Face Replace Activity

Can Mathematica act as an alternative to Image Editing Software?

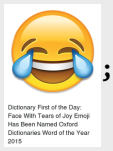
Working with Mathematica's Image Processing functions to overlap, position, alter, and edit images.

Introduction

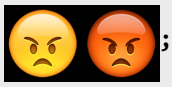
Initialization

In[41]:=

```
tearsofjoy =  ;
```




```
pouting =  ;
```



```
surprised =  ;
```

```
neutral =  ;
```

```
smiling =  ;
```

```
frowning =  ;
```

```
image =  ;
```

```
image2 =  ;
```

```
image3 =  ;
```

```
FacialFeatures[image];
(* If running FacialFeatures here takes too long or it causes an error,
run the initialization cell without it *)
```

Goal of Activity

Replace all of the faces in a picture with emoji's. **First** random emoji's, **then** emotion corresponding emoji's.

Steps to do so:

Make sure emoji pictures are the same size and are transparent

Learn how to combine FindFaces and FacialFeatures with ImageCompose

ImageTake - Removing Unnecessary Parts of Images

In[51]:=
```
tearsofjoy
```

Out[51]=

Dictionary First of the Day:
Face With Tears of Joy Emoji
Has Been Named Oxford
Dictionaries Word of the Year
2015

ImageTake takes a specified amount of rows and columns from the image. The first list argument is the specified rows (height), second is the specified columns (width).

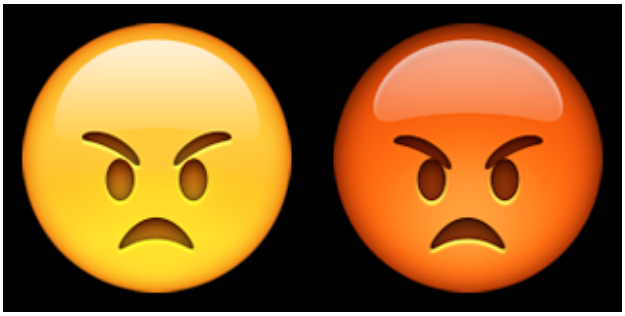
- Tip: one way to easily estimate how much to crop/take is to first look at the Image Information window for image dimensions and to hover over the preview of the original image to see pixel coordinates.

In[52]:=
```
tearsofjoy2 = ImageTake[tearsofjoy, {1, 200}, {1, 200}]
```

Out[52]=

In[53]:=
```
pouting (* 322 x 160 img *)
```

Out[53]=



Because we only want one of the pouting emoji's not both, we want to cut the width in half, so we will ImageTake only around half of the columns.

In[54]:=
```
pouting2 = ImageTake[pouting, {1, 160}, {1, 160}]
```

Out[54]=

ImageCrop - Cropping Margins of Images

Margins in our images will make the resizing of the emoji's to the size of faces in pictures inconsistent. Let's crop them off.

