**Hi William**

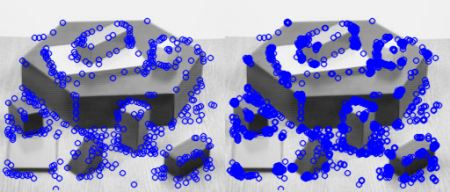
Enhancements

1. Glare

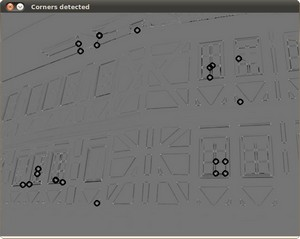
I did a bit of research & I agree that glare seems to be a problem that everyone has issues with.

I also do not know opencv capabilities – so my question may be crazy. But – can opencv find 3 of the corners of the rectangle? See attached image. Is it possible to find the three farthest corners & then extrapolate the 4th corner? With a card – we can always assume a rectangle.

This option could be explored. From the result below it seems that it won’t be an exact science, again. I will have to fiddle with settings until I can find a solution, if there is one.







1. Image 8. The result (88) has been flipped upside down. Without ocr’ing the image – is there are way to determine that it is upside down.

Unfortunately there really is no way to do this, not even with advanced ocr detection as these packages require text to be in correct orientation.

Prepare for production

Your code needs to be made into a function we can call from GO.

We need to prepare the code to be included in our code base. Your function will be called by our GO web service. We need to remove the read / write functionality from your code. Can you develop a wrapper function – that will call your code. On our side – we will replace the wrapper function with our GO service.

I can create a function that takes as input a cv2 image and returns the cv2 image object. How you will call this from Go I am not so sure about because in python it reads an image into a cv2 object and then does the processing from there. Would you be able to work with a function that simply takes the cv2 image object and returns the processed cv2 image object? The other option is to still specify the input image name and the output image name and have python save the output as an image on hdd.

1. Code cleanup
   1. Remove code that has been commented out.

OK

* 1. Develop a wrapper function - main()
     1. Move reading & writing the files to the wrapper function - main()

OK

* + 1. Wrapper calls your current code
       1. Passes the image

What image object will you be passing? What I mean is that the image at the moment is a cv2 image object.

Will the go program pass:

1- Name of input image file on HDD?

2- Image file as cv2 image object?

3- Another image object type?

* + - 1. Passes back the cropped image (if it succeeded)

Will python pass back:

1- Name of exported image file on HDD?

2- Image file as cv2 image object?

3- Another image object type?

* + - 1. Returns status (int)
         1. 0 = FAILURE
         2. 1 = SUCCESS

OK

* + - 1. If(SUCCESS)
         1. Write files to disk - current write functionality

OK