Image Processing

Template matching with distortion

Instructor: Dr. Nguyễn Thị Ngọc Diệp

Class: 2021 INT3404 20

Student: Nguyễn Mạnh Dũng 18020370

1. INTRODUCE

- Nowaday, detecting or finding small parts of an image which match a template image were commonly used in manufacturing as a part of quality control, a way to navigate robots or detecting edges in images. That technique was called “template matching”.
- Template matching algorithms are all about finding areas of the current image (unknown) which are simillar to the template (known).
- The algorithm allows us to detect the position of the object within the current image, some can detect rotated object, scaled object, even object with distortion (main topic).

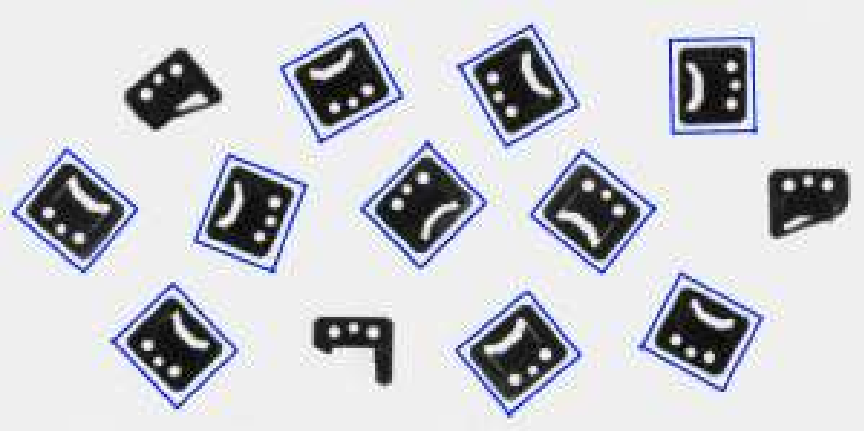

Figure 1: Template matching example

2. TEMPLATE MATCHING ALGORITHM

2.1. Template matching with Cross-correlation

- With Cross-correlation, we have input are the image that we need to detecting and the template of that part(s).
- All the outcome of template matching will be original image with bounding box in the part(s)/area(s) that are similar to the template.
- Call the image is I and the template is T = 2*k + 1
- Because we need to use “same” correlation so before correlate 2 arrays we need to adding k pad to the edge.
- We use T to correlate with I, slide-window is refered to 1, it’s shown by this equation:

$$R(x,y) = \sum_{s,t} \big(T(s,t).I(x+s,y+t)\big)$$

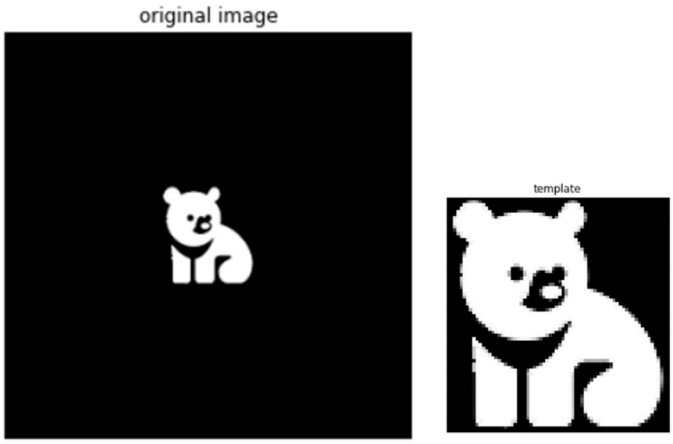

Figure 2: Original image and template

For example:

- This is the template T that we need to detect from original image.
- By using cross-correlation, the areas have exactly the same value with the template will become brightest areas in the correlation result *(Figure 4.)*, in other word, the pixel have the biggest value after correlation is the center of the part that need to be detected.

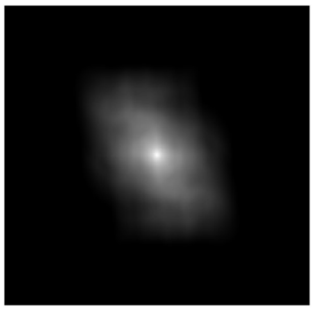

Figure 3: The brightest point is the center point

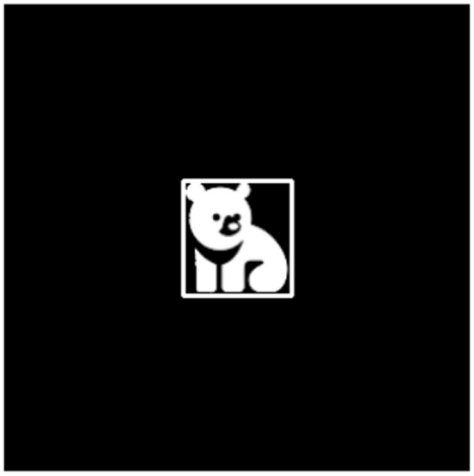

Figure 4: The result

- The weakness of this algorithm reveals when came to the image have a lot of white (bright) areas.
- In this example, my teddy-bear background is white, the teddy-bear itself is black:

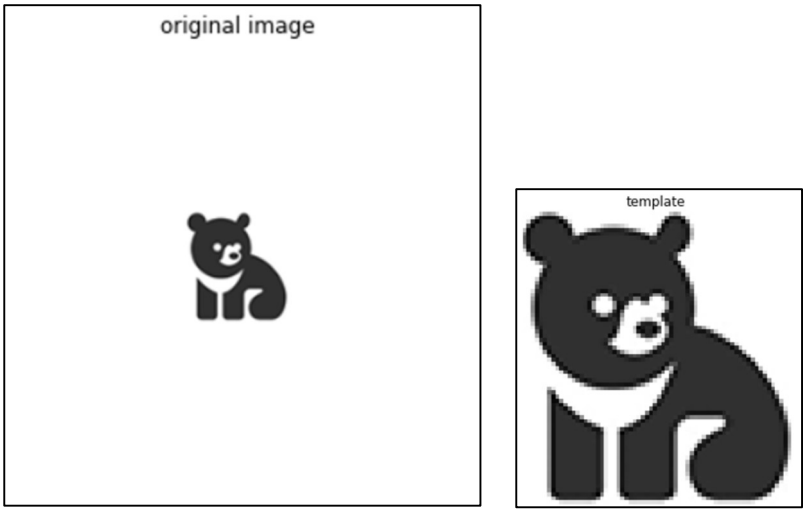

Figure 5: Image that have a lot of bright areas

- After correlation, the result makes the function confused where to draw the bounding box. The yellow areas in contour graph show the brighter in correlation result.

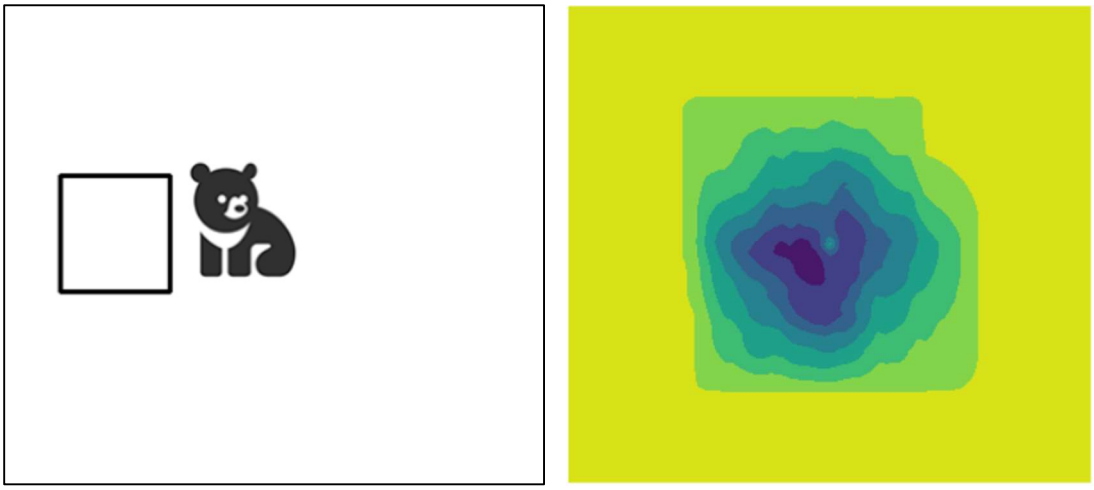

Figure 6: There's no brightest point

- To deal with this issue, we simply can negate original image. Unfortunately, this solution might can solve problem in this example, but it can not be apply to all situation. Nagate image won't work with image with high contrast, it means when there're a lot of bright and dark area in the same picture, negate image is nearly nonsense. For that, we need another algorithm.