In[55]:=
```
pouting2
```

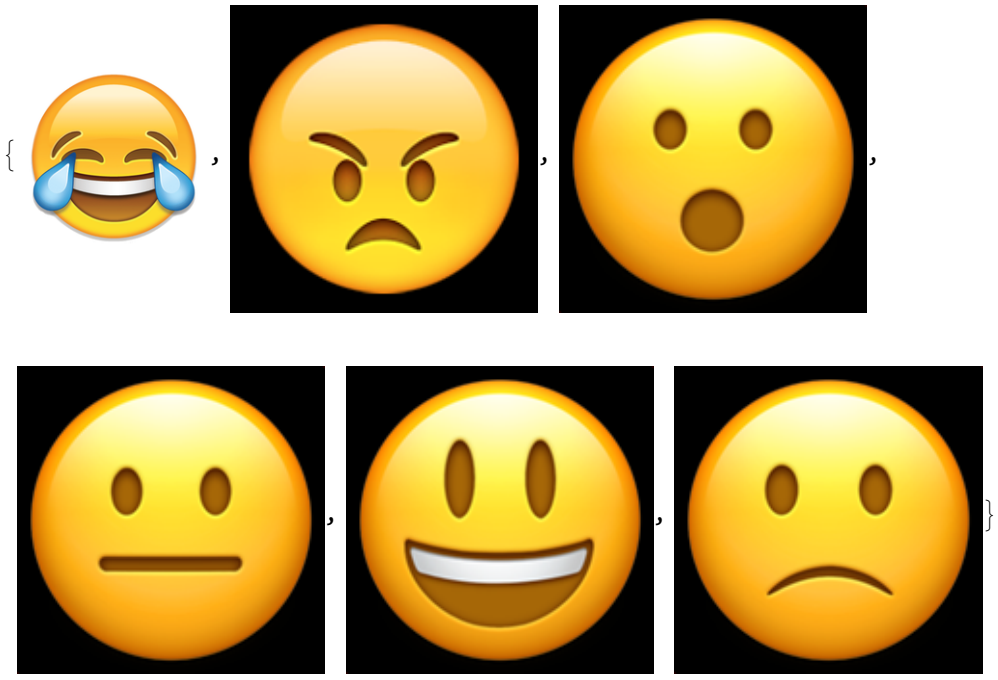
Out[55]=

In[56]:=
```
ImageCrop[pouting2]
```

Out[56]=

In[57]:=
```
emojis = {tearsofjoy2, pouting2, surprised, neutral, smiling, frowning}
```

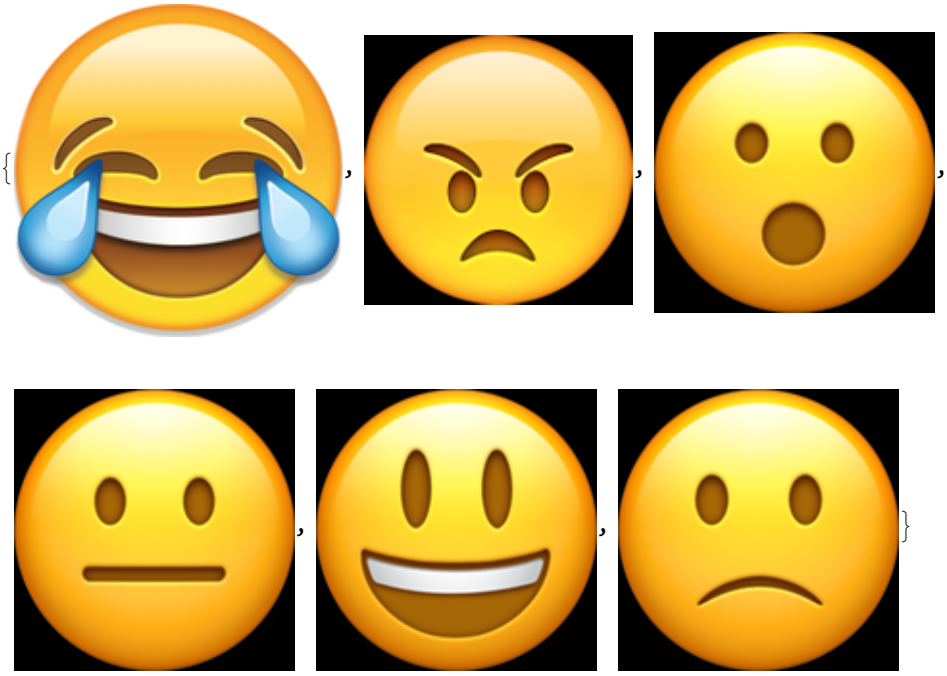
Out[57]=



Map allows us to apply ImageCrop to every emoji in the emojis list.

