

REFLECTION

The reflection should explain the strategies you used to detect the **presence** and **theft** of the Red Panther (Red ruby) and how you **implemented** these strategies in code. The reflection should also discuss **how well your system worked** and **suggest improvements**.

In the core and completion parts, I use a strategy that checks if each image contains red pixels of the red ruby or not, which is the redness value that I recalculate by using $\text{red pixel value} / (\text{green pixel value} + \text{blue pixel value})$. I got the idea that I wanted to find the value of the redness compared to other colours because it will get more redness than thinking with the equation, $\text{red pixel value} / (\text{red pixel value} + \text{green pixel value} + \text{blue pixel value})$. There is more redness value of red colour to distinguish red and other colours well. Then I found the red value of the red ruby in each image, there is always a redness value more than 1 in both core and completion part. Therefore, I consider every pixel in the image and then store the highest redness value in the image then determine if it is greater than 1 or not. If it is greater than 1, it will be considered as a presence of red ruby (Ruby present), if not, it will be considered as theft (Ruby was stolen) along with a sound alert.

The challenge part is where there are movements of the red ruby and then secretly swap from the real red ruby to the fake red ruby into another part of the screen, which rises me to the idea of detecting which pixel was red, and after that find the two pixels that are farthest by Euclidean distance, how far apart are these two pixels because if the red ruby is swapped, the distance will be much higher, so it reveals that the red ruby has been swapped and also the shape of the red ruby changes, which leads to the changes of the distance as well. From the test, it was revealed that the distance value of the real red ruby is less than 5000, so the theft can be detected easily.

However, there might be a problem if there is a new fake red ruby that has a maxdistance that is less than 5000, which may cause some confusion between fake and real red ruby. Although, during the process of swapping, it is still possible to tell whether the theft has occurred or not, but if the swap has been completed, there may be a problem in distinguishing between the real red ruby and the fake red ruby. It can be improved by finding the maxdistance of the actual red ruby and storing it, so we can check whether the found red ruby is real or fake, for example, the range of the real red ruby is between 4650-4850. So, outside of that range, it is a fake red ruby.