Bizzy Moore
Professor Bridgeman
Professor Dumitriu
CS Independent Study
Spring 2020

Annotated Bibliography

Language Recommendations for Machine Learning
https://www.c-sharpcorner.com/article/best-programming-language-for-machine-learning/

I visited this website when I was researching which languages were the best to use for machine learning. It lists the most popular languages that are used for machine learning and briefly explains the advantages of using those languages.

Python Machine Learning Tutorial
https://www.pyimagesearch.com/2019/01/14/machine-learning-in-python/
https://www.pyimagesearch.com/2018/03/12/python-argparse-command-line-arguments/
https://www.scikit-yb.org/en/latest/api/classifier/classification_report.html
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html

All of the websites above are websites that either contain tutorials about doing machine learning using Python, or further explain a method or concept that is mentioned in the tutorial. The first link is the tutorial that I used as a starting point for my code.

Extracting frames from AVI using VLC Media Player
https://www.raymond.cc/blog/extract-video-frames-to-images-using-vlc-media-player/
https://www.isimonbrown.co.uk/vlc-export-frames/
https://www.labnol.org/software/extract-image-frames-from-video/18137/

The above links were visited when I was searching for a program that would easily extract frames from a video and save them as images. I ended up using the media player, VLC, which was introduced to me in the first link. This website included number instructions on how to use VLC, which was very helpful. They were in windows, so I had to interpret the necessary changes in order to get it to work with a Mac.

Keras
https://keras.io/models/sequential/
https://www.pyimagesearch.com/2018/09/10/keras-tutorial-how-to-get-started-with-keras-deep-learning-and-python/
https://www.pyimagesearch.com/2018/12/31/keras-conv2d-and-convolutional-layers/

The links above helped me become more familiar with the Keras machine learning library. The first link is from Keras' website, and explains its Sequential model. The middle link is a tutorial that I found that helps you get started with Keras. The last link further explains how to use a convolutional network in Keras. These were helpful because I used a Sequential Keras model and a convolutional neural network for my project.

Scikit-learn
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classification_report.html
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html#sklearn.metrics.precision_recall_fscore_support
https://developers.google.com/machine-learning/crash-course/classification/true-false-positive-negative
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html#sklearn.metrics.confusion_matrix

Scikit-learn is a Python library. The tutorial that I used included a couple of methods from this library: *classification_report and confusion_matrix*. The links above include the Scikit-learn API for these methods, as long as a website that explains the concepts behind how to interpret the results of these methods.

Saving and Loading Trained ML Models
https://machinelearningmastery.com/save-load-machine-learning-models-python-scikit-learn/
https://machinelearningmastery.com/save-load-keras-deep-learning-models/
https://keras.io/getting-started/faq/#how-can-i-save-a-keras-model

Once I figured out how to successfully train a machine learning model, I had to determine how to save that model, so that I could load it into a different program. The links above gave me knowledge on how to do this. The first two links are from a blog that gave me good information. The last link is from the Keras API and gave me details on what specific Keras methods to call to complete this task.

Input and Outputs
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
https://www.quora.com/What-are-the-definitions-of-input-output-and-target-in-machine-learning
https://towardsdatascience.com/the-abc-of-machine-learning-ea85685489ef

In the beginning, I was confused by the relationship between inputs and outputs, and X and Y in the tutorials and sample code that I was looking at. The three links above provided me with more information on this topic and cleared up any confusion that I had.

Loss and Optimizers
https://algorithmia.com/blog/introduction-to-optimizers

The parameters for Keras' *compile* method include values for an optimizer and a loss function. I was not familiar with these concepts. The link above provided me with sufficient enough information that I was able to use the method with an understanding of what it was asking for its parameters.

Dealing with Paths in Python
https://stackoverflow.com/questions/18805348/how-rename-the-images-in-folder
https://docs.python.org/3/library/os.html
http://tutorial.simplecv.org/en/latest/examples/basics.html
https://pillow.readthedocs.io/en/stable/reference/Image.html

Once the images were provided with their particular classification from the model, the program was supposed to sort the images into folders based on their categorization. The simplest way to do this was to copy the images into a new folder. In order to do this, I needed to deal with the paths of the images. The links above provided me with information about this concept.

Accessing an Image Object from a Python list
https://realpython.com/storing-images-in-python/

One option I thought of for moving the images was moving the actual PIL Image object. The link above discusses the concept of doing this. However, I ended up not doing this. Changing the directory paths of the images was much simpler.

Copying/Moving Files in Python
https://docs.python.org/3/library/shutil.html
https://www.geeksforgeeks.org/python-shutil-copy-method/

I needed to determine how to actually copy the images from one folder to another. The two links above introduce a library called *shutil* which allows you to do this at a high-level.

'for' loops in Python
https://treyhunner.com/2016/04/how-to-loop-with-indexes-in-python/

One challenge that I faced with this project was that I had to learn and become familiar with Python's syntax as I went along. The "for" loops in Python are different from how they are in

Java. They do not iterate with a variable such as *i* and increase its value with every cycle. The link above familiarized me with how “for” loops work in Python.

Determining Current Operating System Python is Working With

https://stackoverflow.com/questions/1854/what-os-am-i-running-on
https://docs.microsoft.com/en-us/dotnet/standard/io/file-path-formats

When I was dealing with the path names of the images, I had to make sure that I made my program work with both Windows and Mac systems. This is because these two operating systems have some differences. One example of this is that one uses a forward slash in its path names, while the other uses a back slash in its path names. This meant that I had to determine which operating system the program was running on, so it would know which way the slashes should go for the file names. The two links above gave me information on this.

Backslashes as Strings in Python

https://www.pitt.edu/~naraehan/python2/tutorial7.html
https://docs.python.org/2.0/ref/strings.html

In Python, the backslash (\) is used with escape characters. However, the file path for Windows uses backslashes. Therefore, I had to look up how to write a backslash in Python if it was used for escape characters. The two links above helped me with this.

PurePath in Python

https://docs.python.org/3/library/pathlib.html

In order to get the filename of the image, I had to get the last element from the image’s path. The pathlib library provided methods to easily do this. The link above discusses how to do this.

Sorting lists in a list in ascending order in Python

https://stackoverflow.com/questions/4174941/how-to-sort-a-list-of-lists-by-a-specific-index-of-the-inner-list

In the *HABs_Classify.py* program, there contains a main list that contains other lists that contain information about each image’s classification. We wanted to sort these lists by a particular numerical value that they contained. The link above describes how to sort a lists of lists based on an element that occurs in the embedded lists.

Cross-validation
https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation

When we tested our CNN model, we wanted to make sure that we were doing it in an accurate way. A popular technique to test machine learning models is cross-validation. The link above describes this technique and how to implement it.

Results to feed to predict
https://machinelearningmastery.com/generate-test-datasets-python-scikit-learn/

When we wanted to import our own images into our model, we had trouble getting them into the correct format. This is because before that we were using a method to transform our results into the right format. However, this method did not do everything we wanted it to, so we needed a way to format the data from scratch. In order to learn how to do this I looked up how to generate test datasets for Python machine learning models. The link above popped up and was helpful in achieving this.

Keras Image Preprocessing
https://stackoverflow.com/questions/43469281/how-to-predict-input-image-using-trained-model-in-keras
https://machinelearningmastery.com/how-to-load-convert-and-save-images-with-the-keras-api/
https://stackoverflow.com/questions/32994196/how-to-use-vstack-in-a-for-loop-in-order-to-append-csr-matrix-matrices-together/32997278

The three links above were crucial in me determining how to transform the images into the correct format to feed into Keras' *predict* method. This is because these links each provided sample code that showed me an example of what I was looking for. I took small pieces from these three sample codes, and when I put them all together, I was able to transform the images into the correct format.

How to Train a Final Machine Learning Model
https://machinelearningmastery.com/train-final-machine-learning-model/

The link above provided me with information on how to train a final machine learning model. I wanted to make sure I was saving my final model correctly.

Argmax function
https://docs.scipy.org/doc/numpy/reference/generated/numpy.argmax.html

The “Argmax” function was used a few times in my code for this project. It first came up in the tutorials that I was using. The link above explained to me what it does and helped me understand its function and purpose.

Keras Sequential Model vs. Keras Functional Model
https://medium.com/@hanify/sequential-api-vs-functional-api-model-in-keras-266823d7cd5e

The Keras library provides two different machine learning models that you can use: the Sequential Model and the Functional Model. The link above described to me the difference between these two models. We used the Sequential model for our project because it is the easiest to use and understand.

Command prompts for Windows
https://www.onmsft.com/feature/command-prompt-basics-working-with-files-and-folders

The command line prompts for Windows and Mac differ from each other. I needed to be sure to address both commands in my user manuals in order to make sure that the user knew how to run the program on either system. I am not familiar with the Windows command line because I work mainly on a Mac. The link above familiarized me with the Windows prompts.