In[58]:=
```
emojis = Map[ImageCrop, emojis]
```

Out[58]=

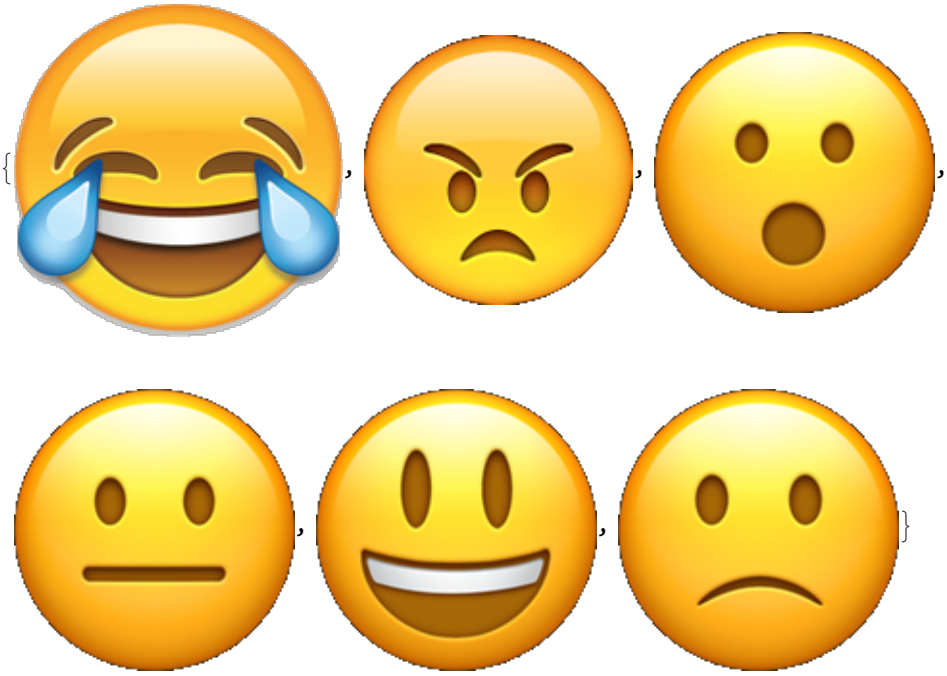


RemoveBackground - Creating Transparent Backgrounds for Images

Next, all we have to do is make the backgrounds of the emoji's transparent, so we will use the Map shortcut /@ to apply RemoveBackground to all the emoji's in the emojis list. The below code is equivalent to doing: Map[RemoveBackground, emojis]

In[59]:= **emojis = RemoveBackground /@ emojis**

Out[59]=



Now emoji pictures are all set, we need to figure out how to overlap them onto the faces of images.

FindFaces - Moving onto Face Detection

FindFaces will return Rectangles with different coordinates that represent the face locations.

In[60]:= **FindFaces[image]**

Out[60]=

```
{Rectangle[{61.5, 163.5}, {107.5, 209.5}], Rectangle[{66.5, 214.5}, {110.5, 258.5}],
 Rectangle[{111.5, 305.5}, {158.5, 352.5}], Rectangle[{165.5, 285.5}, {211.5, 331.5}],
 Rectangle[{232.5, 228.5}, {282.5, 278.5}], Rectangle[{252.5, 161.5}, {295.5, 204.5}],
 Rectangle[{276.5, 306.5}, {323.5, 353.5}], Rectangle[{327.5, 171.5}, {376.5, 220.5}],
 Rectangle[{354.5, 324.5}, {407.5, 377.5}], Rectangle[{403.5, 269.5}, {444.5, 310.5}],
 Rectangle[{463.5, 310.5}, {513.5, 360.5}], Rectangle[{497.5, 231.5}, {541.5, 275.5}]}
```

In[61]:=
```
Show[image, Graphics[FindFaces[image]]]
```

Out[61]=

Instead of rectangles, we want to replace their faces with emoji's instead.