2.2. Template matching with Correlation coefficient

- Correlation coefficient is a technique which slightly the same as correlation, the biggest difference between these 2 algorithms is when using correlation coefficient, we take the template minus avenge value of that template before correlate:

$$T'(x,y) = T(x,y) - \frac{1}{w.h}\sum_{x'',y''} T(x'',y'')$$

- We also do that with the image:

$$I'(x+x', y+\ y') = I(x+x', y+y') - \frac{1}{w.h}\sum_{x',\ y''} I(x+x'', y+y'')$$

- Correlation coefficient is also shown by this equation:

$$R(x,y) = \sum_{x',y'} (T'(x,y)\ .\ I'(x+x', y+y')$$

- The correlation coefficient takes the value of [-1, +1]
- These pre-processing (T' and I') can solve the problem of cross-correlation, we can deal with imgae that have high constrast which is a bit more complex than usual.

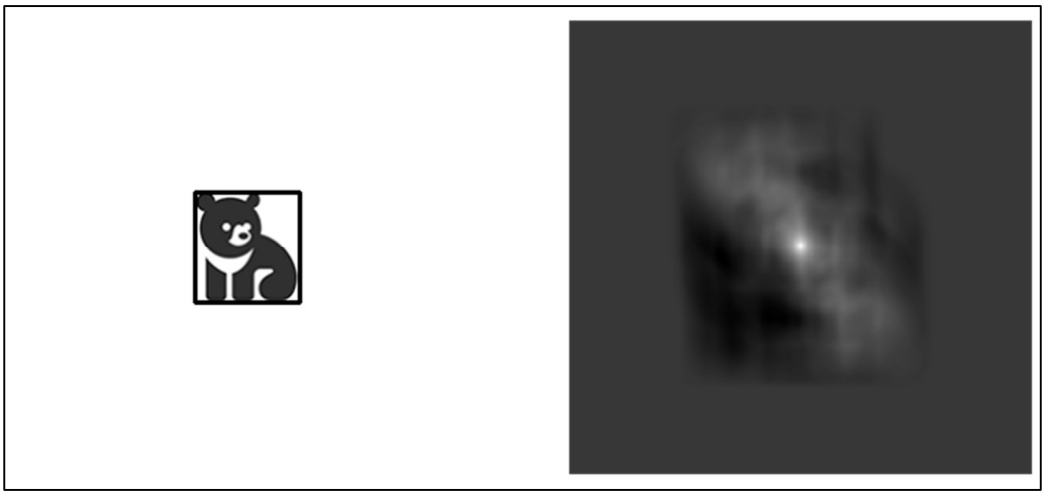

Figure 7: CCEFF for example in figure 5

- In the previous example, if we use correlation coefficient (CCOEFF), the white background was turn into dark color due to the change of template and image. For more detail of why change the template and image like we discuss above can solve the white background problem, look up this example:

If we have 2D template like shown:

$T(x', y') =$

0	1	0
1	1	1
0	1	0

If we calculate the cross-correlation and this kernel go throught some areas that have the value like this: $I_1 =$

1	1	1
1	1	1
1	1	1

This can not be considered as the similar part to the template. Howerver, the correlation operator still return value=5, which is the highest value that can be in this operator with this template.

Now, let do the same example but with the correlation coefficient:

Step 1: *Calculate the new template*

$$T'(x, y) = T(x, y) - \frac{1}{w.h} \sum_{x'', y''} T(x'', y'')$$

After operate, we have new template:

$I_1'(x', y') =$

-5/9	4/9	-5/9
4/9	4/9	4/9
-5/9	4/9	-5/9

Note: *Without going anywhere futher, at this point we can see that if the area we are stuck have the value like this: $I_1 =$*

1	1	1
1	1	1
1	1	1

, so there's no need to fear that the operator will return the highest value. In fact, the return result is value=0, which is not the part we are looking for, so the white problem solved!
If we didn't notice this fact, that's fine, the next step will be:

Step 2: *Calculate the new image*

$$I'(x + x', y + y') = I(x + x', y + y') - \frac{1}{w.h} \sum_{x'', y''} I(x + x'', y + y'')$$

After done that, from some array like I_1, we will have: $I_1' =$

4/9	6/9	4/9
6/9	1	6/9
4/9	6/9	4/9

Step 3: *Correlation coefficient*

The final result we have is: $R_1(x, y) =$

-1/81	4/9	-1/81
4/9	40/81	4/9
-1/81	4/9	-1/81

Compare to the value $R_2(x,y) =$ | 19/81 | 20/81 | 19/81 |
|---|---|---|
| 20/81 | 8/27 | 20/81 |
| 19/81 | 20/81 | 19/81 |

if we CCOEFF this $I_2 =$ | 0 | 1 | 0 |
|---|---|---|
| 1 | 1 | 1 |
| 0 | 1 | 0 |

with the template.

