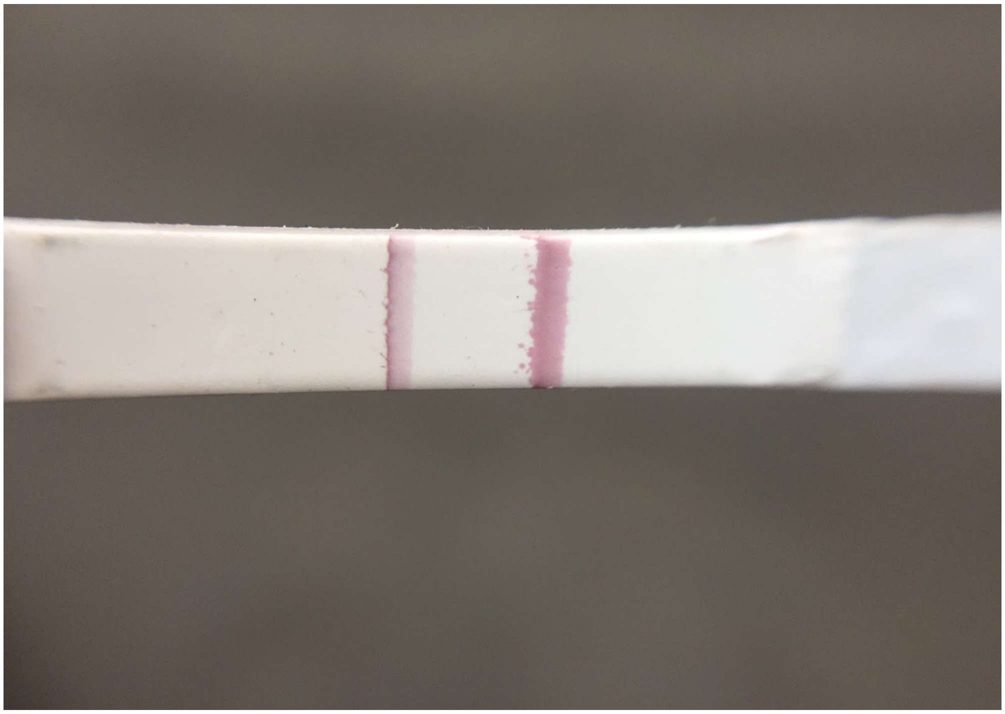
CROPPING BACTERIAL TEST STRIPS FROM IMAGES



ANJANA YADAV

Date of Assignment: 17.05.2018

Date of Submission: 19.05.2018

INTRODUCTION

The given task is to crop the bacterial strips(white part including purple stains), from the given image using an automated algorithm. The task can be divided into three steps as discussed below:

1. Image Preprocessing
2. K-Means Clustering
3. Image Cropping

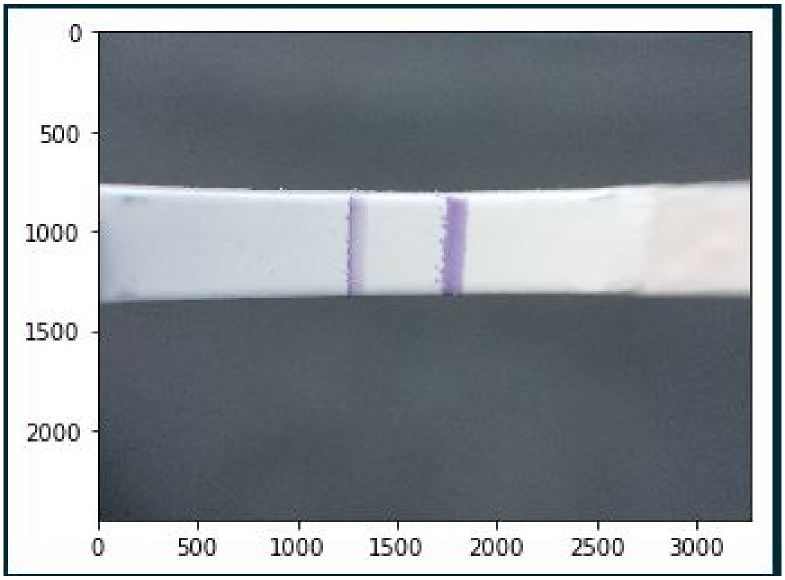
PACKAGES USED

1. OpenCV
2. Numpy
3. Matplotlib
4. Sklearn
5. PIL

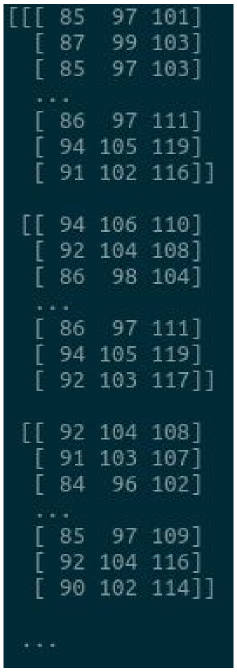
INPUT:

The input image taken here is IMG_3082.JPG.

(The images attached as screenshots are taken using the matplotlib package pyplot function)



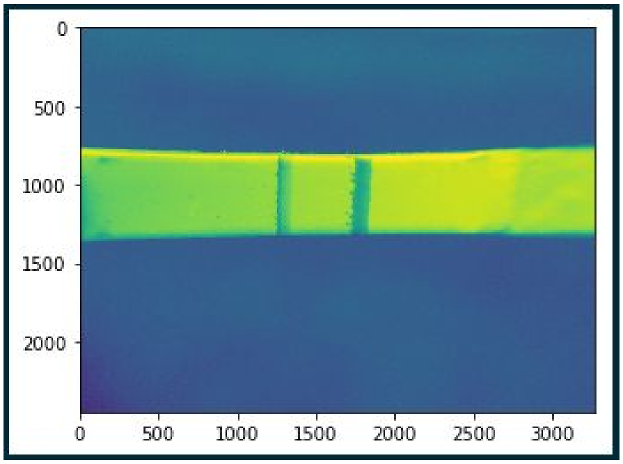
The pixel matrix of the coloured image looks like below. We can see that each point is a vector of three values of RGB:



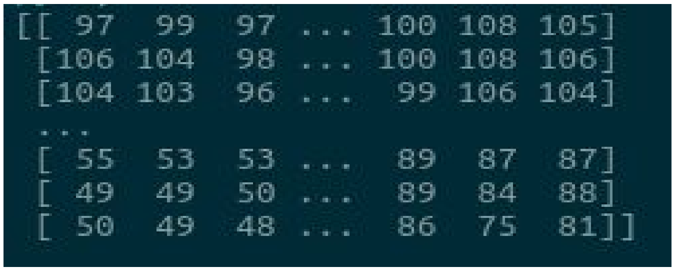
```
[[[ 85  97 101]
  [ 87  99 103]
  [ 85  97 103]
  ...
  [ 86  97 111]
  [ 94 105 119]
  [ 91 102 116]]

 [[ 94 106 110]
  [ 92 104 108]
  [ 86  98 104]
  ...
  [ 86  97 111]
  [ 94 105 119]
  [ 92 103 117]]

 [[ 92 104 108]
  [ 91 103 107]
  [ 84  96 102]
  ...
  [ 85  97 109]
  [ 92 104 116]
  [ 90 102 114]]

 ...
```

IMAGE PREPROCESSING:

The input image is converted into the grayscale image. This is done as the K-Means clustering algorithm requires vector points of dimension <=2 only. The coloured image has points of vector size 3 ie. RGB pixels. Converting the image to Grayscale will create each point vector of singular value between 0-255. The gray scale image is:



The gray scale matrix looks like below. The matrix consists of 2448 (number of rows in image pixel matrix) vectors each of length 3264 (number of columns in image pixel matrix). Thus each point is single valued.