Working with Rectangles - Adjusting emoji's to properly overlap the squares

In[62]:=
```
boxes = FindFaces[image]
```

Out[62]=
```
{Rectangle[{61.5, 163.5}, {107.5, 209.5}], Rectangle[{66.5, 214.5}, {110.5, 258.5}],
 Rectangle[{111.5, 305.5}, {158.5, 352.5}], Rectangle[{165.5, 285.5}, {211.5, 331.5}],
 Rectangle[{232.5, 228.5}, {282.5, 278.5}], Rectangle[{252.5, 161.5}, {295.5, 204.5}],
 Rectangle[{276.5, 306.5}, {323.5, 353.5}], Rectangle[{327.5, 171.5}, {376.5, 220.5}],
 Rectangle[{354.5, 324.5}, {407.5, 377.5}], Rectangle[{403.5, 269.5}, {444.5, 310.5}],
 Rectangle[{463.5, 310.5}, {513.5, 360.5}], Rectangle[{497.5, 231.5}, {541.5, 275.5}]}
```

Rectangle takes in two arguments, the bottom left vertex coordinate and the top right vertex coordinate of the desired rectangle.

1) Resizing emoji dimensions to face dimensions - Finding the widths and heights of each face rectangle

Width will be the difference of x-coordinates in each list in Rectangle, height will be the difference of y-

coordinates in each list.

```
In[63]:= r = Rectangle[{1, 1}, {4, 4}]
Out[63]= Rectangle[{1, 1}, {4, 4}]

In[64]:= r[[1]]
Out[64]= {1, 1}
```

This will return the difference between the two x-coordinates in the Rectangle r, showing the width to be 3.

```
In[65]:= r[[2]][[1]] - r[[1]][[1]]
Out[65]= 3
```

Now we will use this to find the widths and heights for all of the face squares in the image. The # you see in the below maps represents the usage of a pure function - basically in each iteration of the Map, the element will go in place of the #. Here is a simple example where each element in the list is added to one.

```
In[66]:= # + 1 & /@ {1, 2, 3}
Out[66]= {2, 3, 4}

In[67]:= widths = #[[2]][[1]] - #[[1]][[1]] & /@ boxes
Out[67]= {46., 44., 47., 46., 50., 43., 47., 49., 53., 41., 50., 44.}

In[68]:= heights = #[[2]][[2]] - #[[1]][[2]] & /@ boxes
Out[68]= {46., 44., 47., 46., 50., 43., 47., 49., 53., 41., 50., 44.}
```

Huh, all of the rectangles are actually squares. We don't actually need the heights variable then.
Note: In version 12, it seems that FindFaces is more accurate and the rectangles are not actually all squares.
Note - An alternate way to find the width would be the square root of the area of each rectangle.

ImageCompose - Overlapping emoji's over the faces

2) Finding the locations of the faces to correspond to emoji's

ImageCompose will overlap the image in the second argument over the image in the first.

In[69]:=
```
ImageCompose[image, smiling]
```

Out[69]=

Places center of emoji at {200, 200}

In[70]:=
```
ImageCompose[image, smiling, {200, 200}]
```

Out[70]=

Places bottom left vertex of emoji at {200, 200}

In[71]:=
```
ImageCompose[image, smiling, {200, 200}, {0, 0}]
```

Out[71]=

Hmm, FindFaces only gives us the vertices of the face squares through Rectangles, so...
- We can either calculate the middle of the face squares to place center of emoji's there
OR
- Place vertex of emoji's at the given vertex from face squares

Create a list of the coordinate positions of bottom left vertices of each face square by just getting the first argument of each Rectangle.

In[72]:=
```
vertices = #[[1]] & /@ boxes
```

Out[72]=
```
{{61.5, 163.5}, {66.5, 214.5}, {111.5, 305.5}, {165.5, 285.5},
 {232.5, 228.5}, {252.5, 161.5}, {276.5, 306.5}, {327.5, 171.5},
 {354.5, 324.5}, {403.5, 269.5}, {463.5, 310.5}, {497.5, 231.5}}
```

In[73]:=
```
Show[image, Graphics[FindFaces[image][[1]]]]
```

Out[73]=

In[74]:=
```
ImageCompose[image, emojis[[1]], vertices[[1]], {0, 0}]
```

Out[74]=

Nice! Now we just need to resize the emoji.

