Chris Piech and Mehran Sahami
Code in Place

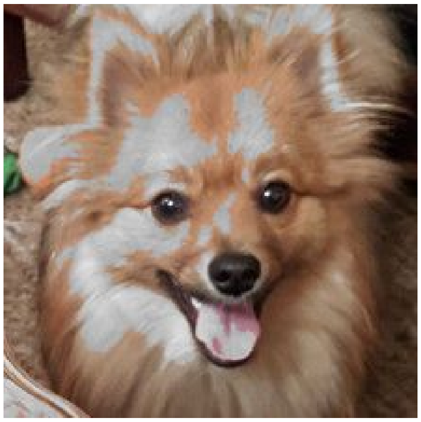
Section #4
May 4, 2020

Section Handout #4: Images

This week you get to write programs that manipulate images in exciting ways. It's a fun time and also gives you insight into how some real photo-editing software like Adobe Photoshop works!

1. Our Section Filter

Write a program that applies a "Narok" filter to an image.



Note: Write all of your code in the main() function.

Hopefully, you find the filter aesthetic. It will certainly help you with your assignment. This filter heavily uses the idea of a pixel average. For a pixel, we say its "**pixel average**" is the average of its red, green, and blue components.

To apply this filter, do the following for each pixel:

- **If the pixel is "bright,"** then you should make it greyscale. We consider a pixel to be bright if the "pixel average" is greater than 153 (which is 0.6 * 255, or sixty percent of the max brightness value). For this problem, to make a pixel grey, we are going to set each of its red, green, and blue channels to be equal to the average.
- **If the pixel is not bright,** then you should leave it the same color.

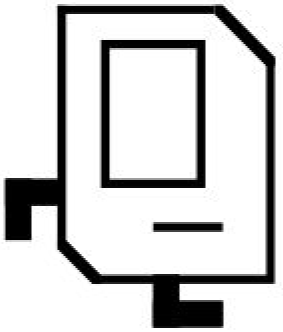
2. (Optional) Trim-Crop

Write a function

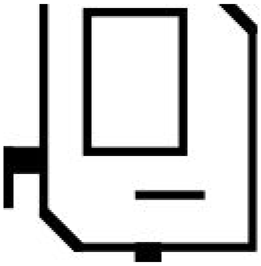
```
def trim_crop_image(original_img, trim_size):
```

which returns a new image that is the same as the original image but with `trim_size` number of pixels removed from each side (top, bottom, left, or right) of the original image. You may assume that `trim_size` is less than half of either the dimensions (width or height).

For example, suppose we have this picture of Karel:



If we called `trim_crop_image()` on this picture above and removed `trim_size = 30` pixels from all sides, it would produce an image that looks something like this:



Note that the dimensions of this new image are smaller (both the width and the height have been reduced by 60 pixels).

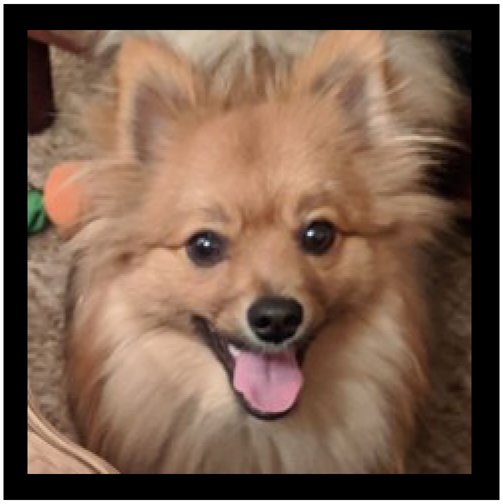
3. (Optional) Add Border

Need an extra challenge? How would you add a black border to an image? Define a function

```
def add_border(original_img, border_size):
```

That takes an image and returns a new image that has a border of width `border_size` added to each side of the image (top, left, bottom, right). The inner picture remains the same size. For example, suppose we have an original image simba-sq.jpg:

If we add `border_size = 10` pixels to the above image, we would get this:



Notice that the bordered image is 20 pixels wider (10 pixels on the left and 10 pixels on the right) and 20 pixels higher.