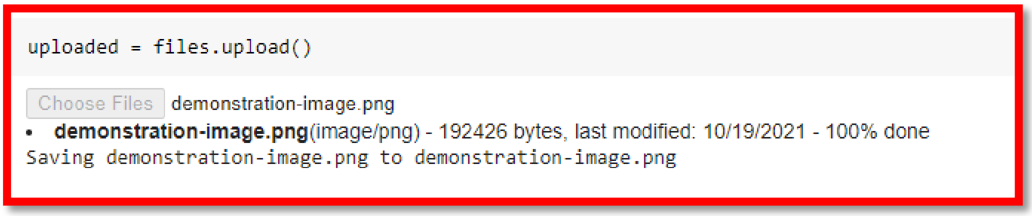
Harsh Kumar Sharma
RA1911027010082
CSE – BD (N2)

Image Segmentation (Machine Learning Model)

Source Code and Outputs:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from google.colab import files
uploaded = files.upload()
```



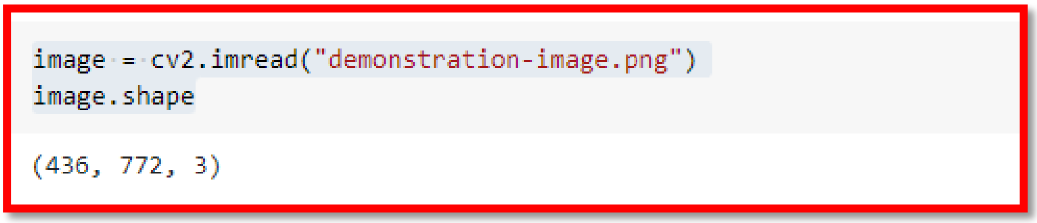
```
uploaded = files.upload()
```

demonstration-image.png

- **demonstration-image.png**(image/png) - 192426 bytes, last modified: 10/19/2021 - 100% done

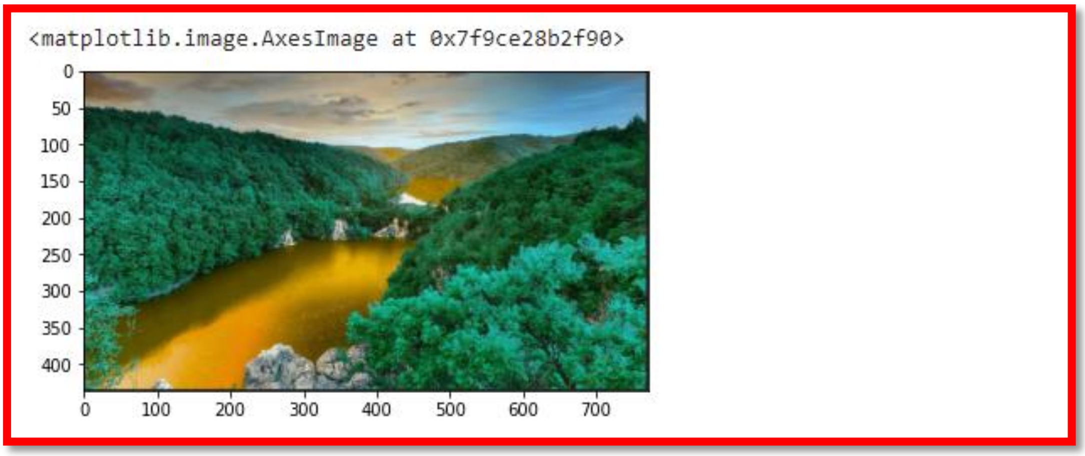
```
Saving demonstration-image.png to demonstration-image.png
```

```
image = cv2.imread("demonstration-image.png")
image.shape
```



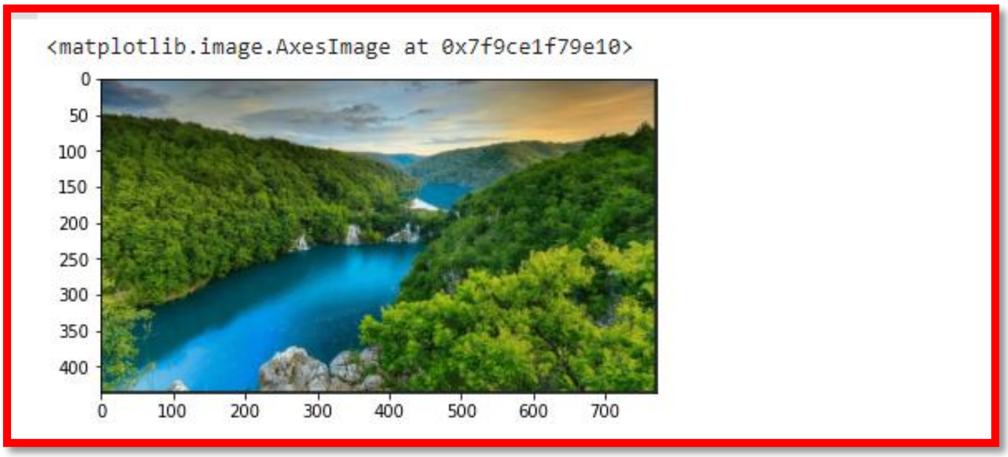
```
image = cv2.imread("demonstration-image.png")
image.shape