In[75]:=
```
ImageCompose[image, ImageResize[emojis[[1]], widths[[1]]], vertices[[1]], {0, 0}]
```

Out[75]=

Instead of just the crying laughing emoji, let's use a random emoji so that it will be different each time.

In[76]:=
```
RandomChoice[emojis]
```

Out[76]=

In[77]:=
```
ImageCompose[image, ImageResize[RandomChoice[emojis], widths[[1]]], vertices[[1]], {0, 0}]
```

Out[77]=

Now to make an image that will finally replace all of the faces with emoji's. Because you can only ImageCompose one image at a time, we need to do several ImageComposes on top of each other using Table.

In[78]:=
```
Table[x, {x, 1, 5}]
```

Out[78]=
```
{1, 2, 3, 4, 5}
```

How many times should we repeat the ImageCompose? The amount of faces that there are in the image. Use Length of FindFaces (boxes) to find this number.

In[79]:=
```
Length[{1, 2, 3}]
```

Out[79]=
```
3
```

In[80]:=
```
Length[boxes]
```

Out[80]=
```
12
```

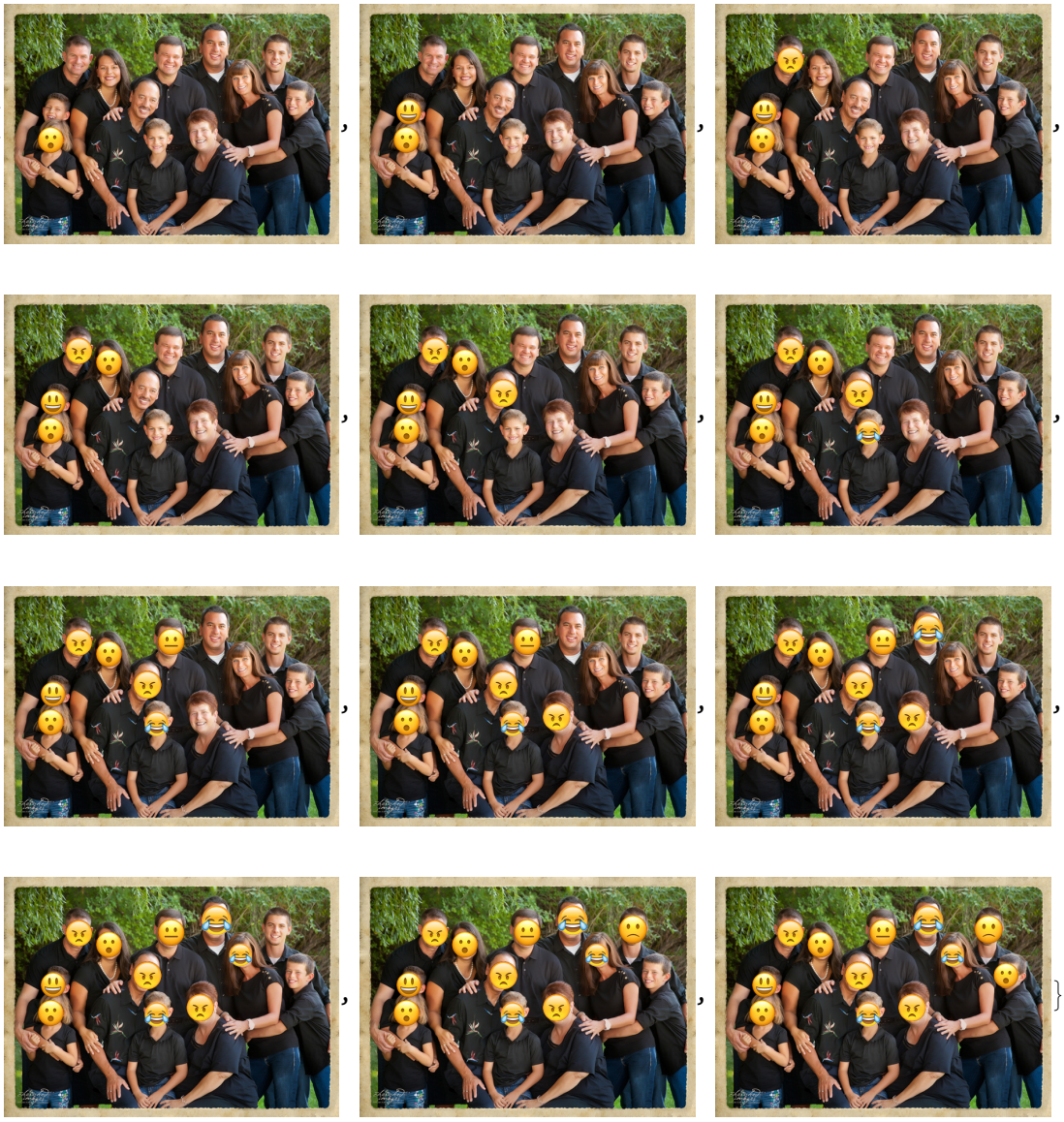
In[81]:=
```
RandomChoice[emojis]
```

Out[81]=

Final Product 1 - Random Emoji's

In[82]:=
```
imageFinal = image;
Table[
 imageFinal = ImageCompose[imageFinal, ImageResize[RandomChoice[emojis], widths[[n]]],
   vertices[[n]], {0, 0}], {n, 1, Length[boxes], 1}]
imageFinal
```

Out[83]=



Out[84]=

Combined Code:

```
(* emojis set up *)
tearsofjoy2 = ImageTake[tearsofjoy, {1, 200}, {1, 200}];
pouting2 = ImageTake[pouting, {1, 160}, {1, 160}];
emojis = {tearsofjoy2, pouting2, surprised, neutral, smiling, frowning};
emojis = ImageCrop /@ emojis;
emojis = RemoveBackground /@ emojis;
