Hi William

I have done a few other changes, besides what is mentioned below.

I have changed the rescaling interpolation method because I found out the one I used had was filtering out edges.

I have done some digging and found a way to install scipy without having to install the whole anaconda distribution. The issue was, to install scipy one needed to have the intel fortran compiler on your system. I found prebuild compilations of numpy and scipy at [http://www.lfd.uci.edu/~gohlke/pythonlibs/#scipy](http://www.lfd.uci.edu/~gohlke/pythonlibs/" \l "scipy)

In the source code I have added temporary test code, this is just to help me to quickly test different image files for success. I will remove this in future.

~~Ryan~~

~~I’ve tested against a larger set of images. I executed for each of your various versions, on each file.~~

~~The mages have been renamed – to make it easier to track.~~

~~Unless I am missing something - the following command is what we want~~

~~python scan.py -i “in-image” -o “out-image” -f 5,5 -P 0.04~~

~~It found images for~~

* ~~1~~
* ~~6~~
* ~~7 - rotated image 90 degrees clockwise~~
* ~~8 - found top half & flipped vertically~~
* ~~9 - tough one – there are two blocks – excluded the top block~~
* ~~A - distorted image – suggest we catch distortion & throw error~~
* ~~D~~

~~Issues we need to work in – in order of priority~~

1. ~~Text distortion – the process is blurring the text. - This is~~ **~~important~~**

* ~~Is it possible to either~~
  + ~~Clean up the blur~~
  + ~~Use the x /y coordinates to crop the original image – rather than produce a new image~~

~~I have disabled the bw option which transformed the original images to a black/white format~~

1. **Resolve rotation issues for images - at least provide the same orientation that the original image had.**

* **7**
* **8 - This may be resolved in # 4 & # 8 below**

~~I am note sure what you mean because for image 7 is Portrait and the card is also portrait. Image 8 is Landscape and the image is Landscape also. I have built code to detect if the original image is landscape and the card is not then to rotate the card accordingly, same if the card is portrait.~~

Image 8 is the one where I was able to use detection method 2 (based on image contrast). **Should I code the program to test method one 1st for all images and if it doesn’t detect then it tries method 2. Note that method 2 only works where there is a dark background (such as image 8)**

1. ~~Save color. If we can crop original image – as suggested in #1 – that should fix this. Otherwise – is there another way to bring color back.~~

~~done (as above)~~

1. ~~Catch severe image distortions and throw an error - such as “Image not Readable”. One way to do this is to assume basic x / y proportions. Example~~

~~Correct proportions~~

* ~~Letter paper - 8 ½ x 11 proportion = 8.5 / 11 = .77~~
* ~~Insurance card - 5.4 cm x 8.5 cm proportion = 5.4 / 8.5 = .64~~

~~Incorrect proportions (distorted)~~

* ~~Image 88 - 358 pixels x 1914 pixels proportion = 358 / 1914 = .19~~
* ~~Image AA - 61 pixels x 371 pixels proportion = 61 / 371 = .16~~

~~If proportions are less than .50 – then reject image. If we made the~~ **~~proportion an argument~~** ~~– the calling program could adjust the proportion – based on what the size of the “target” in the original image.~~

1. ~~Have process always return a status~~

* ~~Success~~
* ~~Edge not found~~
* ~~Image not readable~~

1. ~~Add a debug argument.~~

* ~~If null (default) – do not write output image – unless status is “success”~~
* ~~If not null (1) – handle output as you are now.~~

This is currently what the program does. (after making changes to test the ratio, and also flipping the image according to the main image’s rotation. If it detects something, it will write the output file, if not it will give one of two errors at the command prompt: “Edge not found” or “Image not readable”.

**If you need something else then can you please explain to me a little bit more what you mean here.**

1. See what can be done to successfully crop the following images. I realize the issue is the glare. But many of these cards have a glossy finish & slight glare is a very likely occurrence.

* 3
* 4
* ~~5 detected with scan.test(True, 8, 8, 0, 0.1, 'images\\5.jpg')~~

~~I have tried to detect these images with option 2 detection, but since there is not enough contrast between card and background this will not work. Using the standard method does not detect the borders because the edges of the card is too blurred.~~

1. Is there a way of handling the issue in image 8? We are losing all needed content – because the card has a line that separates the header from the body. It chooses the header and crops the body of the card. This card is for one of the largest insurance companies. We need to be able to successfully crop this card. **View my question at point 2**
2. Is there a way to handle image 9. This is similar to the issue with image 8. Does the process see two blocks – but choose the prominent one? If the process is able to identify two complete rectangles, can it be smart enough to realize that both rectangles have the same two x axis coordinates? Only their Y coordinates are different. In which case – crop at

* The lowest x  highest x  highest Y coordinates

This would successfully crop the entire image. This would also resolve the issue with image 8.

**What I can do here is try to detect two separate blocks in each image and then paste them together into one image?**