User Manual

12/12/2024

CRAFT (Ceramic Recording Automation and classiFication Team)

Project Sponsor:

Dr. Leszek Pawlowicz

Faculty Mentor:

Vahid Nikoonejad Fard

Course Organizer:

Isaac Shaffer

Team Members:

Kimberly Allison, Aadarsha Bastola, Alan Hakala, Nicholas Wiley

Table of Contents

1. Introduction

Team CRAFT is pleased that you have chosen our sherd classification systems for all of your archeological needs. Our sherd classification system provides an accurate, consistent, deep learning model, while also accounting for differences in environment, equipment availability, and archeological expertise. The purpose of this manual is to help you, the client, install, configure, and maintain the sherd classification system going forward. Our aim is to provide you with a system that can be utilized for a long time.

2. Installation

The following subsections detail initial installation steps for the mobile application, conveyor belt system, and deep learning model.

2.1 Mobile Application

The mobile application will be provided in an apk format which a user can copy into their Android device, then the user can use file explorer of their choice to open the apk file which will prompt the user to install the application. The application then is installed in the user's smartphone.

For IOS currently distribution is unavailable as in IOS devices it can be only distributed through App Store. If a user wishes to run in IOS they must obtain the source code for the application and launch it in release mode. Even doing this the app will run for one week only after which the Developer Certificate expires, and the same process needs to be repeated. Following code can be used to run the app in IOS in release mode:

```
cd mobile
flutter run --release
```

2.2 Conveyor Belt

The conveyor belt application requires that the user have a version of Python installed. Specifically, the program was developed in Python 3.11, however any version of Python will suffice so long as it is new enough to function with the program's required libraries, version specific compatibility per library can be found in the respective documentation online. To install the required libraries a text document can be located on the code's repository detailing all command line commands to be inputted. After these steps have been completed the user is ready to use the conveyor belt program.

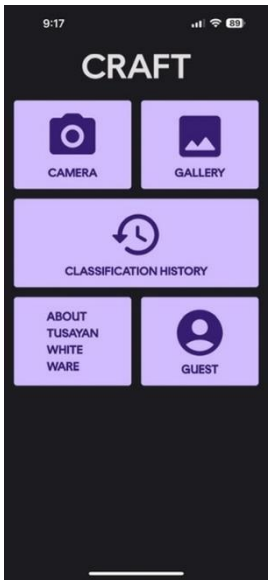
2.3 Deep Learning Model

For training the deep learning model two Anaconda environments were created. The first conda environment can be created using the requirements.txt file included in the source code found at CRAFT/ai. The channel “conda-forge” will need to be added for the packages to be found but once this is done conda will handle all the heavy lifting leaving you with a set up environment. The second conda environment was only used to train the most recent ConvNeXt model from the code you provided us in a downgraded TensorFlow environment for integration into the mobile app. There is no requirements.txt file for this as the environment should be the same as yours.

3. Configuration and Daily Operation

Routine maintenance tasks will ensure that the software functions at an optimal level. The following subsections detail the steps necessary to keep the software operating smoothly.

3.1 Mobile Application



For the mobile application there will be not much to configure, since the installer package is provided, just running the installer should install the complete application.

When the user opens the app, in the homepage they see a few options:

1. Camera
2. Gallery
3. Classification History
4. About Tusayan White Ware
5. User Account

Selecting Camera will allow the user to take an image of a sherd, which then will be classified using our image classification model.

On selecting Gallery, users can select an image which will then be classified using our image classification model.

After users have classified the image, they are able to save the classified image and corresponding metadata to the device's local storage.

