Working Prototype and Known Problems Report

Product name: Automatic Cell Counting
Team name: Cell Counting Project
Date: 06/01/2021

List of functions not working correctly:

- While the tool is counting on an image, no indication is given, so it looks like the tool is not working. The application may be marked as Not Responding while this occurs as well. The first prediction after the program starts takes especially long.
    - Steps to reproduce:
        - 1. Browse for any image
        - 2. Select an image
        - 3. Wait for prediction to appear (check the anaconda prompt regularly to make sure there are no crashes)
        - 4. Repeat steps 1 and 2
    - Location of fault:
        - Lack of code in counterUI.py
    - Possible actions for removal of fault:
        - Add a label that is shown before calling code from predictor.py
- If the user tries to follow the python and conda related installation instructions while a conda environment is already active, it is possible that trying to run the tool will crash immediately due to an import error. You must make a fresh python3.7 environment off of to run this application.
    - Steps to reproduce:
        - Install according to the instructions on the readme, but using miniconda3 instead of anaconda
        - Run the command “python runner.py”
    - Location of fault:
        - Unknown
    - Possible actions for removal of fault:
        - Unknown.
- The “Next Image” button when using the browse directory function does not go away if the user then tries to open a specific image for predicting.
    - Steps to reproduce:
        - 1. User must open a directory with at least 1 image in it
            - User should see the download buttons as well as the next image button
        - 2. User must then browse for a specific image
        - 3. The user will see the download buttons as well as the next image button
    - Location of fault:

        - counterUI.py function named browseDirectory
    - Possible action for removal of fault:
        - Change the code to use button classes which can remove themselves from the window and remove the button after the user browses for a specific image.
- If the user browses for a directory with at least 1 image in it and then browses for a different directory with 0 images in it, the “Next” button will cycle to the original directory’s next image.
    - Steps to reproduce:
        - 1. User must open a directory with at least 1 image in it
            - User should see the download buttons as well as the next image button. There will also be the predictions of the first image.
        - 2. User must then open a directory with 0 images in it
            - User should see the download buttons as well as the next image button. The predictions of the original image in the first directory opened will still be there since we did not want to clear output if there was no new image to predict on.
        - 3. The user must click the “Next Image” button
            - User should see the download buttons as well as the next image button. The predictions of the second image in the first directory opened will show up, unless the first directory had only 1 image in it, in which case it will repredict on the first image.
    - Location of fault:
        - counterUI.py function named browseDirectory
    - Possible action for removal of fault:
        - Change the code to use button classes which can remove themselves from the window and then remove the next button after opening an empty directory or a directory with 0 images in it.
- If the user browses for a directory with only one image in it, then the “next image” button still appears.
    - Steps to reproduce:
        - 1. Make a directory with only one image in it
        - 2. Open app
        - 3. Browse for and select that image
    - Location of fault:
        - counterUI.py function named browseDirectory
    - Possible action for removal of fault:
        - Add a check for number of images in the directory. If only 1, then only show the UI elements that normally show up when selecting a single image.
- Browsing directory causes the terminal to not exit the program even if the tkinter window is closed because of the while loop. This loop **cannot** be exited by ctrl+c. The user must exit out of the terminal to end it.
    - Steps to reproduce:

        - 1. User must open a directory with at least 1 image in it
        - 2. User must then close the tkinter window
        - 3. The user will see that the program has not terminated inside the command prompt
    - Location of fault:
        - counterUI.py function named indexLoop
    - Possible actions for removal of fault:
        - Change the loop to not go back to the beginning of the directory
        - Change the functionality to not use a while loop that waits- instead just iterate the next image in the list
- Browsing for a specific image causes the terminal to not exit the program even if the window is closed. This unintended functionality was found when testing in a fresh environment. It did not occur in the environment we had previously set up and had tested in. This loop is able to be broken by ctrl+c.
    - Steps to reproduce:
        - 1. User must browse for a specific image
        - 2. User must then close the tkinter window
        - 3. The user will see that the program has not terminated inside the command prompt
    - Location of fault:
        - Unknown
    - Possible actions for removal of fault:
        - Unknown
- Buttons have a white background on Mac, even though they should be yellow.
    - Steps to reproduce:
        - 1. Run the application on Mac
    - Location of fault:
        - Unknown
    - Possible actions for removal of fault
        - Look into ttk Buttons instead of Tkinter buttons
- Images that do not have cells on them at all on rare occasions do get marked as having one or two cells in them.
    - Steps to reproduce:
        - 1. Run application
        - 2. Upload image that has small, round-ish objects like earbuds
    - Location of fault:
        - In the model
    - Possible actions for removal of fault
        - Train the model better