Nelson Foster

DATS 6203 - 11

Machine Learning II

*Individual Final Project Report - Group 3*

***Introduction***

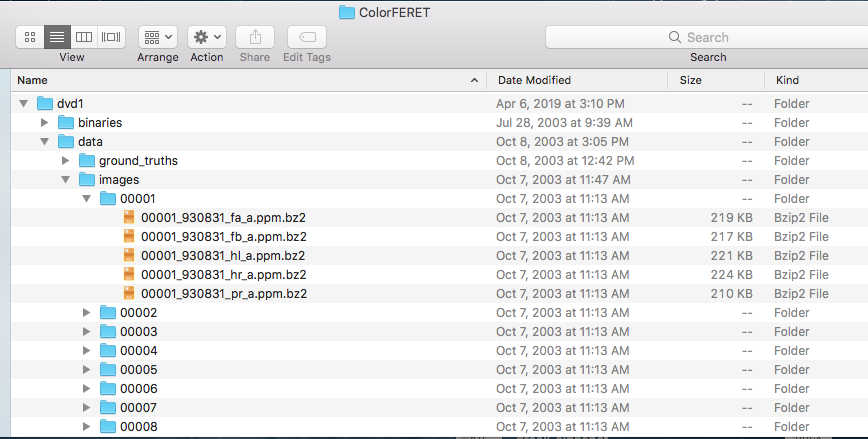
Facial recognition has been at the forefront of deep learning applications. Our project sought to leverage publicly available facial image datasets and frameworks to train a deep network to correctly classify facial expressions.

*Description of Individual Work*

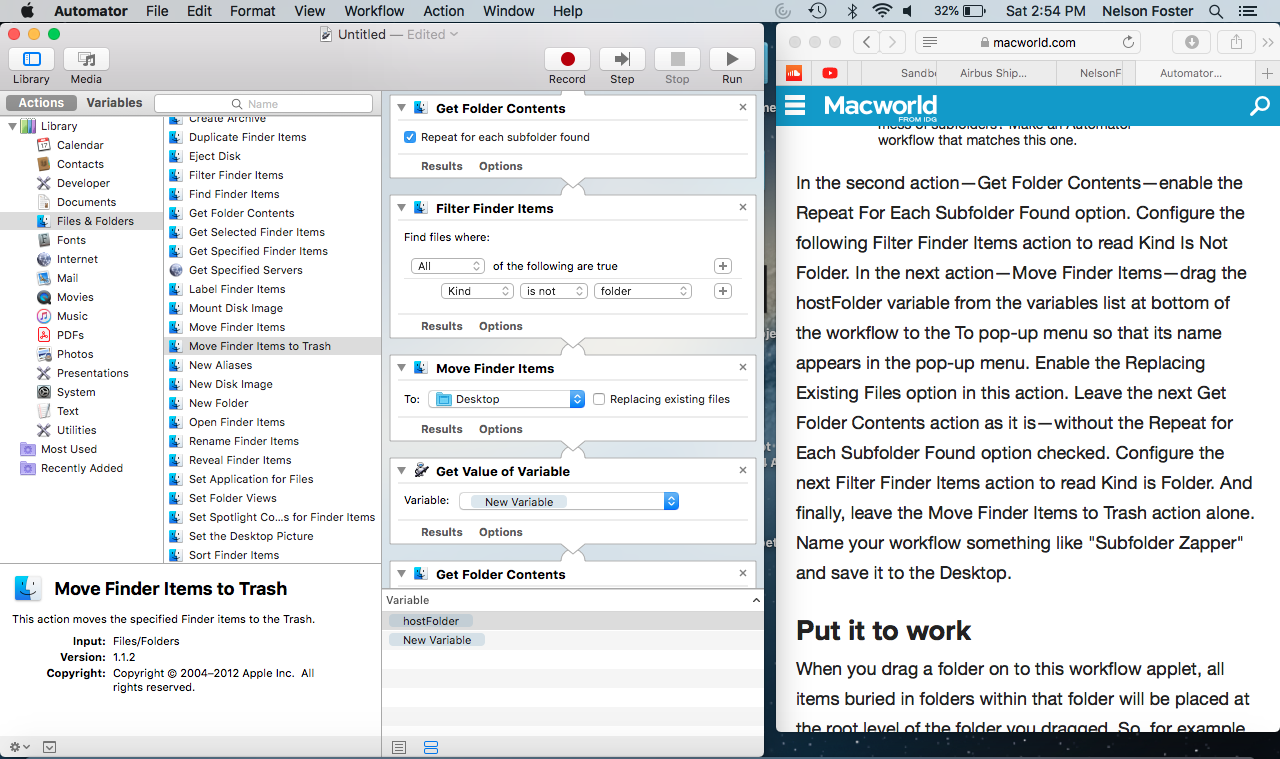
After we decided to work with facial recognition, I researched the various repositories, blogs, research, and other data resources provided by the instructor. Robust databases of facial images are not as readily available as others, so the first one that I was able to find that was accessible was the YaleFaces dataset, which only includes 165 images of facial expressions. After consulting with the professor, we decided to move forward with using YaleFaces as an initial point of departure for designing and training a deep network, in the hopes of adapting it to a larger, more complex dataset.

