Protocols for Color Classification on iNaturalist or other photographic datasets:

1. Download the dataset from iNaturalist
   1. [Insert the necessary steps
2. Download the photos from iNaturalist
   1. Create a local folder called “Observations”. This will be the location where all the photographs will be downloaded.

A picture containing background pattern

Description automatically generated

* 1. Open the [Script name for the Downloader] Script. It is necessary to modify a couple of variables to fit your work.
  2. First, change the “***dest”*** variable to the file path of your Observations folder. (“***dest” =*** destination)
     1. Graphical user interface, text, application, Word, email

        Description automatically generatedYou can do this easily by opening the File Explorer and selecting the Observations folder. Depending on the OS and its version, there should be an option to copy the file path.
     2. Paste the path for the “Observations” into the script. In a Windows environment, the result should look something like this:

1. dest **=** r"C:\Users\username\Documents\Observations"
   1. Second, change the ***“df”*** variable to the path of the iNaturalist spreadsheet. (“***df”*** = dataset file)
      1. Refer to step 2.c.i for how to do this easily
      2. Paste the complete path for the spreadsheet into the script. In a Windows environment, it should look something like this:
2. df **=** pd.read\_csv(r"C:\Users\username\Documents\spreadsheet.csv")
   1. Graphical user interface, text, application, email

      Description automatically generatedSave the changes and run the script. This will start the process of downloading the photos in the dataset file. Depending on the size of the dataset and your internet speed, this may take several minutes. The script will notify you of progressing by giving the current picture out of the total pictures (indexed at 0) (ie, if you have 100 images, once the third image is downloaded, the script will output 2/100)
3. Obtain estimates of K and HSV. Determining a good estimate of K separates the flower(s) from the background. A good estimate of HSV captures the basic color profile of the flower and will be used in subsequent steps.
   1. Use a random sampling method to choose N images from the dataset.
   2. Open the [Script Name for color cluster visualizer] Script.
   3. Go to the last line in the script to the function called ***“get\_summary\_visual()”***
   4. Change the first argument to the path of the first image of a flower
      1. Similar to steps 2.c.i and 2.d.i, open the File Explorer and select the image of the flower and copy the path to that file.
   5. Change the second argument to the value “***k”***, the number of clusters used in K-Means Clustering. A good initial value for ***“k”*** is 5, which will give you a frame of reference on how K-Means Clustering splits an image into major color clusters.
   6. The resulting line should appear as below:
4. get\_summary\_visual(
5. r"C:\Users\username\Documents\Observations\Observations123",
6. 5
7. )
   1. Run the script. Once the script finished running, a new tab in browser should appear, giving the average HSV values for each cluster as well as an image with the highlighted cluster.

Graphical user interface

Description automatically generated

* 1. A group of white flowers

     Description automatically generated with medium confidenceRepeat steps a-g, changing the value of ***“k”*** until clusters with flowers are completely separated from the background. It is OK if flowers appear in multiple clusters.
     1. Note that as ***“k”*** increases, the flower will more completely separate from the background, but the computational time also increases, so high values (> 10) may take some time to complete.
     2. It’s recommended to increase the value of ***“k”*** in increments of 2 if significant parts of the flower are still mixed in the background. As you increase ***“k”*** and the flower becomes more separated, increase in increments of 1.
  2. Repeat steps a-h for each image in the randomly selected set, making any adjustments to ***“k”*** as needed. By the final image, you should have a ***“k”*** which reliably separates the flower from the background.
     1. Note that your value of ***“k”*** will probably never completely separate the flower from the background 100% of the time. Use your best judgement to determine the “k” value that best separates the background from most images. If there is background noise mixed with the flower, make sure the value of ***“k”*** mitigates the effect.
  3. Once a reasonable ***“k”*** value is selected, repeat steps a-h on each image in the randomly selected set, taking note of the HSV values of each cluster in which flowers appear until a general estimate of the upper and lower HSV bounds are derived

1. Obtain summary statistics from all downloaded photographs
   1. Open the [Script name for the Data Collector] Script.
   2. Insert the values for upper and lower HSV bounds determined in 3j to the ***“lower\_bound”*** and ***“upper\_bound”*** variables in this script. E.g.
2. lower\_bound **=** (112, 26, 0)
3. upper\_bound **=** (155, 165, 255)
   1. Change the ***“df”*** variable to the path of the dataset spreadsheet.
      1. To do this easily, open the File Explorer and select the dataset file. Depending on the OS and its version, there should be an option to copy the file path.
      2. In a Windows environment, it should look something like this:
4. df **=** pd.read\_csv(r"C:\Users\username\Documents\spreadsheet.csv")
   1. In the second to last line, change the second argument in the ***“image\_summary”*** function to the value of desired value of k (determined in Step 3.i.) for K-Means Clustering: (e.g. below k=15)
5. df["KMeansData"] **=** df["Path"].apply(
6. **lambda** x: image\_summary(
7. x,
8. 15,
9. lower\_bound,
10. upper\_bound
11. )
12. )
    1. In the last line, change the argument in **“*df.to\_csv()”*** to the path of the final spreadsheet to the location and name you want.
13. df.to\_csv(r"C:\Users\username\Documents\image\_data.csv")
    1. Run the script. Depending on the dataset size and the value of k, this may take a significant amount of time. It may need to run overnight.
14. Classification
    1. The classification steps needed would vary with the floral colors specificities for your species of interest, but you now have the raw data required to make those classifications.
       1. For *Geranium maculatum,* we use a percentile of saturation values. Based on greenhouse data, we classified anything above the 80th percentile as dark, below the 20th percentile as light, and anything in between as medium.