Clearly the R2 was more similar to the template than the R1, which is true. ***The white background has been solved by CCOEFF.***

2.3. Using CCOEEF to detect template with disortion

- Can CCOEEF be used to detect template with disortion?

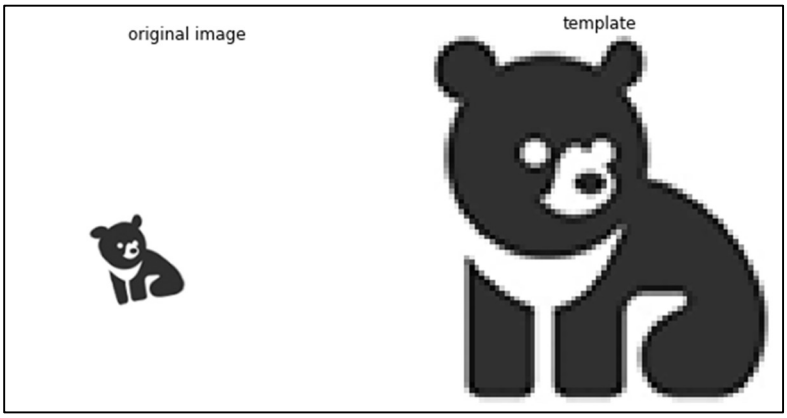

Figure 8: Disortion example 1

- We have the original image with the teddy-bear which I've already bend a little bit at his head and rotate.

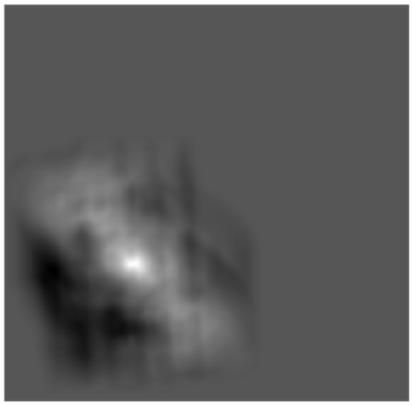

Figure 9: Example 1 result

- The outcome is close enough to the correct answer, the reason why the bounding box doesn't cover all the part of the teddy-bear is because the template's height and weight. In my code, the bouding box has been drawn

by detecting the top left point, then the bottom right is calculated by adding height and weight of the template.

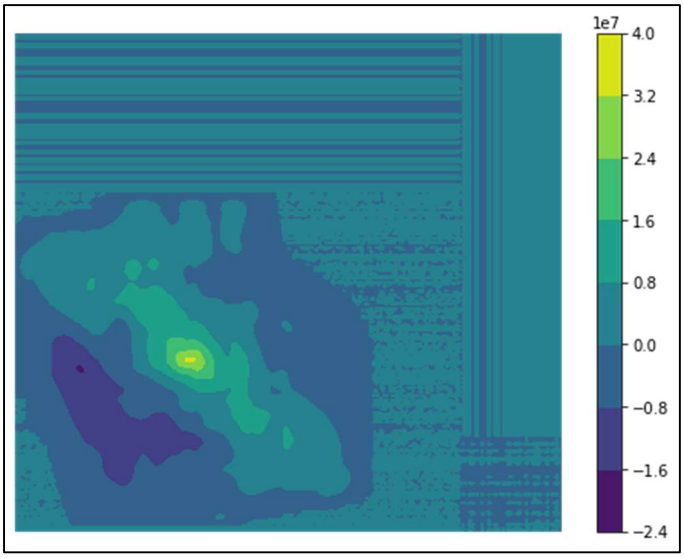

Figure 10: Example 1 convour graph

- As we can see in the contour graph of this result, the brightness areas are spread all over the diagonal and the the contour graph of the previous example (with out disortion) also have the same shape of the brightness areas, which can lead us to conclude that CCOEFF can still detecting image with disortion because of maintaining the main pattern of the brightness areas after operated.
- Let take another example:

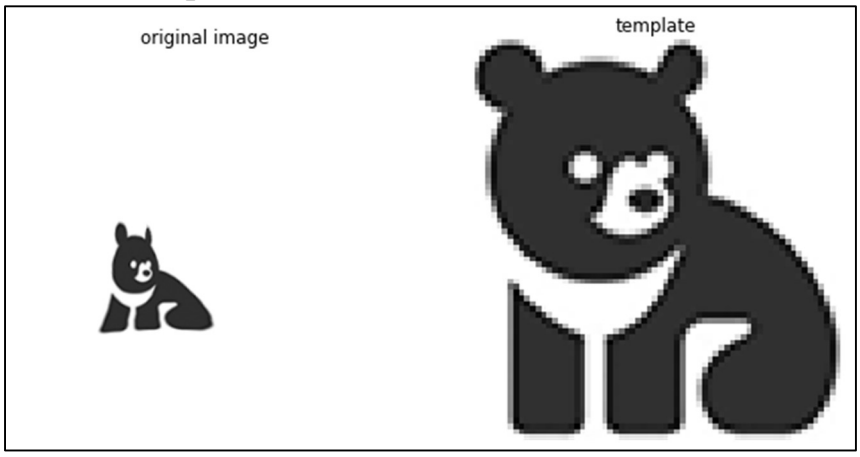

Figure 11: Example 2

- After CCOEFF:

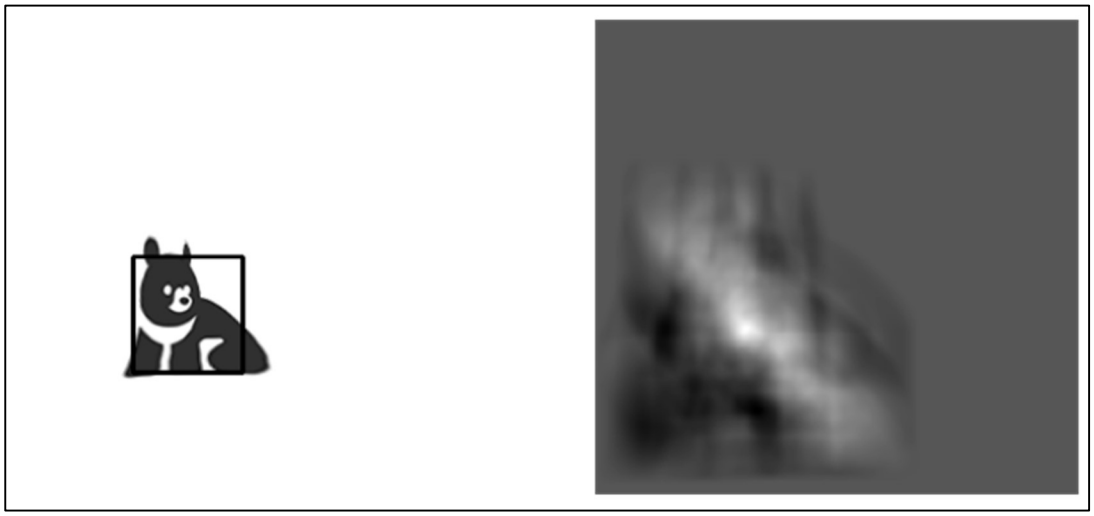

Figure 12: Example 2 result

- Once again, the bounding box still can not cover all the parts due to template's shape. And the contour graph still has the main brightness areas in its diagonal like we concluded above.

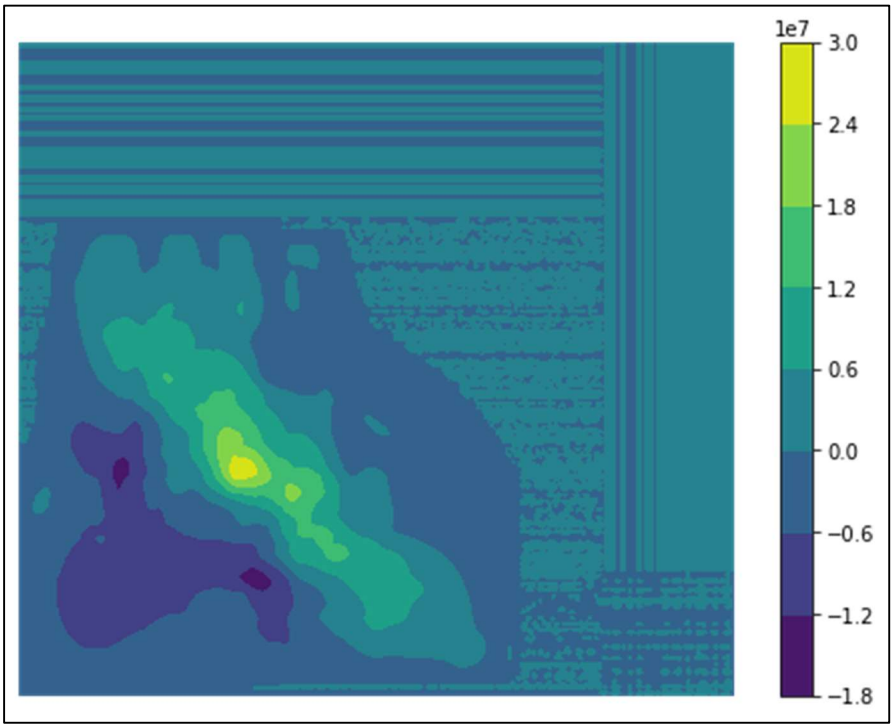

Figure 13: Example 2 convour graph

- Another example with the “super” rotated, blent and scaled teddy-bear:

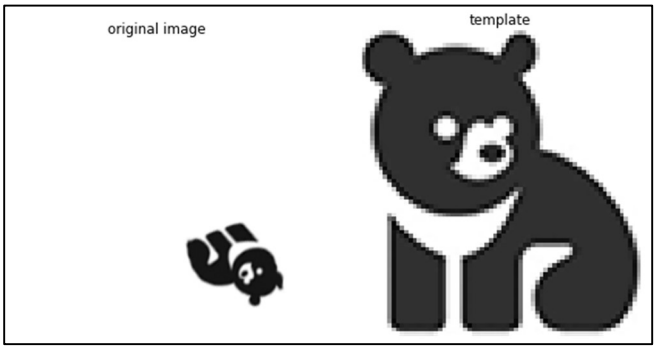

Figure 14: Example 3

- The result:

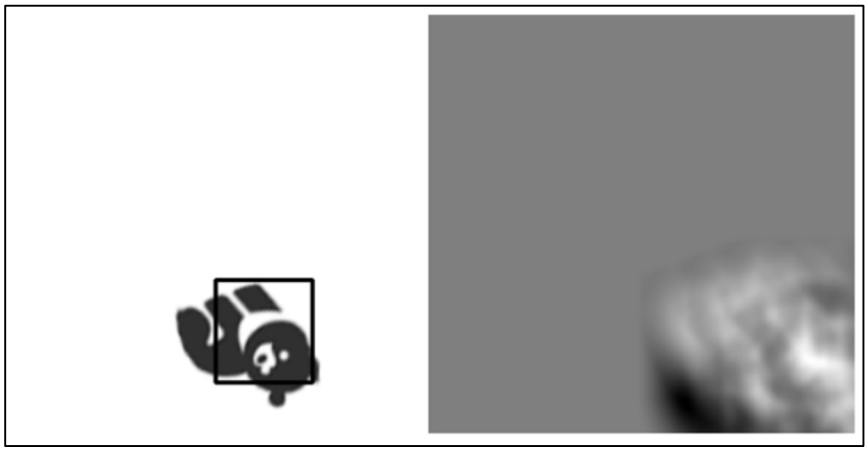

Figure 15: Example 3 result

3. CCOEFF TESTING WITH REALIFE EXAMPLE

- Template matching testing with flower image:

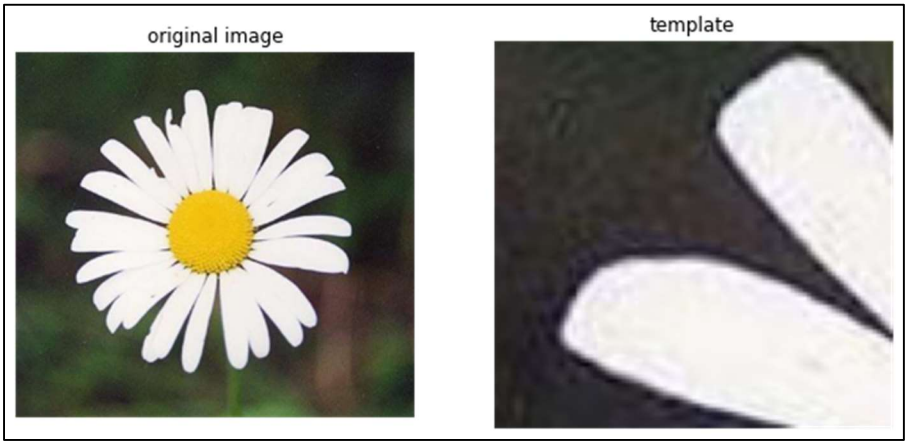

Figure 16: Real example

- The template was little bent by me *(seem unnoticeable)*, but after go through CCOEFF, the return result is perfect.

