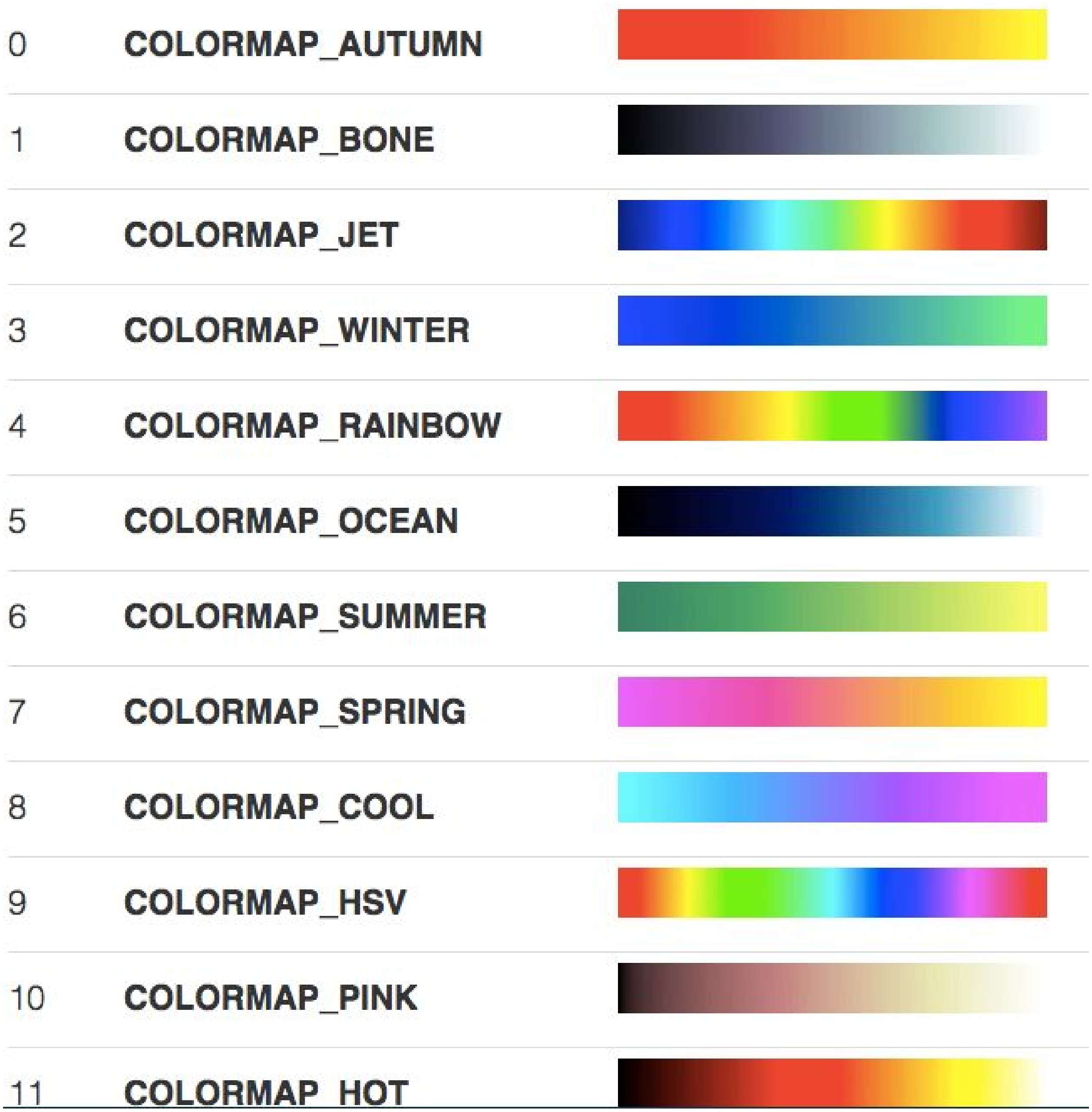
Here, the gray scale image of a dandelion was given. Seeing the image, it can be understood that there are two major sections, which is grass and the flower. Even in the grass the there is the texture of blades that is quite distinctly visible.

There are some inbuilt functions in OpenCV Module that allow us to colour gray scale images. Looking up the colormaps online, 4 colormaps (namely Jet, HSV, Bone and Summer) were tried.

These work by replacing lower grayscale values with colours to the left of the spectrum and higher ones with the colors to the right end of the spectrum.

