

Joshua Boben and Alfonso Miguel Perez

5/19/16

Per. 5

Using Computer #62

Conclusion

1. Alice and Barb have different ideas about what a “manipulated image” is. Decide whether you think that each of them is right, wrong, or somewhere between. Write an argument in support of your ideas.

Alice: “All images are manipulated. For one thing a camera is sensitive to certain kinds of light and the developer controls the exposure level. Even our human eyes have a limited number of pixels! There are ‘only’ 120 million rods and 6 million cones in each retina, so our vision is pixelated just like a digital image. And our vision is also highly processed – even the blind spot in each eye gets filled in. Out of all those millions of light detectors, only about 1 million ganglia neurons go from the eye to the brain. There is no such image as seeing the ‘real’ thing.”

Barb: “Of course there is a real image. Certain kinds of manipulations are accurate and others tell lies.”

I agree with Barb. Certain image manipulations are accurate and only show us the image in a clear way. Other image manipulations change the picture into something else, classifying it as telling lies.

2. Under what circumstances is an image yours to use? Yours to distribute? Yours to sell? Write about your thoughts on this question in the context of downloaded images and images you take with a camera.

An image is yours to use if you made the picture or somebody granted you permission to use the image. It is yours to distribute it if somebody gave you permission or if you own the image in any way. The same applies for selling images. This also applies to other products as well and not just images.

3. Reflect on the team dynamic and on the design process. What were areas for improvement? What steps could you take next time to make those improvements?

The team dynamic was experimental to say the least, we had to constantly change the change to the image. One area of improvement is how we download and save the image file we changed. Next time we should take an image that is easily convertible as to not have trouble trying to change it. We should have also focused on 1.4.7 more than 1.4.6 so we could have focused more on this project.

Name of folders needed to create

1. C:/original_pictures/ (contains images)
2. C:/output_pictures/ (contains product of the modified images)

What the client wants: The client, a family, wants a frame to be applied on a large collection of pictures which feature members of the family. The client would also like some geometric shapes in the image. They want it to be memorable and for it to represent what the family is interested in. Our client would like a range of options regarding the image operations.

What we know about the project requirements: We have to take multiple images and be able to make different kinds of changes to it depending on what the customer wants. The customer would be able to add geometric shapes to it. This product would feature multiple images since that is part of the requirements the customer wants.

How to handle images in Python: We searched this on Google and it recommended us to use Pillow and opencv.

Pillow is a PIL fork. PIL stands for Python Imaging Library

Opencv is an open source machine learning library.

All images are from <http://www.istockphoto.com/>

Documentation

At first when we tried the code, it produced lots of errors when using Image. When we restarted Canopy and ran the code, it went away.

Code so far:

```
from PIL import Image, ImageFilter
import os
```

```
im = Image.open("C:/Users/jvbob/Desktop/Images/beagle-picture-id537410686.jpg")
print im
im.show()
```

We added more images into the code. We also removed the print im from it as well.

```
from PIL import Image, ImageFilter
import os
```

```
im = Image.open("C:/Users/jvbob/Desktop/Images/beagle-picture-id537410686.jpg")
im.show()

im2 =
Image.open("C:/Users/jvbob/Desktop/Images/cooking-barbecue-picture-id684116350.jpg")
im2.show()

im3 =
Image.open("C:/Users/jvbob/Desktop/Images/family-helping-their-son-to-ride-a-bike-picture-id678513632.jpg")
im3.show()

im4 =
Image.open("C:/Users/jvbob/Desktop/Images/young-ethnic-girl-shows-off-her-fullgrown-chicken-to-a-family-friend-picture-id685771906.jpg")
im4.show()
```

We decided to remove the `im.show` from the rest of the code and only include it in the end. We also used `Image.blend`. We experimented with different values using `Image.blend`. We tried 9.9, which made the image look very strange. 0.5 was the perfect balance between `im` and `im2` without making the combined images look weird.

```
from PIL import Image, ImageFilter
import os

im = Image.open("C:/Users/jvbob/Desktop/Images/beagle-picture-id537410686.jpg")

im2 =
Image.open("C:/Users/jvbob/Desktop/Images/cooking-barbecue-picture-id684116350.jpg")

im3 =
Image.open("C:/Users/jvbob/Desktop/Images/family-helping-their-son-to-ride-a-bike-picture-id678513632.jpg")

im4 =
Image.open("C:/Users/jvbob/Desktop/Images/young-ethnic-girl-shows-off-her-fullgrown-chicken-to-a-family-friend-picture-id685771906.jpg")
nnfenfw
```

```
Image.blend( im, im2, 0.5).show()
```

The lines of code above was to test how to blend and open images. This will help us for making the actual product.

[This link](#) and [this link](#) are the finalized product as seen and can be analyzed on Github.

WRITE-UP

We came up with our program after going through multiple examples that we found on the internet. Google, Stackoverflow, and similar websites assisted us in this project. Our idea was to create an algorithm which was able to successfully modify the image to meet the needs of our client, a family. We chose images from the website <http://www.istockphoto.com/> to demonstrate what our product was capable of doing. Some of the code seen in this document were the trial to find out how to use the packages to see how to modify images and to get a general idea of how they work.

Our product would blur, apply the contour filter, make a thumbnail, blend and rotate, add a border, and print text on the image. At first we decided to get rid of the blend feature, but then we realized it would be better to leave it in there as it produced some interesting images.

This program needs two folders named original_pictures and output_pictures in the C drive. The images we want to modify will be put in the original_pictures directory in the C drive. The modified images will be saved and can be accessed in the output_pictures folder in the C drive.

We use image, imageops, image filter from the Pillow packages to help us manipulate the images. We need Pillow and os packages installed in order for our program to run. These packages should be available for installation on Enthought Canopy if they are not already available for you.

Our program has a python class which we pass in the full path of the files and call each function to manipulate the image. The class file is called HandleImage.py and the testing program is called HandleImageFinal.py. The class is named HandleImage.