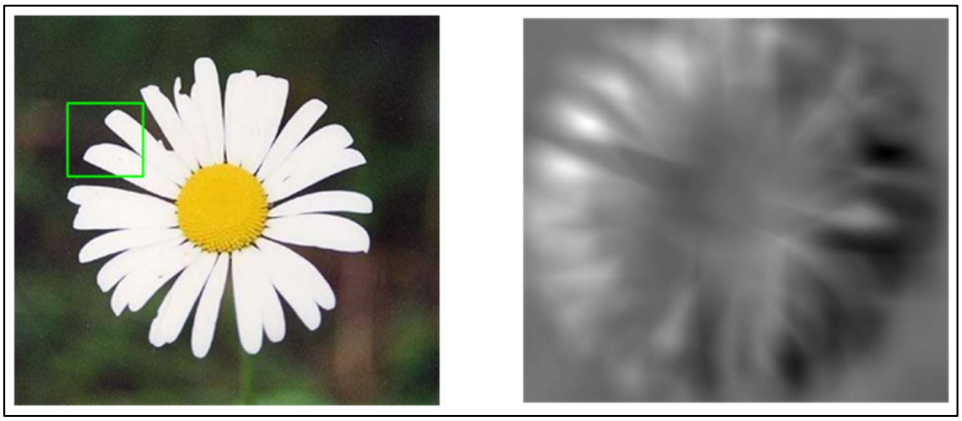

Figure 17: Real example result

- As I mentioned before, the template matching is all about finding part which is unknown that might look similar to the already known template. To demo that function, we have some picture in NBA Final 2019 of Golden State Warrios (it's Stephen Curry to be exact):

Figure 18: NBA Final 2019

- Pretend that we only have the first picture and now we want to detect whether have the Golden State Warrios'slogo or not.

Figure 19: Template is GSW's logo

- After went through CCOEFF, we have these results:

Figure 20: Detecting logo result

- The second picture was not good, the bounding box wasn't in its place. In this, I just realize that I only took the gray channel of image go through CCOEFF, so I started wonder what if I take all the 3 color channels R, G, B to go through the operate?

Figure 21: Detecting logo using 3 channels

- It's shown that 3 color channels are working better, it might be because the logo is yellow, so the green and red channels might higher than the blue, that means it's easier than if we only have 1 coler channel.

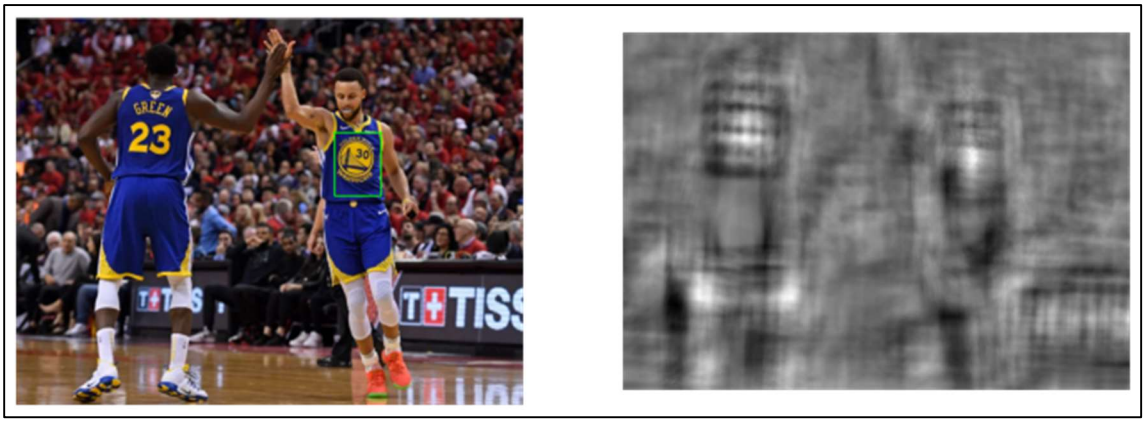

Figure 22: Detecting logo using 3 channels

- This example led me to think what if I can analyst the color of the template, them boost the channels which present the template better, then the difference between the right and wrong spot might be clearly to detect. *(However, I still working on how to do that)*

```
if len(img.shape) == 3:
    print("3 channels image")
    b, r, g = cv2.split(img)
    temp_b, temp_r, temp_g = cv2.split(template)
    res_b = cv2.matchTemplate(b, temp_b, method)
    res_r = cv2.matchTemplate(r, temp_r, method)
    res_g = cv2.matchTemplate(g, temp_g, method)
    res = np.add((res_b), (res_r), (res_g))
```

- In the last example:

Figure 23: Where is my logo?