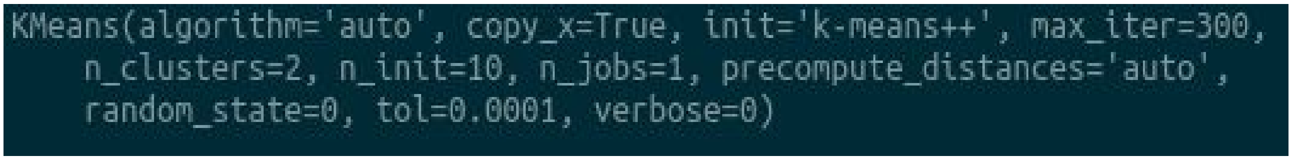


```
[[ 97  99  97 ... 100 108 105]
 [106 104  98 ... 100 108 106]
 [104 103  96 ...  99 106 104]
 ...
 [ 55  53  53 ...  89  87  87]
 [ 49  49  50 ...  89  84  88]
 [ 50  49  48 ...  86  75  81]]
```

K-MEANS CLUSTERING:

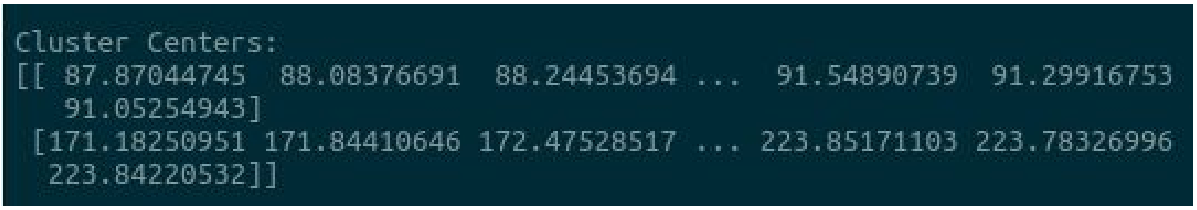
In order to crop the image, we require the matrix x-y coordinates of where the image strip starts and ends. K-Means Clustering is an unsupervised learning method that divides the vector points into clusters such that inter-cluster distance is maximised and intra-cluster distance is minimised.

In the input image, the strip region is the first cluster and the background Black region is the second cluster. In the grayscale image, the strip region pixels are closer to white colour ie. 255, whereas the background region is closer to black ie. 0, on the scale of 0-255. Thus these tend to form two different clusters by K-Means algorithm. The Scikit Learn Package provides an inbuilt function of K-Means.



```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=2, n_init=10, n_jobs=1, precompute_distances='auto',
    random_state=0, tol=0.0001, verbose=0)
```

The grayscale matrix is passed as input into the function. After fitting two different clusters are formed with cluster mean values as follows:



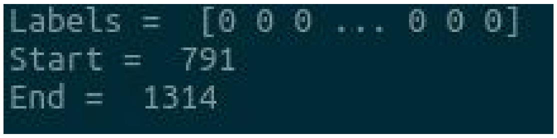
```
Cluster Centers:
[[ 87.87044745  88.08376691  88.24453694 ...  91.54890739  91.29916753
   91.05254943]
 [171.18250951 171.84410646 172.47528517 ... 223.85171103 223.78326996
  223.84220532]]
```

The K-Means algorithm measures the distance of every pair of pixels in the gray matrix . The pixels of smaller distance are placed in the same cluster. The algorithm runs 300 times, each time upgrading the clusters. Absolute distance measure metric is taken in order to find the distance between two pixels. Finally the result is two clusters.