I requested and was granted access to the National Institute of Standards & Technology’s (NIST) Color Face Recognition Technology[Color FERET](https://www.nist.gov/itl/iad/image-group/color-feret-database) database, a color image facial recognition database with over 11,000 images used to evaluate and research biometric systems.

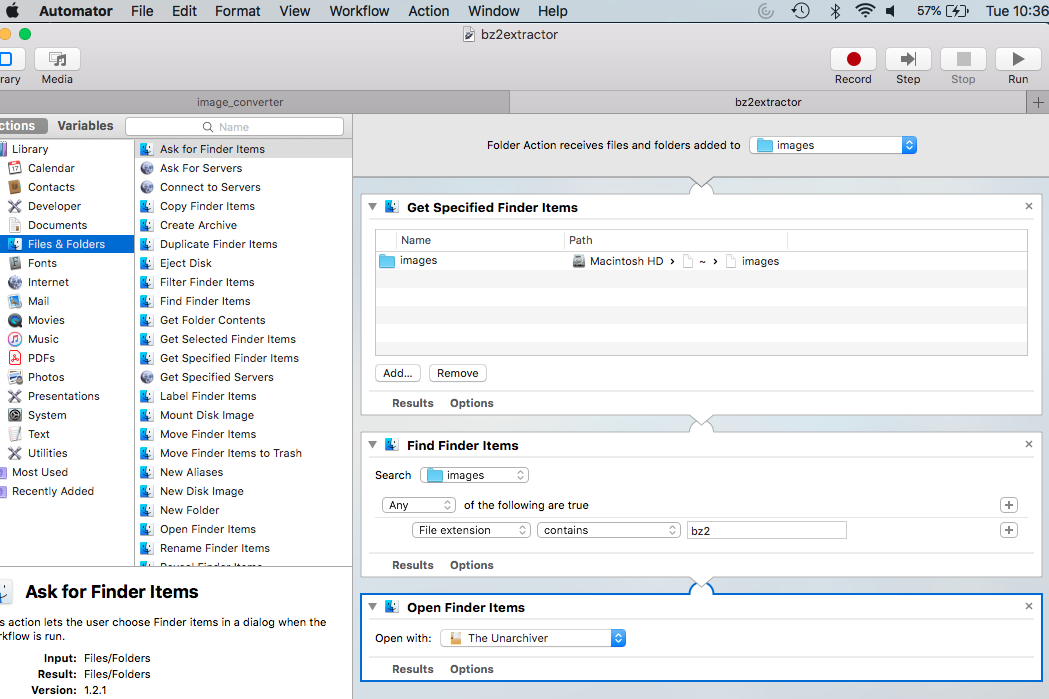
For most of the frameworks to function properly, images have to be in certain formats (JPEG, GIF) and be in a common directory. Unfortunately, ColorFERET’s images were collected between 1993 and 1996, so they were in a .PPM format. Further, because the initial dissemination of this data was in DVD format, the data, metadata XML files, ground truth data, documentation, and images, were split into various nested folders and subfolders. Further, all the image data was in a dated archive format (BZ2). Though the dataset came with software to extract the files, the software is obsolete and incompatible with OSX.



