Team:16
Version:1
Date:05/05/2023

Standard Operating Procedure

I. **Protocol Title:** Data Analysis Workflow- RGB Value Calculation and Graph Plotting

II. **Brief Description of Protocol:** This protocol outlines a step-by-step procedure for analyzing image data by calculating RGB values and plotting bar graphs. The protocol focuses on extracting RGB values from test strip images, visualizing the results through bar graphs, and interpreting the obtained data. Additionally, it provides guidelines for assessing pixel changes between grayscale images.

Team:16
Version:1
Date:05/05/2023

I. **Protocol Title:** Data Analysis Workflow- RGB Value Calculation and Graph Plotting

II. Materials:

- Test strip images
- Anaconda: Python programming environment
- Jupyter Notebook: Data Analysis and Visualization
- Required Python Libraries: os, cv2, numpy, matplotlib.pyplot

III. Procedure:

1. **RGB Calculation:**
   a. Star by setting the working directory to the folder that has all the reaction test strip images.
   b. Import the required/necessary libraries such as OpenCV, NumPy, and Matplotlib.
   c. Load the images using the cv2.imread() function and store them in separate variables.
   d. Now compute the average pixel values for Red, Green, and Blue color channels in each image using cv2.mean() and store them in separate variables.
   e. Convert the average pixel values to a list and Using the round() function round them to two decimal places.
   f. Print the average pixel values for each color channel for each image.
2. **Bar Graph Plotting:**
   a. The previously obtained RGB values will now be subtracted from 255 to enhance the visual representation for better interpretation. (Detailed explanation is given on pages 2 and 3)
   b. Set and define the labels, values, and colors for the bar graph plot.
   c. Use a for loop to loop through the clause list and plot a bar graph for each image using plt.bar() function.
   d. Add a legend to the bar graph chart using plt.legend.
   e. Set labels for the x-axis and y-axis using plt.xlabel and plt.ylabel.
   f. Set the chart title using plt.title.
   g. Show the plot using plt.show() function.
3. **Pixel Change:**
   a. Select two pictures for comparison and convert the two images to grayscale using cv2.cvtColor() and cv2.COLOR_BGR2GRAY
   b. As the image is edited to focus on the area of interest resize the grayscale images using cv2.resize() function.
   c. Compute and calculate the absolute difference between the two grayscale images using cv2.absdiff()
   d. To find the pixels that have changed threshold the difference image using cv2.threshold()
   e. Count the white pixels in the threshold image using cv2.countNonZera().
   f. Compute the total number of pixels in the grayscale image using the shape attribute (to get the dimensions) of the numpy array.

g. To determine the percentage of changed pixels divide the count of white pixels (which is the region that has changed) by the total number of pixels in the image. Multiply the obtained ratio by 100 to obtain the percentage value, which provides a quantitative measure of the pixel changes in the analyzed region.
h. Print the obtained percentage of pixels that have changed.

IV. Guidelines for Interpreting Results and Making the Data-Driven Decisions

1. **RGB Values Interpretation:**
   a. The calculated RGB values represent the average pixel intensity for each individual color channel (i.e. Red, Green, Blue) in the focused region of the test strip images.
   b. The higher the RGB value, the higher the intensity of that color channel in the image.
   c. Concept and Interpretation:
   d. Compare the RGB values between the non-tested ad tested regions of the test strip to identify any significant differences.
   e. A change in the RGB values suggests a color change and presence of the drug in the tested region compared to the non-tested region.
2. **Bar Graph Interpretation:**
   a. The bar graph visually represents the RGB values for each image.
   b. Compare the heights of the bars to identify any significant changes or differences between the images.
   c. The RGB graphs are also available in a normalized form for an overall representation.
   d. A higher bar indicates a higher normalized RGb value representing a higher intensity of the merged RGB channels. This helps in identifying patterns or trends in the normalized RGB values like the variation between different test strip images or between other regions of the same test strip.
3. **Pixel Change Interpretation:**
   a. The percentage of pixel change represents the amount of difference between two grayscale images
   b. A higher percentage indicates a larger difference or change in pixel intensity between the two images.
   c. Compare the percentage of pixel change between the different test strip images to identify variations or deviations
   d. A significant change in pixel percentage may indicate a color change or the presence of a substance(i.e. drug) in the tested region compared to the reference image.
4. **Explanation of RGB Values and Subtraction for Visual Representation**
   a. RGB values represent the intensity of red, green, and blue colors in an image. Each color channel has a value from 0 to 255, where 0 is the absence of that color and 255 represents the maximum intensity of that color.
   b. In relation to image analysis, darker color tends to have lower RGB values because they absorb more light and brighter colors have higher RGB values as

they reflect more light, while brighter colors have a much higher RGB value as they reflect more light. So as the color gets darker the RGB value decreases.

c. When plotting Bar graphs to represent the RGB values it is challenging to interpret results directly as color intensity is inversely related to RGB values. To resolve this issue and improve visual representation a simple transformation can be applied.

d. We subtracted the RGB values from 255 to invert the color intensity and now the darker colors get the higher values and the brighter colors get a lower value. This is better for a more intuitive interpretation of the bar graph, as higher values indicate a more intense color and lower values indicate a less intense color.

e. This makes it easier to compare and draw conclusions based on the relative intensities of different colors in the test strip images.

Note: This transformation does not alter the relative difference between the colors; it simply enhances the visual representation for better interpretation.

V. Data-Driven Decision Making

1. **Establish Baseline:**
   a. A reference test strip that is saturated with the reagent which is the baseline for comparison or a Non-tested region for the test strip
2. **Threshold Determination:**
   a. Define by testing on the test strips the set threshold for statistically significant change in RGB values or pixel percentage based on the acceptable level of variation or detection sensitivity.
3. **Comparison and Analysis:**
   a. Compare the calculated RGB values, bar graph patterns, and pixel change percentages with the earlier defined and established thresholds and also compare with other data to see if any deviations exist (with any specific drinks). Make note of color change situations (patterns) in the presence of drugs in alcohol.
4. **Statistical Analysis:**
   a. Apply statistical techniques such as hypothesis testing, ANOVA, regression analysis, or machine learning algorithms(With large datasets) to analyze the collected data. Observe the difference in RGB values or pixel percentage to validate and check if the difference is statistically significant.
5. **Decision Making:**
   a. Based on the statistical analysis, make the decision if the drug is present or absent in the tested region of the test strip. Consider the threshold and other related information.
6. **Valid and Iteration:**
   a. Do further analysis, comparison, and validation through experiments. Seek advice from researchers and experts from various fields to ensure the accuracy and reliability of the analysis. Based on the learning, additional data, or new insights obtained enhance the capabilities of the test strip.

VI. 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/