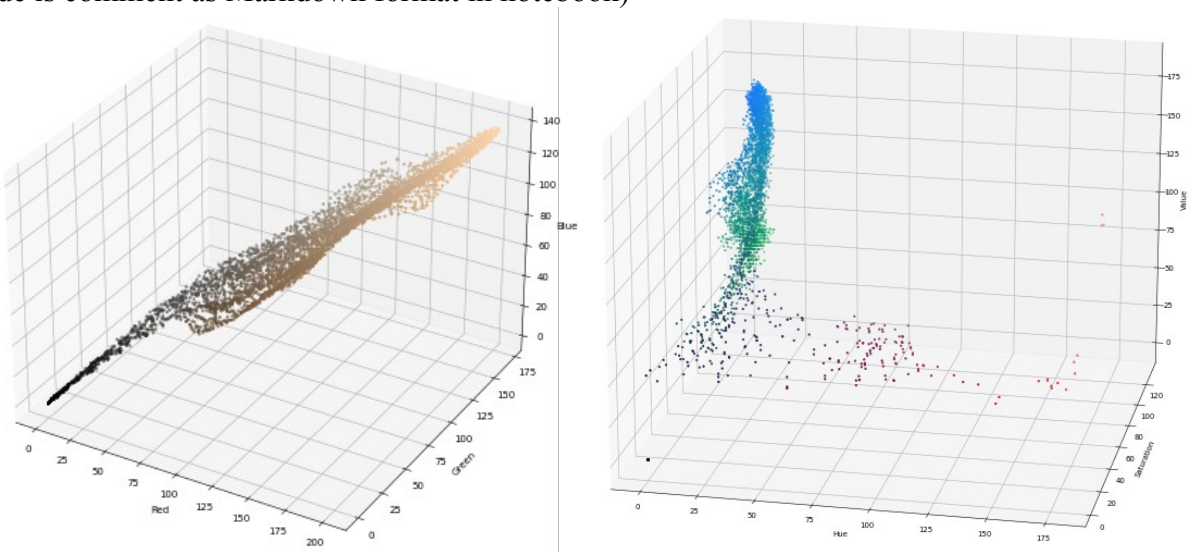
Nebula Innovation Pvt. Lim.
Assignment
Submitted by: Shobhit Sharma
Email ID: sharma.38@iitj.ac.in

Programming Language Used: **Python 3** (python 3.8.3)
Used **Libraries** are:

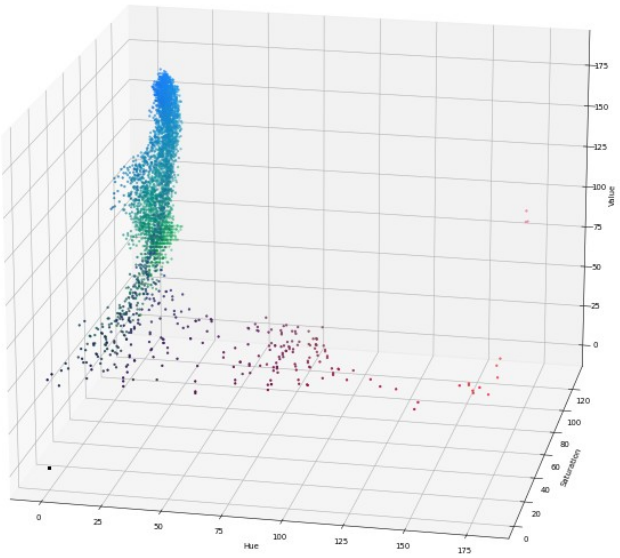
library	**Known as**	**version**
OpenCV	**cv2**	4.5.1
NumPy	**numpy**	1.18.5
Operating System	**os**	
matplotlib	**matplotlib**	3.2.2
mpl_toolkits	**mpl_toolkits**	

