

Task 2 :BlurOrNot

Assumptions:- Input image should have minimal (600X400) size for the proper prediction performance. It will work on smaller size images also but accuracy may get compromised.

Run from terminal:- import BlurOrNot as bn

bn.BlurOrNot("file_path.file_extension")

E.g:- bn.BlurOrNot("/home/sanchit/Desktop/TA/testing_blur/2.jpg")

Logic behind the function

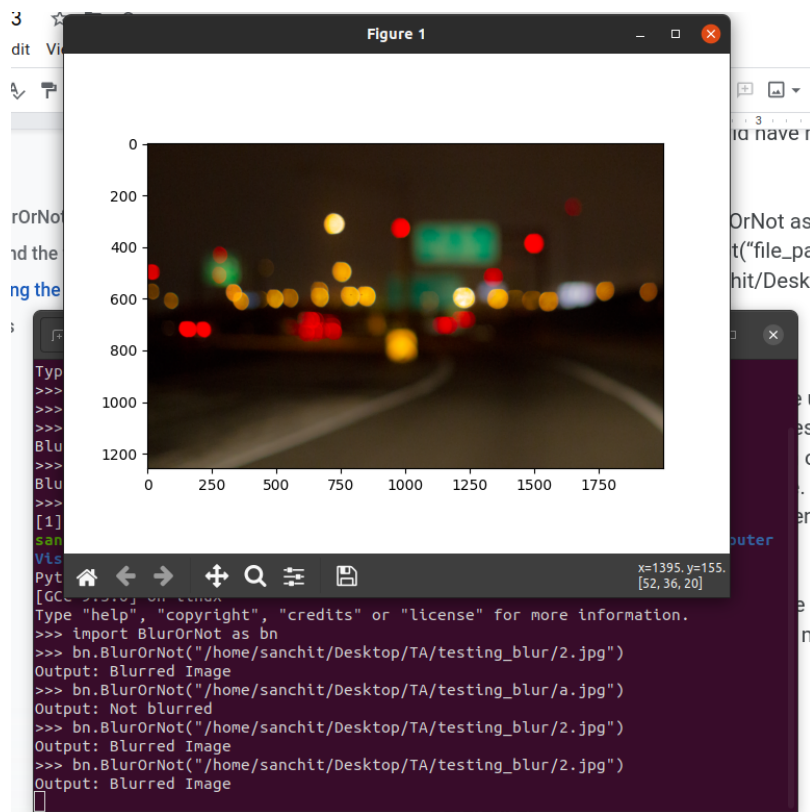
Inspired by results from paper[1], I have used the pre-existing skimage's laplacian algorithm which shows the best **recall** value on blur vs Non-blurry image. Furthermore, I have **calculated the variance of the laplacian** of the image.

- Blurry image= less variance value.
- A proper image will be sharper, hence high variance will be there

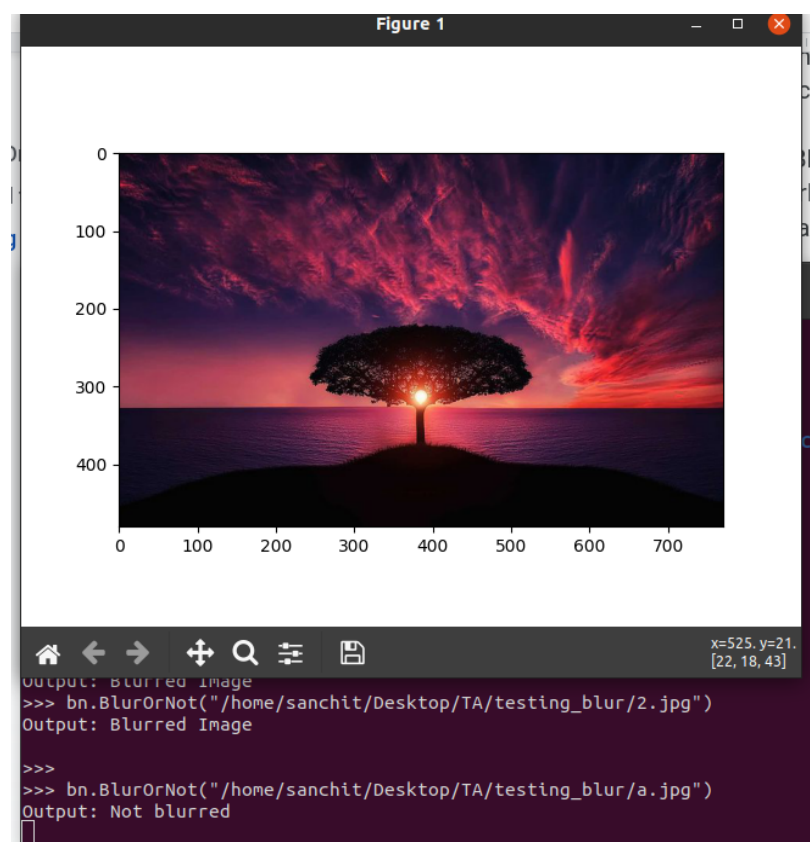
Thresholding the variance

For proper prediction, I tested the kaggle dataset "blur dataset"[2] which consist of **sharp images, focussed and motion blurred** images to analyse the proper threshold value, which comes out to be **0.003** (You can check my rough coding work available [here](#). It will show how I got this threshold value).

Output Screenshots



Output: Blurred Image



O/p: Non-Blurry image

References

1. iBlurDetect: Image Blur Detection Techniques Assessment and Evaluation Study by Roxanne A. Pagaduan, Ma. Christina R. Aragon and Ruji P. Medina
 2. Blur dataset available at
<https://www.kaggle.com/kwentar/blur-dataset/version/2>
-