  
    classifier = "dog",  
    show_classifier = TRUE,  
    resize_x = 224,  
    resize_y = 224,  
    outbox = TRUE  
  )
```

When an image appears, use the left mouse button to select a point, and the right mouse button to signal completion and to move to the next image. To skip through any of the images, use the right mouse button prior to selecting any points with the left button. If the selected point goes out of the bounds of the image, the x and/or y coordinate is adjusted to the nearest set of coordinates on the inside edge of the image.

Here is a screen shot of the first image with several points selected.



This is the output file generated with a bounding box based on the selected points.



Here is a screen shot of the second streamed image with two points selected.



This is the second output file generated with a bounding box based on the selected points.



The resulting data frame will have the bounding box coordinates, the classifier, the image width and height, and box color for each of the images. Note that the y-coordinate extends from the top down, instead of the bottom up for these images.

box_coords

##	file_name	x_left	y_top	x_right	y_bottom	size_x	size_y	classifier	color
## 1	S2.jpg	19	9	201	223	224	224	dog	red
## 2	W1.jpg	41	5	149	216	224	224	dog	red

The **boxer2** function streams images for point capture in the same way as **boxer** does, however it provides the options to add multiple bounding boxes per image, and to select separate classifiers per bounding box. As input, it requires a data frame that defines the classifiers that will be used. You will be prompted to provide the classifier ref # for each of the boxes.

```
dog_df <-  
  data.frame(  
    ref = (1:2),  
    class = c("Skip", "Waltz"),  
    color = c("red", "yellow"),  
    stringsAsFactors = FALSE  
  )  
  
box_coords2 <-  
  boxer2(  
    names = c("SW1.png"),  
    file_path_input = system.file("extdata", package = "boundingbox"),  
    file_path_output = tempdir(),  
    classifier = dog_df,  
    show_classifier = TRUE,  
    outbox = TRUE  
  )
```

Here is an example of output from the boxer2 function.



box_coords2

##	file_name	x_left	y_top	x_right	y_bottom	size_x	size_y	classifier	color
## 1	SW1.png	0	9	122	110	286	320	Waltz	yellow
## 2	SW1.png	157	123	284	245	286	320	Skip	red

Note with both functions it is possible to output just the data frame with bounding box coordinates and not the output images by using the default **outbox = FALSE** setting. You can separately produce the image files by feeding a data frame with bounding box coordinates to the **outBox** function.

Installation

boundingbox can be installed from CRAN or Github.

CRAN

```
install.packages("boundingbox")
```

