Team:16
Version:1
Date:08/06/2023

Experimental Validation Form

Protocol Title: Evaluation for Validating the Effectiveness of K-Means Clustering for Image Processing.
Hypothesis: Hypothesize that the K-Means Clustering technique effectively scales, adjusts, and improves the overall visualization of test strip images, by image segmentation which groups similar colors together and highlights the region of interest.
Types of Data: The Type of Data used is an image dataset and the evaluation includes RGB color values for each pixel image obtained from image processing. The data is a 2D array of pixels, where each pixel contains Red, Green, and Blue values.
Controls: The control in this EVP is the K-means clustering algorithm and the number of clusters is specified in the parameters to determine the detailing in segmentation and clustering results.
Number of Samples: Test strip images are the dataset. On each specific image K-Means clustering technique is applied. Variations do exist in the sample in terms of image size, resolution, content, and color distribution.
Brief Description of Protocol: This EVP outlines the validation process that is used for evaluating the effectiveness of the K-means clustering technique in image processing. The protocol involves loading the image on the image processing environment and then applying K-means clustering to the pixel data. Then comparing and visualizing the original uploaded image and the clustered image which is segmented and grouped into different colors to catch the nuances of the variations.
Experimental Justification:

1. K-means clustering is a very popular and widely used technique for image segmentation and clustering.
2. The potential is tremendous as it can effectively separate pixels into distinct clusters based on their color or intensity values.
3. This evaluation can provide detailed insights into the usefulness and applicability
4. Makes it easier for human interpretation or subsequent image processing tasks that rely on color-based analysis.

Experimental Limitations

1. The effectiveness of the K-Means clustering may vary on the characteristics of the image dataset such as complexity, color distribution, and noise.
2. As the number of clusters changes the segmentation results can also vary. As the number of clusters is the control and therefore it can be tricky to decide the optimal number of clusters.
3. K-means clustering may struggle with handling overlaying, blending, or complex types of clustered images.
4. It can be limited to the specific dataset as it is not generalized to all types of images.

Team:16
Version:1
Date:08/06/2023

I. **Protocol Title:** Evaluation for Validating the Effectiveness of K-Means Clustering for Image Processing.

II. **Hypothesis:** Hypothesize that the K-Means Clustering technique effectively scales, adjusts, and improves the overall visualization of test strip images, by image segmentation which groups similar colors together and highlights the region of interest.

A. **Null Hypothesis(H0):** There is no significant color change or effect on RGB values of images due to normalization.

B. **Alternative Hypothesis(Ha):** There is a significant effect on the RGB values of the image due to normalization.

III. **Materials:**

- Test strip images
- Smartphone Camera or imaging device for capturing test strip images
- Anaconda: Python programming environment
- Jupyter Notebook: Data Analysis and Visualization
- Required Python Libraries: os, cv2, numpy, matplotlib.pyplot
- Scikit-learn library for clustering algorithm

IV. **Procedure:**

A. Load and Prepare the Image Data:

1. Start by setting the working directory to the folder that has all the reaction test strip images.
2. Import the required/necessary libraries such as OpenCV, NumPy, and Matplotlib.
3. Load the images using the cv2.imread() function and store them in separate variables.
4. Now convert the image using the cv2.cvtcolor() function to RGB colorspace.
5. Now flatten the image to a 2D array using the rehspae() function.

B. Define the Number of Clusters, Apply KMeans, And Reshape

6. Set and define the number of clusters to 7(This can change from image to image).
7. Create the Kmeans() object which will take the number of clusters and the random state as 0 and fit the pixel data using fit().
8. Get the labels and clustered pixel values.
9. Reshape the original image value to the original image shape using np.unit8() to convert the elements of the array to 8-bit integers.
10. Each pixel in the clustered image will be assigned the color of its corresponding cluster

C. Visualize the Results:

11. Display the original and clustered image side by side using the matplotlib library
12. Adjust the axis labels, and ticks to match the dimension of the clustered image.
13. The Original image does not need to have axis labels and ticks and this will make it easier and give a cleaner visualization
14. Save and analyze the result or display it for further analysis

Team:16
Version:1
Date:08/06/2023

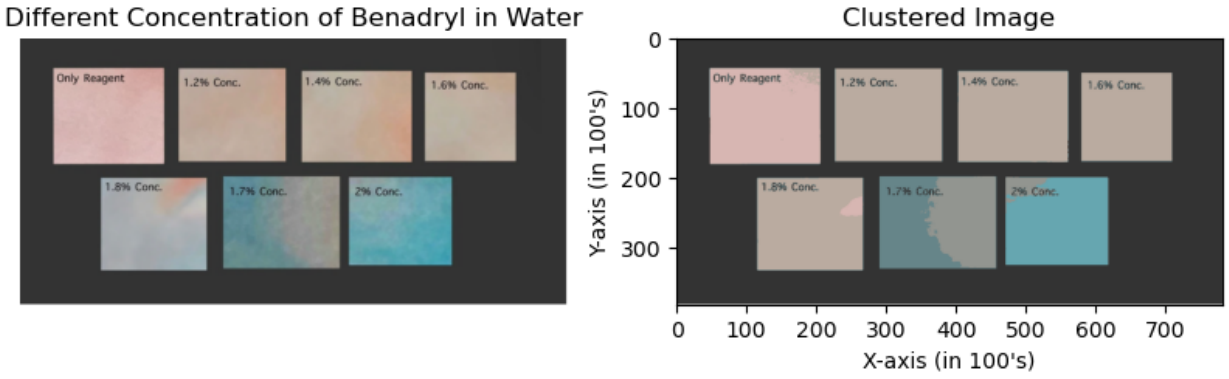
15. Compare the clustered image with the original image to assess the quality of the segmentation and clustering.

D. Example Code Snippets:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

# Load the image and convert to RGB
img = cv2.imread('Project5.png')
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

# Flatten the image to a 2D array of pixels
pixels = img.reshape(-1, 3)

# Define the number of clusters
num_clusters = 7

# Create the KMeans object and fit to the pixel data
kmeans = KMeans(n_clusters=num_clusters, random_state=0).fit(pixels)

# Get the labels and clustered pixel values
labels = kmeans.labels_
clusters = kmeans.cluster_centers_

# Reshape the clustered pixel values to the original image shape
clusters = np.uint8(clusters)
clustered_img = clusters[labels.reshape(img.shape[0], img.shape[1])]

# Show the original image and clustered image
fig, axs = plt.subplots(1, 2, figsize=(10, 5))
axs[0].imshow(img)
axs[0].set_title('Different Concentration of Benadryl in Water')

axs[1].imshow(clustered_img)
axs[1].set_title('Clustered Image')
```

Team:16
Version:1
Date:08/06/2023

```
# Convert the image to RGB color space
img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

# Get the image dimensions
height, width, _ = img_rgb.shape

# Set the tick positions and labels for the x-axis
x_ticks = np.arange(0, width, 100)
x_tick_labels = np.arange(0, width, 100)
axs[1].set_xticks(x_ticks)
axs[1].set_xticklabels(x_tick_labels)

# Set the tick positions and labels for the y-axis
y_ticks = np.arange(0, height, 100)
y_tick_labels = np.arange(0, height, 100)
axs[1].set_yticks(y_ticks)
axs[1].set_yticklabels(y_tick_labels)

# Set the axis labels for the clustered image
axs[1].set_xlabel('X-axis (in 100\'s)')
axs[1].set_ylabel('Y-axis (in 100\'s)')

# Remove ticks and labels from the original image plot
axs[0].axis('off')

plt.show()
```



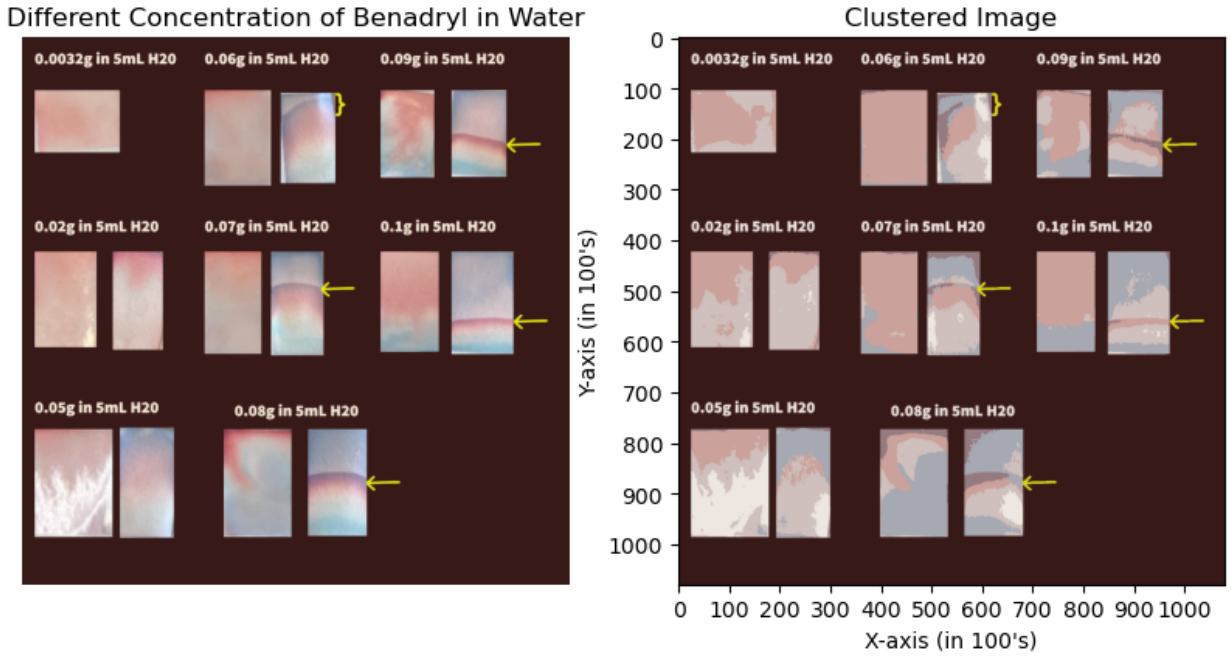
Team:16
Version:1
Date:08/06/2023

V. Result and Data Analysis:

K-means clustering gives a nice distinct visual contrast and differentiation by assigning specific cluster based on it RGB values.Th clustered image highlights distinct regions with different color. When comparing the clustered image with the original image it can be seen that the clustered image has certain level of detail preservation, color accuracy, and overall visual quality in the clustered image is enhanced.

Test Case 1:

Number of Cluster Applied =7

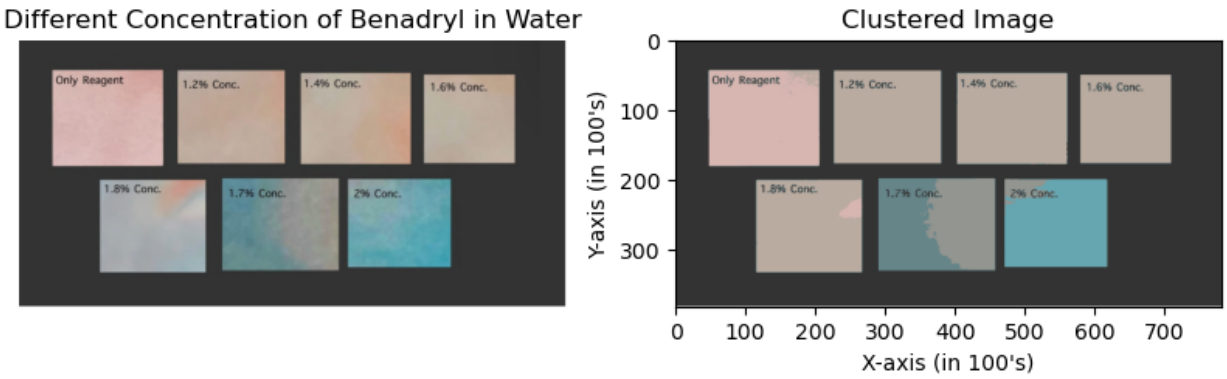


The picture on the right is the Original image and image on the left is the Clusted Image. This properly does image segmentation and grouping similar data points (colors). By dividing the image into 7 clusters the algorithm aims to identify and group pixels that share similar color characteristics. Dividing into 7 clusters was ideal for these images. When done more or less than 7 the clustering did not work effectively. Yellow arows are points towards the area of interest where the color has risen.

Team:16
Version:1
Date:08/06/2023

Test Case 2

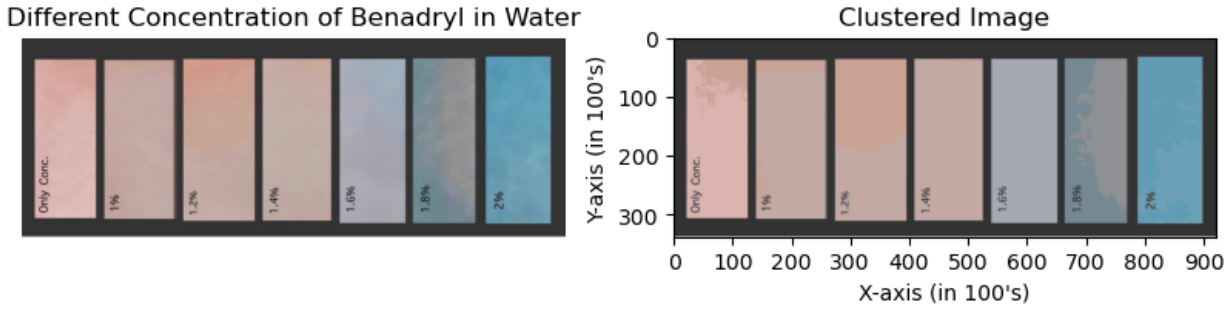
Number of Cluster Applied = 7



The picture on the right is the Original image and image on the left is the Clusted Image. Only a few selected test strips have been used to do a further analysis. Image segmentation and grouping of similar data points (colors). By dividing the image into 7 clusters the algorithm aims to identify and group pixels that share similar color characteristics. Dividing into 7 clusters was ideal for these images. When done more or less than 7 the clustering did not work effectively. We can see that at 1.7% concentration of Benedryl there is clear distinction of dark blue(towards the left) and light pink+blue(towards the right). And at 2% concentration The blue color is distinctly seen. Color changes as we increase the concentration of bendryl

Test Case 3:

Number of Clusters Applied = 11



The picture on the right is the Original image and image on the left is the Clusted Image. Only a few selected test strips have been used to do a further analysis. By dividing the image into 11 clusters the algorithm aims to identify and group pixels that share similar color characteristics. Dividing into 11 clusters was ideal for these images. When done less than 11 the clustering did not work effectively. Going more than 11 was causing the device (laptop) to slow down. We can see that at 1.8% concentration of Benedryl there is clear distinction of dark blue(towards the left) and light blue(towards the right). And at 2% concentration the blue color is distinctly seen. Color changes as we increase the concentration of bendryl.

Team:16
Version:1
Date:08/06/2023

VI. Future Work/Plan

1. More testing with more research leads to improvements that can address the above-mentioned limitation and enhance the robustness of the findings/results.
2. A robust Machine learning approach can be taken if more data is available. Such as Convolutional Neural Networks (CNN) if a larger dataset with diverse test strip images containing variations is available. For example, face recognition in smartphones. This occurs as the machine learning model has learned the facial stature of the user and can unlock the smartphone in any condition such as various light conditions (in the sun, clubs, bars, pubs, etc). Similarly if enough test strips are there indicating different shades of positive color(i.e. blue) then a Machine learning Model using Convolutional Neural Networks (CNN) can be developed so that users can take a picture of the test strip and immediately get the results on a mobile application.

VII. References

1. Prasannahariveeresh, “Implementing Lane Detection in OpenCV Python: A Step-by-Step Beginner’s Guide - Prasannahariveeresh.” *Prasannahariveeresh - Read My Tech Blogs Here*, 18 Jan. 2023, https://jrprasanna.com/2023/01/18/implementing-lane-detection-in-opencv-python-a-step-by-step-beginners-guide/
2. “Matplotlib Pie Charts.” *Vegibit*, vegibit.com/matplotlib-pie-charts/. Accessed 19 May 2023. https://vegibit.com/matplotlib-pie-charts/
3. (LEDU), E.E. (2018) *Understanding K-means clustering in machine learning*, *Medium*. Available at: https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1 (Accessed: 09 June 2023).
4. *Introduction to image segmentation with K-means clustering* (no date) *KDnuggets*. Available at: https://www.kdnuggets.com/2019/08/introduction-image-segmentation-k-means-clustering.html (Accessed: 09 June 2023).