- It’s a mystery why the brightest point is in under the player’s knee. I still can not find the answer of this example. Maybe it’s the CCOEFF limit, the logo in this picture is clear, but the picture itself might be complex, there’re a lot of bright area in the original picture which also bright in result after CCOEFF (the right image), the logo also bright, but is not brighter than these selected areas.

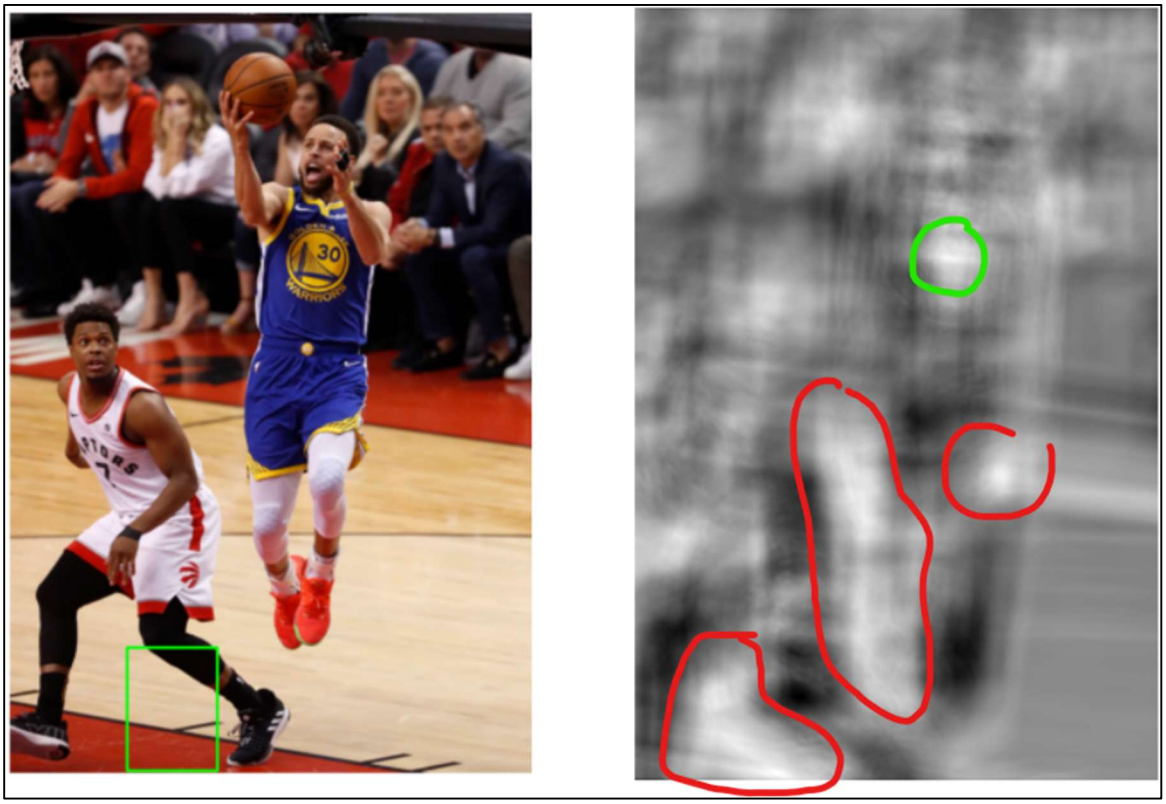

Figure 24: Brightness areas

- I’m still working on to find the answer to this picture. This is the end of my report about template matching with disortion.
- Thank you for reading.

4. REFERENT

Wikipedia:

- *https://en.wikipedia.org/wiki/Template_matching#:~:text=Template%20matching%20is%20a%20technique,to%20detect%20edges%20in%20images*.
- *https://en.wikipedia.org/wiki/Cross-correlation#:~:text=Normalized%20correlation%20is%20one%20of,Pearson%20product%2Dmoment%20correlation%20coefficient*.

OpenCV documents:

- *https://docs.opencv.org/master/df/dfb/group__imgproc__object.html#gga3a7850640f1fe1f58fe91a2d7583695da5be00b45a4d99b5e42625b4400bfde65*

Author's note:

I didn't read any paper at all. I think I might be not confident with academic paper 😯. *Thoes paper just look terrified and it seems like took a month to understand it* ☹. *So that I only read the equation then try to understand and show it's result when apply to different situations.*

English is not on my priority list, however I too tired to translate academic word to Vietnamese, so I decied to write my report by English. There will be a lot of mistake in this report and I really appriciate if you can ignore it. Thank you and have a good day. 😆