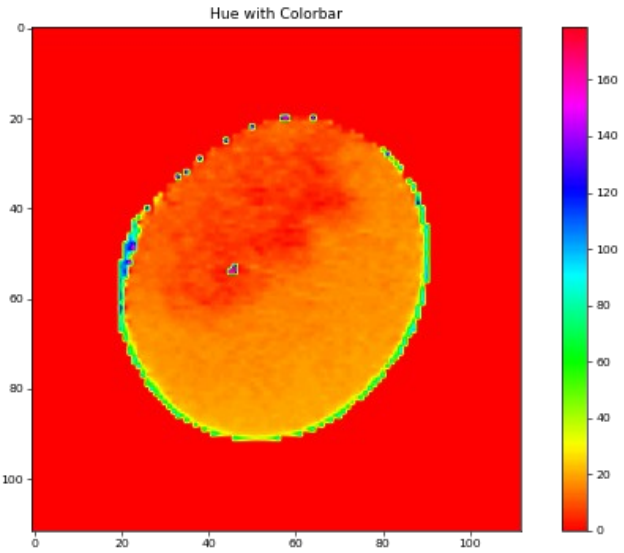
Dev. Environment: Jupyter notebook (inbuilt with Anaconda 3).
System Arch.: Apple with macos.
Processor: 1.8 Ghz Dual-Core I5.
Memory: 8 GB.

Q1
Task: Classify the grains as either diseased or non-diseased.
Initial steps:
a. **Read** Image by: cv2.imread()
b. **Analyse** RGB and HSV color space for image segmentation, particularly detecting purple patch. (Code is comment as Markdown format in notebook)

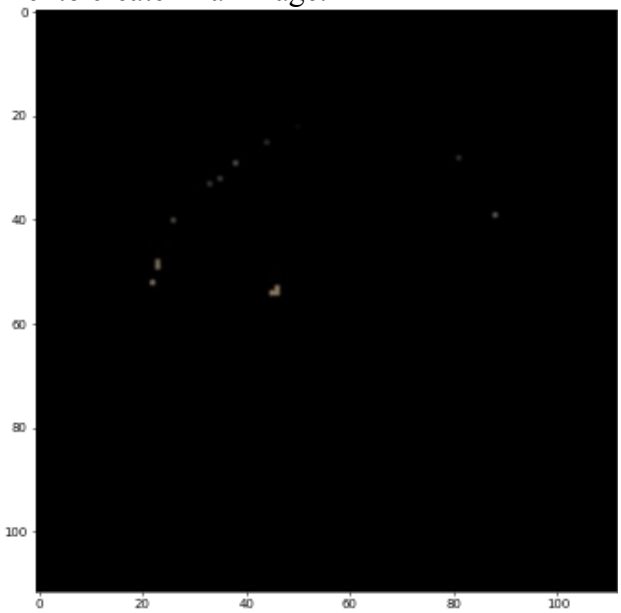


Approach:
a. After analysing RGB and HSV color space, HSV found suitable for image segmentation. Therefore, further processing in done in HSV color space.
b. **Selecting** the appropriate **bright and dark** color **ranges** in HSV for purple pixel by observing 3D mapping of pixel distribution in HSV and Hue colorbar. Therefore, bright range for Hue = 130 and dark range for Hue = 200. Hence, color range for Hue is [170, 200] is selected.





c. Creating **mask** for selected Hue range.
d. **Extract** R, G and B channel based on previously created mask.
e. **Merge** R, G and B channel to create final image.



f. **Count** non-zero (i.e. non-background in our case) **pixels** in final image and if the count is greater than 50 then classify this image (of grain) as Diseased otherwise classify as non-diseased.

Q2

Task: To remove the black shadow in the background using adaptive new adaptive threshold algorithm.

Initial Steps:

1. Considered gaussian filter for incorporating neighbours for local threshold or adaptive threshold.

Filter: [[1, 4, 6, 4, 1],
[4, 16, 24, 16, 4],
[6, 24, 36, 24, 6],
[4, 16, 24, 16, 4],
[1, 4, 6, 4, 1]]

Normalised Filter: obtained by dividing by 256 (i.e. weight of filter)

[[0.00390625, 0.015625 , 0.0234375 , 0.015625 , 0.00390625],
[0.015625 , 0.0625 , 0.09375 , 0.0625 , 0.015625],
[0.0234375 , 0.09375 , 0.140625 , 0.09375 , 0.0234375],
[0.015625 , 0.0625 , 0.09375 , 0.0625 , 0.015625],
[0.00390625, 0.015625 , 0.0234375 , 0.015625 , 0.00390625]]

2.Read image as RGB.