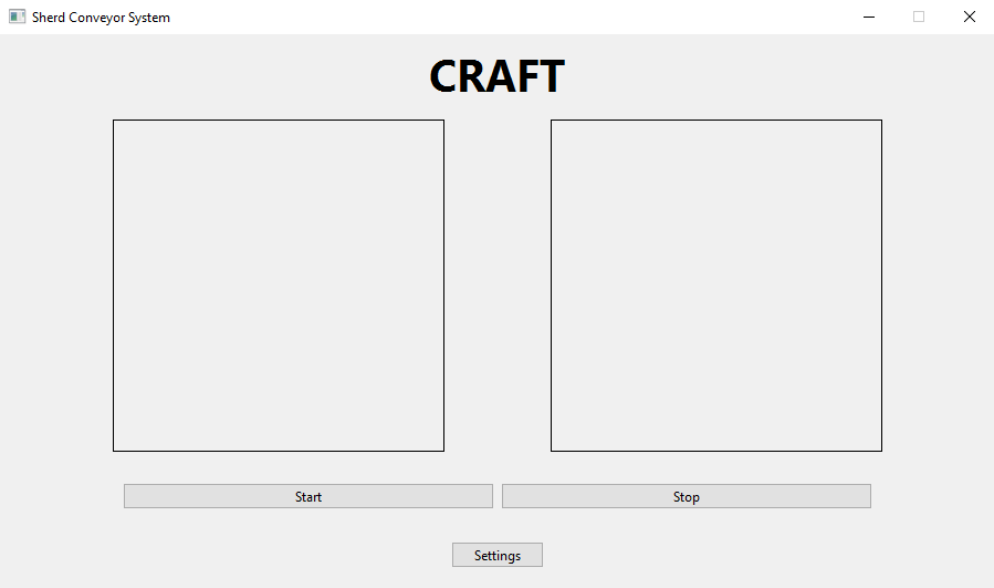
Users then can go to homepage and go to Classification History, where they can see all classifications they saved in their local device storage or cloud storage.

Users then can sync the local storage classifications into cold storage if they are logged in. If they are not logged in, they can go back to the homepage and then select User icon and log in (or register, if required).

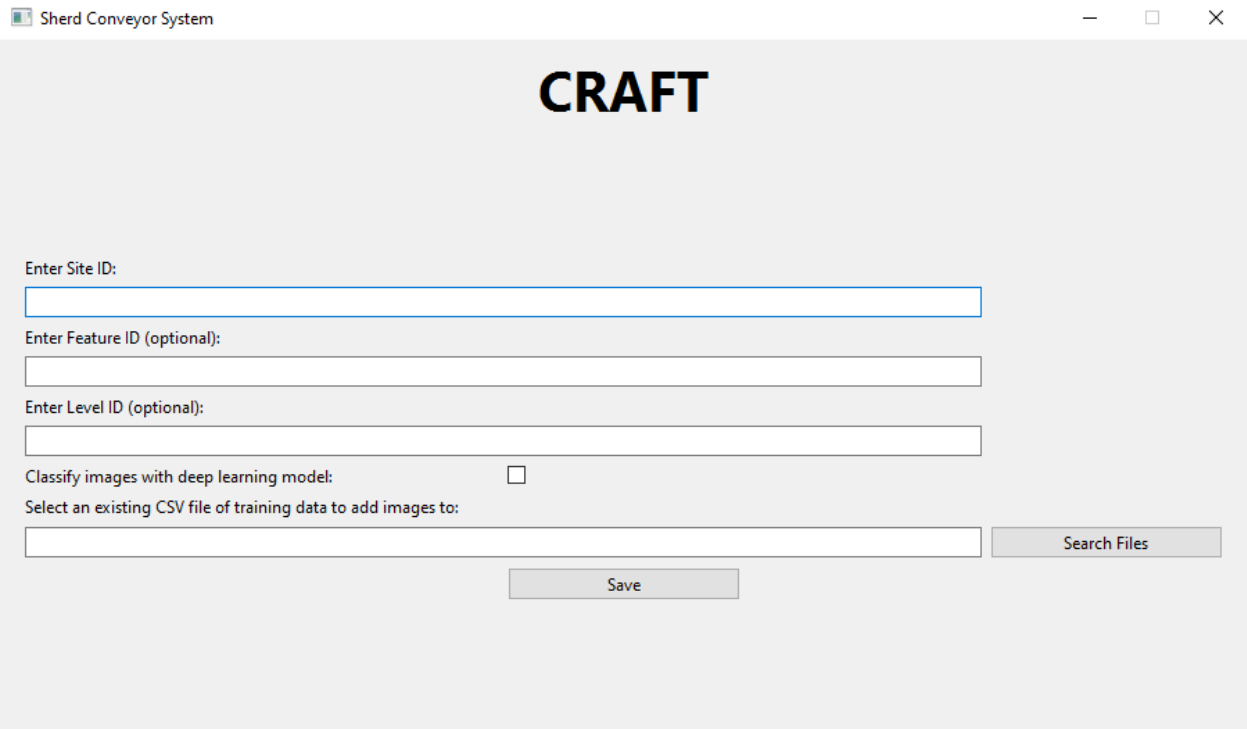
Users can also change the app theme to Dark/Light/System from the Account Page.