IMAGE CROPPING:

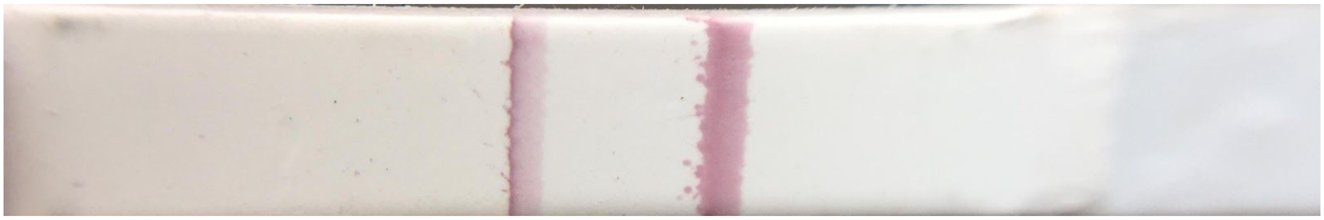
The K-Means clustering algorithm gives the cluster labels which is an array of 1's and 0's stating which cluster the point belongs to. Since the images has horizontal strips, the clustering is done row pixel wise.

In the cropped function, the starting and ending points of the array labels are found. These points are the start and end row values of the white strip region.



```
Labels =  [0 0 0 ... 0 0 0]
Start =  791
End =  1314
```

The PIL package has an inbuilt function to crop the image if the start and end coordinates are known. The cropped image is finally saved in the 'cropp.jpg' file.



ADVANTAGE OF K-MEANS:

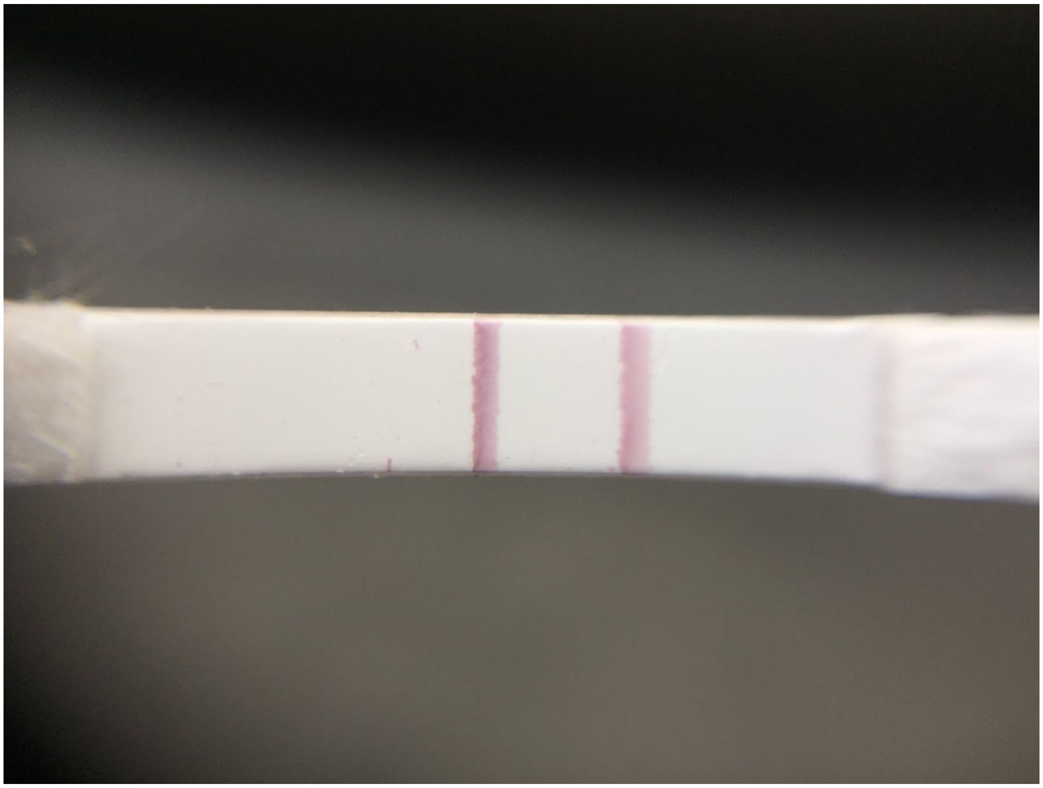
The benefit of using K-Means in order to cluster is that K_Means is not affected by light gradient as long as there is not much difference. Since we take only two clusters, K-Means successfully differentiates between the pixels of large variation. We couldn't take the coloured RGB values of Image for clustering as, then it would cluster them on the basis of RGB variations. Here we can clearly see that the background is greatly different from the white strip which acts as a feature of the image.

