ECAM

BRUSSELS ENGINEERING SCHOOL

COMPUTER VISION LABS :

CHEAT THE ISHIHARA TEST

20305 TARDIF Félix

21/12/2022

Table des matières

Description of the context & expected results

The Ishihara test purpose is to diagnose variations of colour blindness or deficiencies.

I will try to cheat the test using computer vision.

First i will highlight the 2 major colors in the image. There is of course more than 2 colors components in an Ishihara traditional plank but there are always 2 major colors in the drawing, usually green & orange.

In the same time I will change the color of the plank in Black & White, in order to highlight the number written on it.

When the previous objectives are done, I will do the same but with a frame captured by my computer's camera.

Identification of the key challenges of the problem

The main difficulty of this project is to highlight the differents colors that are on the plank, because the test is made in such a way that it is difficult to distinguish the colors.

As I said previously, there are 2 major colors component in an Ishihara plank (one for the background and another one for the number itself). In reality, the majority of the planks are composed of variations of these 2 colors.

For example, the background is composed of several shades of green while the numbers are composed of several shades of orange, as you can see below in the Figure 1. There is a lot of plank doing the opposite, numbers in shades of green and background in shades of orange, like in the Figure 2.

Figure 1: Ishihara plank n°2

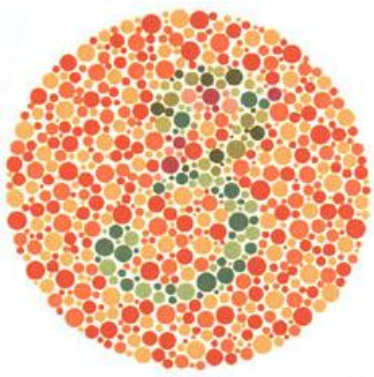

Figure 2: Ishihara plank n°7

Strategy to deal with the aforementioned challenges

After looking on the internet for a while, I discovered a certain type of Colorspace called LAB. *L* for lightness, *a* for red/green and *b* for blue/yellow. It was originally created to approximate human vision.

Using openCV, I first changed the original plank to the LAB colors. Then I only kept the *a* channel and the *b* channel in order to isolate the differents colors. You can see the result in the next Figures.

Figure 3: Plank N°1, Original colors

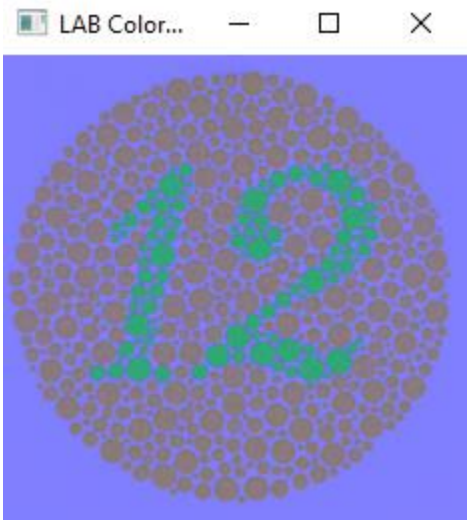

Figure 4:Plank N°1, LAB ColorSpace

Figure 5:Plank N°1, Only the A channel of the LAB Colorspace

Figure 6:Plank N°1, Only the B channel from LAB ColorSpace

As you can see, the number is now clearly visible.

Now comes the moments where we can play a little bit, using OpenCV.

I first used the A-channel result. I applied a *threshold* on it in order to get a black & white image,by playing with the threshold level of the function.

Figure 7: Plank N°1, Raw binary output

Then I applied a *Morphology* operation in order to eliminate the black spaces in between the image.

Finally I used the *medianBlur* function to get a « smooth » output image, by playing with the kernel's value of the function.

Figure 8: Plank N°1, Smooth binary output

The idea was to take differents values for the *threshold* and for the *medianBlur* function, then I had to check the quality of the answer.

Quality of the results

As you can see, the result is not that impressive but it still worked pretty well. But you'll see that it is not that simple to cheat the Ishihara test. Mainly because of the differents type of plank present in the test.

For each type, I need a different combination of *threshold* & *Blur* value. Then, depending of the type of the plank, I need to use the right values in order to get a presentable output image. In the end, the results were good for the green-background planks but it worked a bit less for the orange-background planks.

Then I tried the same strategy, but using the B-channel from the LAB ColorSpace. I did some modifications in my code, like using the *THRESH_BINARY_INV* parameters instead of the basic *THRESH_BINARY* used in the first place and I also changed the *threshold* value. The *medianBlur* value has remained the same.

Surprise : it works nearly the same as with the A-channel, sometime even better ! And for every type of planks !

Obviously I prefered this way of resolving the test thus I used this strategy to elaborate another function able to detect and transform an Ishihara plank provided from my computer's camera.

I will provide a link to the little demonstration of my function :

https://ecambxl-my.sharepoint.com/:v:/g/personal/20305_ecam_be/EfoXjsoAU_hDnpnF7d5Ier0Bm5LzCfkiAf5uKddg0zegzw

How can i take it further ?

Even if I'm happy with my result, I know that there is a lot more I can do with this project.

Here is a non-exhaustive list of what can be improved :

- For now my function only take a frame of the camera and then process it. It would be nice if I could process the plank and display the result while my camera is still on live.

- As I've seen on the internet, the *blur* effect can be part of a « for-loop » in order to get a ouput image of better quality, instead of playing with the *medianBlur* parameters value.

- A new function that analyses the plank's colors and then tunes the parameters (*threshold* & *medianBlur* & *number of iteration for bluring*) of my main function to their optimal value.

- The function is able to detect and translate the number written on the plank (we're slowly heading toward AI computing).