3.2 Conveyor Belt

The purpose of the conveyor belt program is to speed up data upload. Depending on the user's wishes the type of data upload and desired results may change, we will see which settings to alter to achieve your desired outcome. Upon starting the program, the user will see the screen below:



When started the left window will display our camera's view and our right window will display the images taken and prepared for the image model. Before we start the program, we should provide the program with some important information on the settings page. Upon clicking the settings button, the user will find themselves at the screen below:



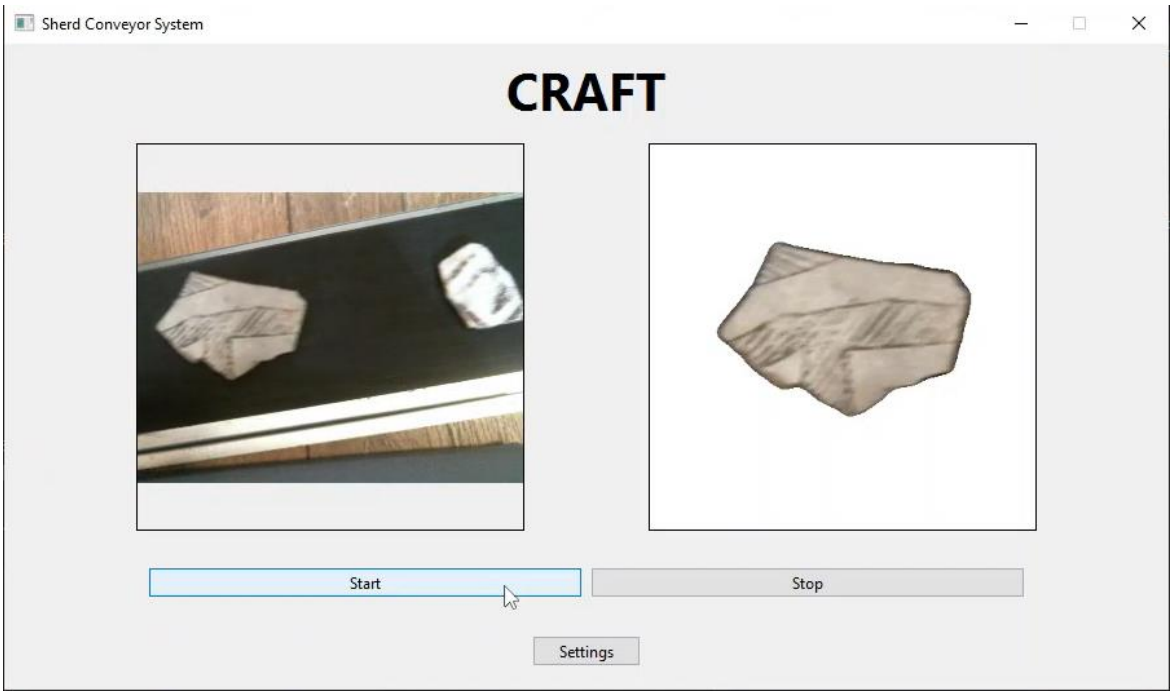
The first three text fields

- SiteID
  - denotes the site the sherds captured in this upload process were found at
- FeatureID
  - denotes the sherd's classification out of the 7 categories of Tusayan white ware
- LevelID
  - denotes the depth they were found at

are used to determine the naming system following the format

- SiteID_FeatureID_LevelID

The directory will be created if none exists or will be used as is if it does exist.
Only the SiteID is required for the program to operate. If we would like to classify images we are going to capture then the box for it should be checked, and if we would like our images and classifications to be saved to an additional csv file, we should click the "Search Files" button, which will open a window of your device's file browser to use and find our desired .csv file. After we have finished tuning the settings to what we want to use the program we are ready to save them, which will return us to the main screen.



Upon starting our program and placing sherds onto the conveyor belt the camera display screen will start up. When sherds move through the center they will be photographed, an unedited original capture will be saved along side

When you have finished uploading this batch of sherds, press the stop button. This will end the data capture process, if classification was enabled after pressing stop the user will be able to see the classifications in the csv file(s). Pressing stop ends the collection process, not the program. If additional batches with different desired settings need to be uploaded, they can be. Once stopped the user has the ability to reconfigure the settings for a new batch (or keep the same settings) and start the program up again to continue use.