Approach:

1. **Padding** the image with **zeros** in boundaries using formulae:

$$P = (F-1)/2$$

2. **Loop over** each pixel and for each **pixel check** the **R** channel value is **less than 80** and is non zero then it means we need thresholding. In other words we need to suppress the **R** channel value or keep it as it is.

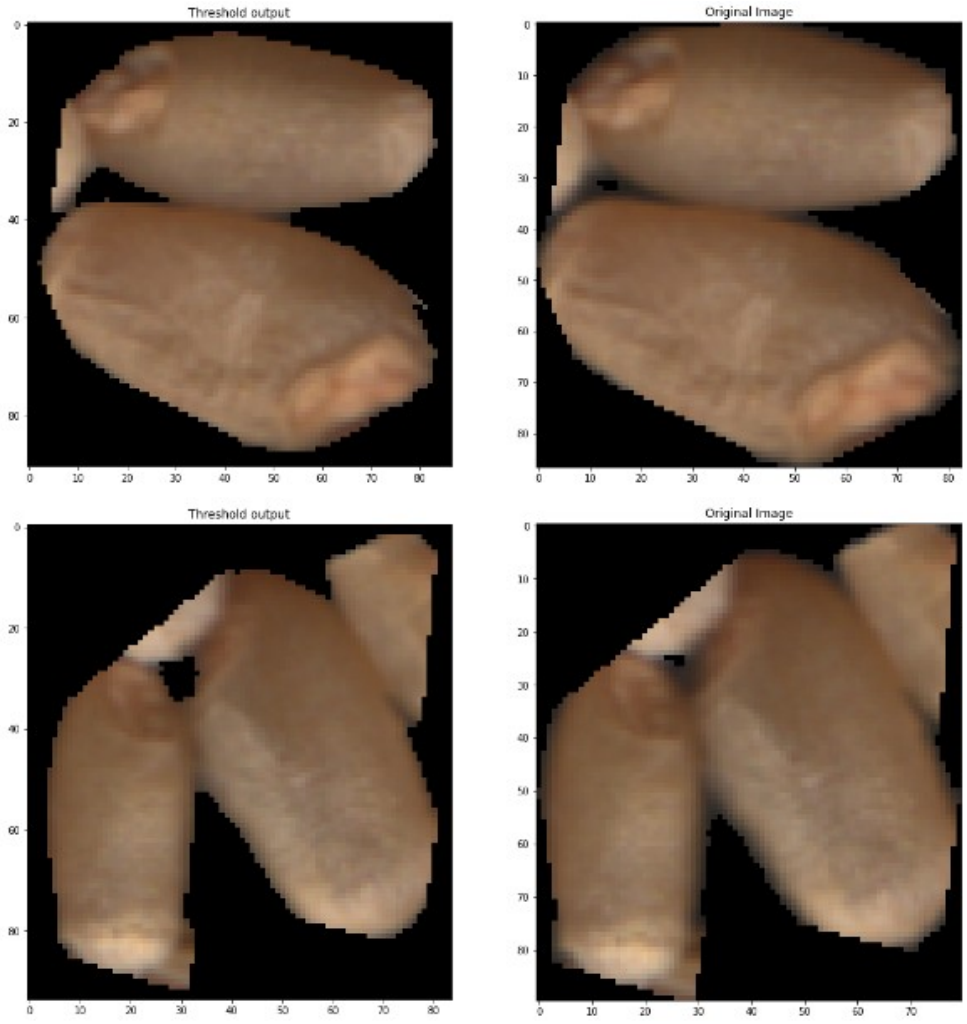
2.1. For thresholding gaussian filter (as mentioned above) is used for adaptation.To do so,convolve gaussian filter with corresponding image window and then subtract the '**C**' constant. After performing this step we have obtained threshold value **Th**.

[Note: Here **C** constant can be varied to control the threshold value.]

2.2 Compare the **Th** value with corresponding pixel value in **R**ed channel and if the threshold **Th** value is less than the Red channel value of that pixel then **set** the pixel value **to zero** in each channel **otherwise** keep it the **same**.

2.3 We can vary the '**C**' parameter to fine tune the output. I have used **C** = 20.

Results:



Conclusion: Different Gaussian filters of different size can be used to improve.