(* findFaces set up *)
boxes = FindFaces[image]; (* can replace image w/ different one *)
widths = #[[2]][[1]] - #[[1]][[1]] & /@ boxes;
vertices = #[[1]] & /@ boxes;
imageFinal = image;(* can replace image w/ different one *)
Table[
 imageFinal = ImageCompose[imageFinal, ImageResize[RandomChoice[emojis], widths[[n]]],
   vertices[[n]], {0, 0}], {n, 1, Length[boxes], 1}]
ImageCollage[{image, imageFinal}](* can replace image w/ different one *)
```

Optional Side Note: If you want to use the position of the center of the face squares

The RegionCentroid Function returns the position of the center of the Rectangle, so the **final product** will instead have:

```
centers = RegionCentroid /@ boxes;
imageFinal = image;
Table[
 imageFinal = ImageCompose[imageFinal, ImageResize[RandomChoice[emojis], widths[[n]]],
   centers[[n]]], {n, 1, Length[boxes], 1}]
```

Combined Code:

```
(* emojis set up *)
tearsofjoy2 = ImageTake[tearsofjoy, {1, 200}, {1, 200}];
pouting2 = ImageTake[pouting, {1, 160}, {1, 160}];
emojis = {tearsofjoy2, pouting2, surprised, neutral, smiling, frowning};
emojis = ImageCrop /@ emojis;
emojis = RemoveBackground /@ emojis;
(* findFaces set up *)
boxes = FindFaces[image]; (* can replace image *)
widths = #[[2]][[1]] - #[[1]][[1]] & /@ boxes;
centers = RegionCentroid /@ boxes;
imageFinal = image; (* can replace image *)
Table[
 imageFinal = ImageCompose[imageFinal, ImageResize[RandomChoice[emojis], widths[[n]]],
   centers[[n]]], {n, 1, Length[boxes], 1}]
ImageCollage[{image, imageFinal}]
```