If the user wishes to instead stop the program rather than continue data upload pressing the “x” (close) button in the top tile bar. All edited sherd images can be found in the user’s specified directory, along with the unedited original images found in an archive directory. The directory created for edited sherd images will contain the csv where files have been archived to.

3.3 Deep Learning Model

If the installation process listed previously was followed, there will be little work left to start training. The first step necessary will be moving a copy of the images to use as training data into the project. Our code base maintains roughly the same directory structure for images as when the project was first given to us so follow the original setup. Next, you need to activate the

conda environment respective to the python script you want to run. After this you should be all set up and ready to train.

4. Maintenance

Due to the nature of the application, little routine maintenance is required to maintain the health of the software. In the subsections below, potential concerns for all aspects of our project are listed.

4.1 Mobile Application

Since the mobile application is already packaged into an apk file for Android, maintaining the codebase is not required, however the distribution in IOS is not possible without releasing it into the Apple App Store, so there might be a chance whenever the app is ready to be released to the App Store, the pods might need to be updated, which should be handled automatically by Flutter. In case of Android platform, re-packaging the app in future might require a Gradle upgrade, which can be updated through the AGP Upgrade Assistant in Android Studio.

If an update/change is to be made then the system needs to have Flutter, Android Studio (for Android SDK), Java Development Kit, Xcode (if on Mac and Running on Apple Device) will be required to be installed.

4.2 Conveyor Belt

The conveyor belt application will likely not need extensive maintenance. Package or program updates should not majorly affect the functionality of the system. Since storage is local, there also should not be issues with handling a large number of sherd photos.

4.3 Deep Learning Model

No maintenance is required for the deep learning model. Updating any of the packages could and most likely will result in the environment being unable to run the code due to how often TensorFlow and Keras completely change their library structure.

5. Troubleshooting

Challenges may arise during the installation process, configuration, or common operation. The following troubleshooting guide identifies and addresses some common issues that may prevent use of the software.

5.1 Mobile Application

Our mobile app is made using Flutter framework, which utilizes wide array of open-source packages from pub.dev, this sometimes might be tricky as constantly newer versions of packages are being released which can cause breaking issues if there is a version mismatch with Flutter SDK, so if there is any error caused by packages, updating all of them is strongly recommended. Any issues related to a runtime error can be caught if the source code is debugged, the issues should be listed in detail in the Debug Console of the IDE. Android pushed Gradle update quiet frequently, which sometimes having a lower version of Gradle won't even compile the application, so if there is such issue using the AGP Upgrade Assistant in Android Studio is recommended to upgrade the Gradle version.

5.2 Conveyor Belt

Before using the conveyor belt, ensure that all packages mentioned in the installation section have been properly and completely installed. Before running the program, check that all necessary information has been entered into the settings page,

Depending on your machine, camera, and system information a common error can occur that occurs *only* upon the first boot of the program on your machine after a restart. This error will cause the program to detect the whole camera view as an object, preventing images from being taken. To resolve this issue simply close the program and boot it up again, it will work from here on out until the device is restarted.

To ensure the best results the conveyor program should be run in a controlled environment with stable lighting. If the conveyor is unstable and shakes results will be subpar, and if the lighting around the conveyor is inconsistent similar poor results will arise. For optimal results ensure the conveyor belt is lit completely with natural lighting that is neither too harsh nor too faint. Environments with well-lit soft defused lighting produce optimal conditions. Camera placement directly overhead the conveyor belt is essential as well.

5.3 Deep Learning Model

The main issue you could run with the deep learning model portion of our project will be issues with packages. Tweaking the environment in any way or setting up a new one can cause far-reaching problems. While using conda environments has fixed some of the headaches we ran into when first working on this project, package issues can still arise from something as little as updating. The best way to fix these issues is to adhere to the requirements.txt file given and refrain from updating packages within an environment. If updating is necessary, we recommend doing so in a newly conda environment so if problems occur you still have a working environment.

6. Conclusion

With best wishes from Team CRAFT: Kimberly Allison, Aadarsha Bastola, Alan Hakala, and Nick Wiley. Team CRAFT has enjoyed the opportunity to work on such a challenging project alongside our client. Although our team is moving on to professional careers, we are happy to answer short emails in the coming months to ensure that the project is operating at the optimal level. Our emails: Kimberly Allison (kimall1998@gmail.com), Aadarsha Bastola, Alan Hakala (alanghakala@gmail.com), Nick Wiley (nwiley333@gmail.com). Team CRAFT is looking forward to seeing the impact the project will make for both hobbyists and professional archeologists.