(436, 772, 3)
```

```
x,y,z = image.shape
plt.imshow(image)
```



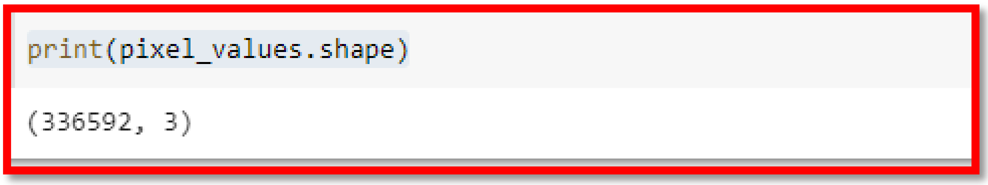
```
<matplotlib.image.AxesImage at 0x7f9ce28b2f90>
```

```
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
plt.imshow(image)
```



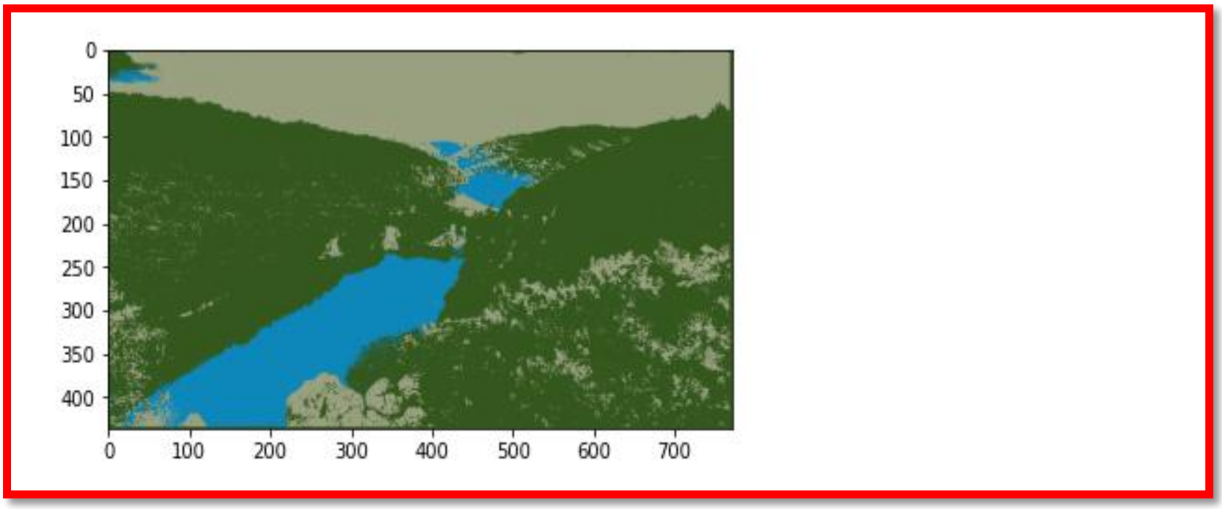
```
<matplotlib.image.AxesImage at 0x7f9ce1f79e10>
```

```
pixel_values = image.reshape((-1, 3))
pixel_values = np.float32(pixel_values)
print(pixel_values.shape)
```



```
print(pixel_values.shape)

(336592, 3)
```

```
criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 100, 0.
2)
k = 3
_, labels, (centers) = cv2.kmeans(pixel_values, k, None, criteria, 10,
cv2.KMEANS_RANDOM_CENTERS)
centers = np.uint8(centers)
labels = labels.flatten()
segmented_image = centers[labels.flatten()]
segmented_image = segmented_image.reshape(image.shape)
plt.imshow(segmented_image)
plt.show()
```



```
masked_image = np.copy(image)
masked_image = masked_image.reshape((-1, 3))
cluster = 2
masked_image[labels == cluster] = [0, 0, 0]
masked_image = masked_image.reshape(image.shape)
plt.imshow(masked_image)
plt.show()
```