The K value is already known as two clusters. Also K-Means provides a faster computation in comparison to other algorithms in this case.

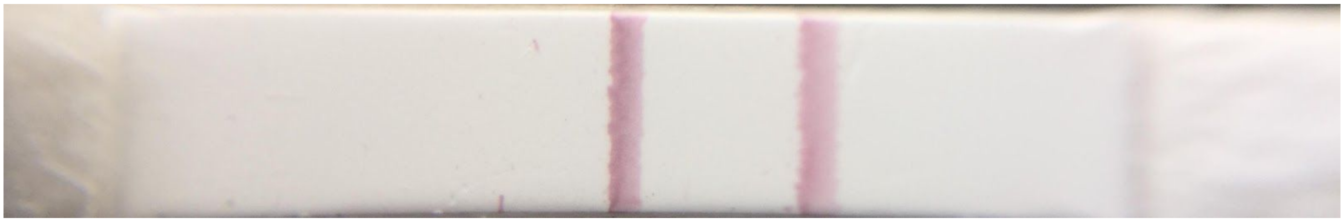
RESULTS

Screenshots of one image from each of the folders are as follows:

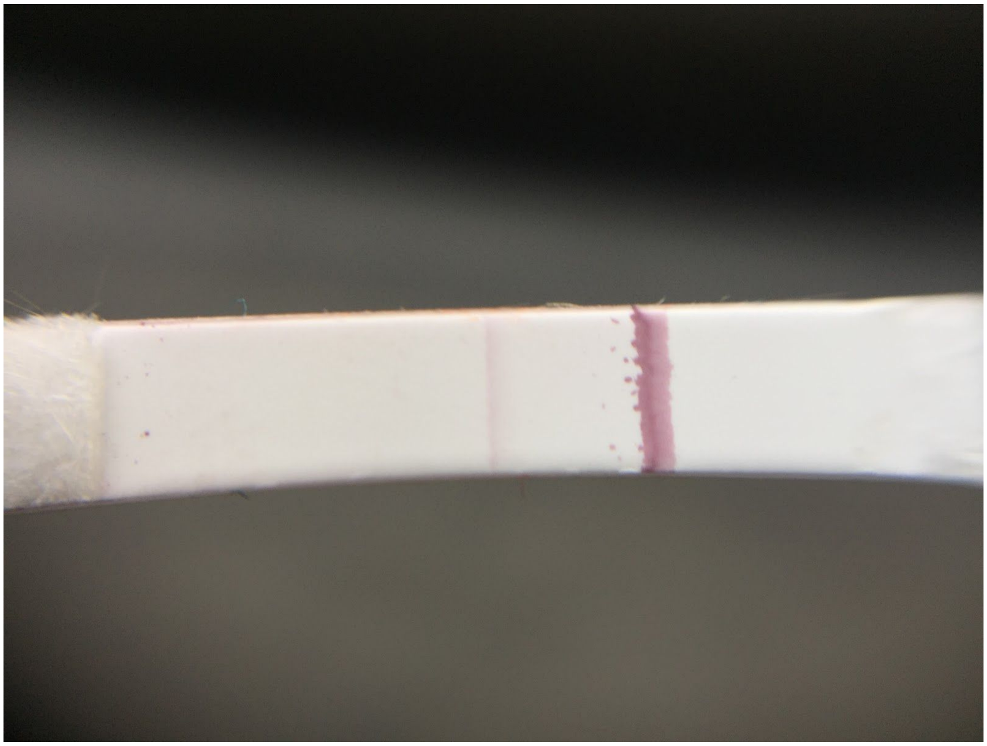
Input: IMG_3086.JPG



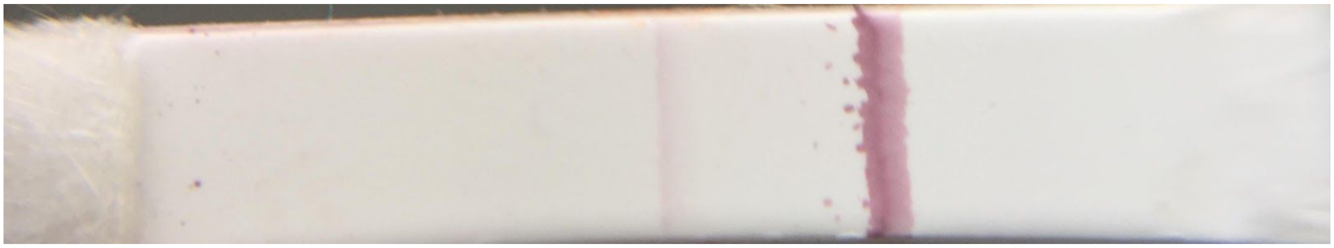
Output: Cropped Image



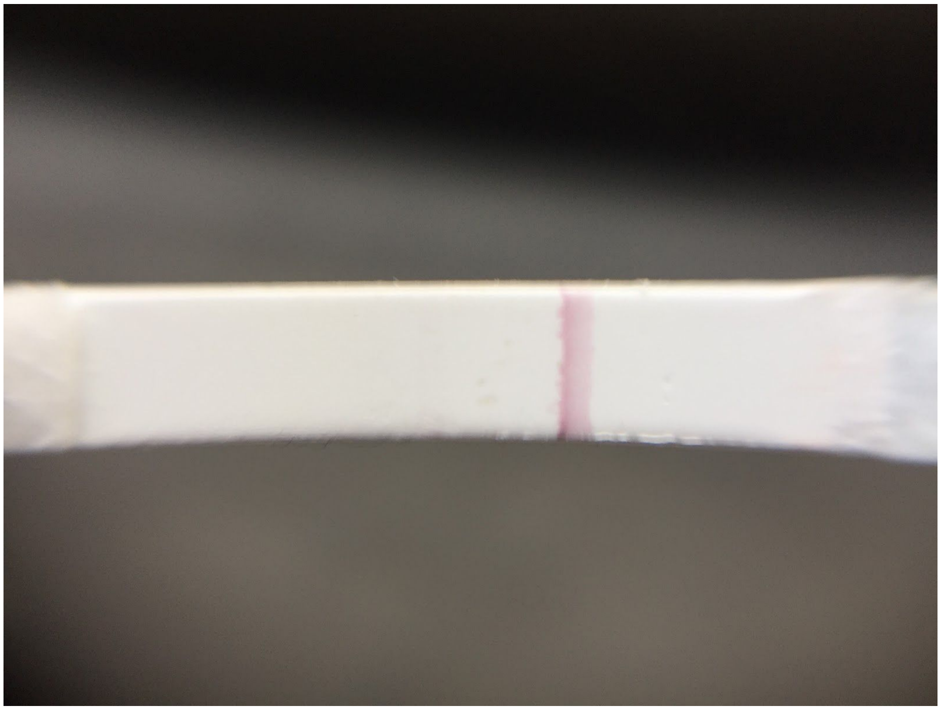
Input: IMG_3112.JPG



Output: Cropped Image



Input: IMG_3158.JPG



Output: Cropped Image

CONCLUSION

The algorithm is able to give accurate cropped image of the strips in less time. Concepts of both Image processing and Machine Learning are thus put into application.

REFERENCES

1. http://matthiaseisen.com/pp/patterns/p0202/
2. http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
3. Artificial Intelligence book by Peter Norvig