Considered color map:

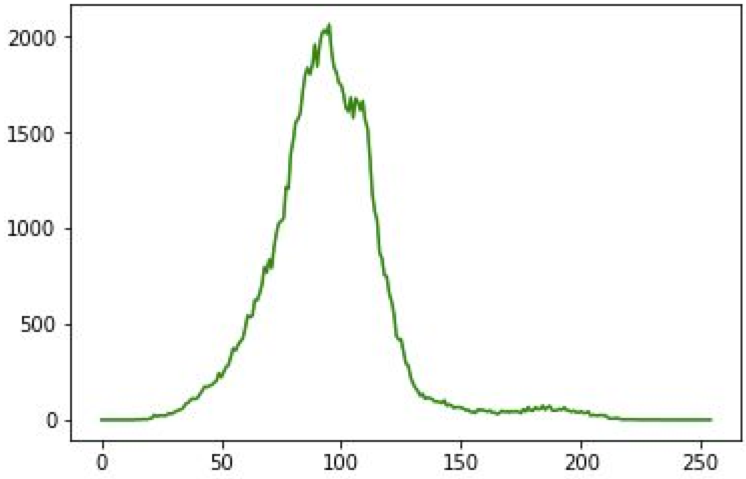


0	COLORMAP_AUTUMN
1	COLORMAP_BONE
2	COLORMAP_JET
3	COLORMAP_WINTER
4	COLORMAP_RAINBOW
5	COLORMAP_OCEAN
6	COLORMAP_SUMMER
7	COLORMAP_SPRING
8	COLORMAP_COOL
9	COLORMAP_HSV
10	COLORMAP_PINK
11	COLORMAP_HOT

HSV and Jet gave results that look far from what we see in real life, however, the distinction between grass blades was better represented in HSV. Coming to the other two, even though Bone gives an image with more clarity, the colouring isn’t exactly right and summer has the same problem.

So, an own colormap has been designed. As we can see, majority of the image is covered in grass. When a histogram is plotted for the image, a majority of the pixels have intensity values lying between 50 and 150. So, it can be assumed that these values represent the grass.

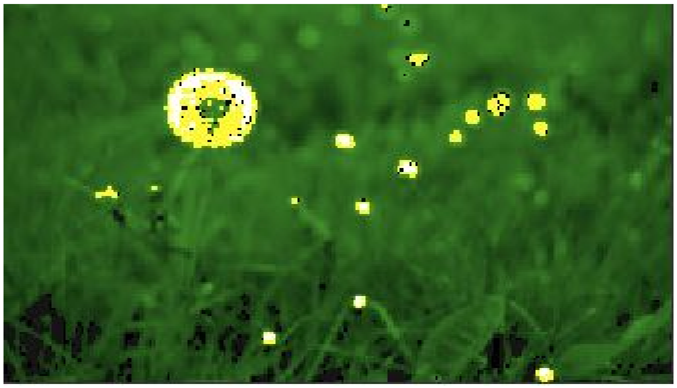
The histogram:



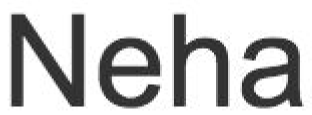
In order to color the image, the image is divided into 4 regions based on its intensity:

1. Intensity between 0 and 50
2. Intensity between 51 and 150
3. Intensity between 151 and 200
4. Intensity between 201 and 255

And based on these ranges, each pixel was assigned the corresponding RGB values, resulting in the colored version.



Now, we are required to embed our name into the coloured image secretly.
First, the image of the name was taken:


Neha

Then, a technique called “Bit Plane Slicing” is employed. Bit plane slicing is a method of representing an image with one or more bits of the byte used for each pixel. As we know, the gray level of each pixel of a digital image is stored in the form of byte(s) in the system. We are dealing with 8 bit numbers here (represents numbers in the range 0-255). Now, a change in the MSB(Most Significant Bit) alters the value encoded by that byte significantly when compared to the LSB(Least Significant Bit). So, the LSB is modified so that the impact is minimal and our task of embedding it secretly is accomplished.

To embed this, it first has to be brought to the same size and hence, is resized. After that, each pixel is converted into its corresponding bitArray. Then, as to assign weights, the corresponding bit is multiplied with its weight. After the name image is converted to Binary, the last bit of the combined image is modified.

The final image: