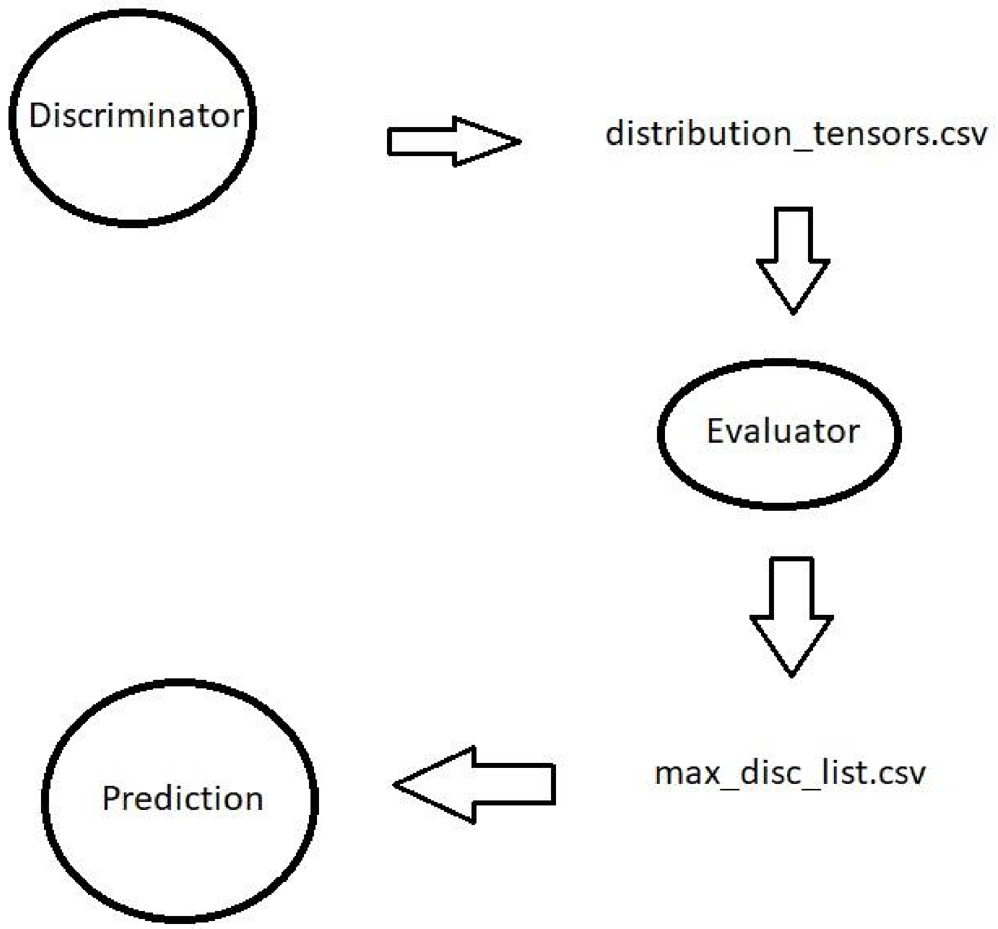
To containerize the MetaCog Algorithm using Docker, a few changes to the code have been made to segment the code into a modular system.

- The argument parser has been changed into a separate class.
    - This is located in 'Args_Class_Module.py'
    - This allows the code to function independently within a container without the need of command line input, while retaining customization from the configuration of default values within the class file.
    - This is imported in the other code segments for using default values.
- The first main segment of code is the Discriminator.
    - This is located in 'Docker_Discriminator.py'
    - First, it reads the static data file from the default value in the args class.
    - Then, a modified version of the 'runDet' function is called.
        - This version simply calls the discriminator and returns the output of the softmax classification, before it is evaluated.
        - This separates the discriminator from evaluation, so that a different evaluation process can be implemented instead of the current, which simply takes the element with the highest value, if needed.
    - After, the output is converted from a Tensor into a numpy array.
        - This is done so that the data can be saved to a file in a way that preserves information. To my knowledge, only a Tensor of strings can be saved to a file in TensorFlow. So, Numpy was used for ease of use.
    - The resulting probability distributions are saved to the file: 'distribution_tensors.csv'
- The second main segment of code is the Evaluator.
    - This is located in 'Docker_Evaluator.py'
    - First, it reads the CSV file from the discriminator segment and loads the data.
    - Then for each array in the file, the element of the maximum value is stored in a new array. This corresponds to the number of the best model for the distribution.
    - The resulting array stores all the selected models and saves to the file: 'max_disc_list.csv'
- The third main segment of code is the Detection Prediction.
    - This is located in 'Docker_fa_cd.py / Docker_fa_glrt.py / Docker_fa_ideal.py'
    - The results of the evaluator are first loaded into an array.
    - Then a second modified version of 'runDet' is called for each value in the array.
        - Previously the evaluator was run here; however, this was removed to make the code more modular. So, instead of this, the value loaded from the file is simply placed here instead. The rest of the function is the same.
    - The code functions normally from here, and the results are saved to corresponding files: FA_cd.csv / FA_glrt.csv / FA_ideal.csv
- A final container is made to display the results to the command line with labels.
    - The ideal container is toggled on/off in the Args Class Module
    - Docker_results.py


Discriminator
distribution_tensors.csv
Evaluator
max_disc_list.csv
Prediction