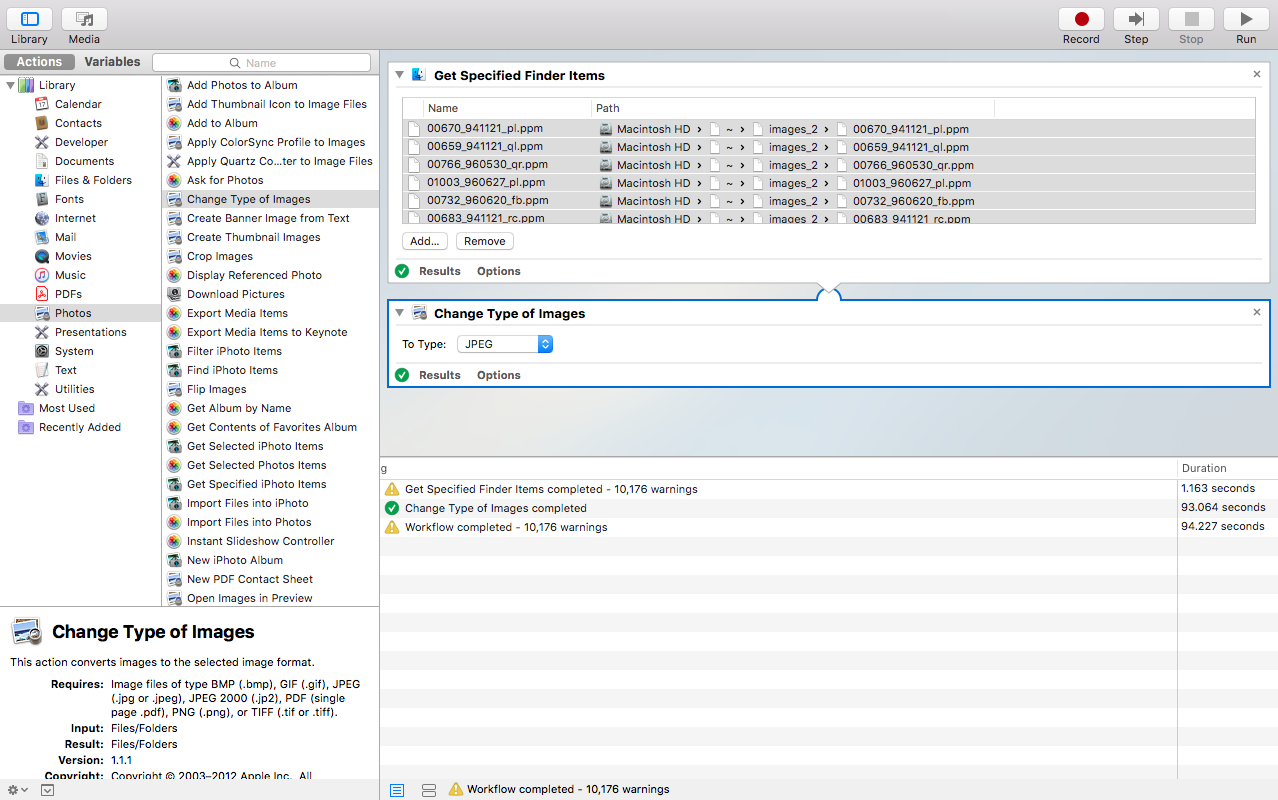
After searching unsuccessfully for packages that would resolve the issue, I found instructions on how to use [Automator](https://support.apple.com/guide/automator/welcome/mac) to develop workflows/mini applications to help with this issue. I first developed a workflow to extract all non-folder elements (i.e. image files) into a single folder:



Then, I developed a workflow that automated use of [The Unarchiver](https://theunarchiver.com/) application, specifically it unarchived the images, saved them in a designated directory, and moved the archive files to trash. This process took approximately six hours to complete due to the size if the file.



Finally, I created a workflow to convert all files to JPEG, to be compatible with PyTorch and Keras.



After these efforts, I directed my attention to developing code in Keras based on Francois Chollet’s *Deep Learning with Python* book. Specifically, I gleaned code from Chapter 5, Deep learning for computer vision. Though I completed the code, we ultimately came to the realization that classes ColorFeret and YaleFaces were incompatible and couldn’t be adapted to the same network without considerable code development. The concept of “expression” meant two different things. For YaleFaces, these dealt with lighting and emotional expressions; but for ColorFERET, expressions meant orientation of the face (frontal, left, right, slightly left or right, etc.):

|  |  |
| --- | --- |
| **YaleFaces’ Classes** | **ColoFERET’s Classes** |
| Class 0: ‘glasses',  Class 1: ‘happy  Class 2: 'leftlight',  Class 3: 'normal',  Class 4: 'rightlight',  Class 5: 'sad',  Class 6: 'sleepy',  Class 7: 'surprised',  Class 8: 'wink',  Class 9: 'centerlight',  Class 10: ‘noglasses'. | regular frontal image  profile left  half left - head turned about 67.5 degrees left  quarter left - head turned about 22.5 degrees left  profile right  half right - head turned about 67.5 degrees right  quarter right - head turned about 22.5 degrees right |

Due to these issues, we ultimately postponed our work on the ColorFERET data and code, to focus our efforts on refining the network for the YaleFaces dataset, where I gleaned some code from various internet resources to attempt visualization of the activation layers, filters, and kernels of the network.

Approximately 85% of the Keras and PyTorch code was borrowed and/or adapted from online resources and GitHub contributors, with various customizations to adapt it to our use case and data. Also, as mentioned earlier in this report, I had to research methods to implement Automator to preprocess the ColorFERET data. This more of a configuration-as-code approach using the application .