**Artifact 2 – Data Structures and Algorithms**

**Artifact Description**

This artifact is a Jupyter notebook which I used to build a neural network that can identify handwritten digits just by seeing a picture of them. I created it last year to practice building convolutional neural networks (CNNs).

**Artifact Justification**

CNNs and building AIs which allow a computer to “see” are a very in demand area of machine learning and having the skill to build and modify one to fit a problem is a very valuable skill. The building of various neural networks, as well as functions to convert the training data from grayscale to RGB showcase skill in training these models. These are complex operations and need to be coded correctly to function properly and scale up to training a model on thousands of rows of data. Regarding data structures, a linked list was used to house the accuracy metrics of the various models used and was then used to build informative graphs with the matplotlib library.

A custom sorting algorithm based on bubble sort was engineered to sort this linked list. The function which accomplishes this takes the linked list to be sorted as one argument, and another argument which is a string corresponding to the desired classification metric to be organized by (accuracy, recall, f1 score, etc.). The best performing model is sorted to be the head node, with each successive node being the next best performing model. This sorting function is called extensively during the visualization portion of the notebook. The creation of a linked list and sorting function demonstrate capability in the fields of data structures and algorithms.

The artifact was also improved by including many more neural networks. Originally only one model was trained, but with my additions three more neural network architectures were built from scratch and a model was built using the popular mobilenet\_v2 architecture. In addition, verification was added when reading the CSV files housing the data, many more informative graphs were added to show model performance, and a summary of the findings was added at the end. Hypertuning of the best model was also added in the improvements.

**List of Improvements by Section**

1. Data Exploration

* Implemented read\_csv\_with\_check function to verify existence of CSV files before transforming them into data frames.
* Revised get\_image function to create a more detailed visual showing what the number is supposed to be in addition to the visual of it.

1. Creating a linked list to hold the models (New, data structures sand algorithms enhancement)

* Created a linked list to house all metrics of all the models to allow for comparison later.
* Created a sort function based on bubble sort which organizes the linked list based on a given classification metric

1. Model 0

* Relabeled original model to this, created a new evaluate\_classification\_metrics function to produce its full accuracy metrics.

1. Model 1 (New)

* Created a new neural network from scratch and trained it.
* Created plot\_model\_scores function which accesses new model metrics dictionary in order to create bar graphs comparing accuracy metrics.

1. Model 2 (New)

* Created another neural network, adjusting the layers of the previous one. Trained it and analyzed performance.

1. Model 3 (New)

* Created new neural network which imports mobilenet\_v2 architecture. Reshaped input and output to match the notebook’s problem. Trained it an analyzed performance.
* Created a new algorithm to convert greyscale image matrixes to RGB format to match mobilenet\_v2’s intended image format.

1. Model 4 (New)

* Created function to hypertune best performing model to improve it using RandomSearch
* Accessed new data structure built to house metrics of all previous models to determine the best performing one by creating a bar graph.

1. Summary (New)

* Added conclusion section to summarize findings and justify why the best and worst models performed as they did.

**Reflection**

The biggest challenges in enhancing and modifying this artifact were modifying the greyscale images to be RGB for the mobilenet\_v2 model, as well as hypertuning one of the models. Modifying the images was challenging as reshaping the array to have the right dimensions proved to be very difficult. Eventually I researched the opencv library in Python which helped me get the arrays in the right dimensions, but before that I was running into a lot of errors while reshaping them.

Hypertuning was a challenge as it was done in a different way than I had when hypertuning previous neural networks. In previous notebooks, I hypertuned by changing the trainable and non-trainable layers of imported models. In this one, I instead researched using RandomSearch to modify the hyperparameters of a neural network I built by hand. As I had not done this before, it required careful reading of keras’ RandomSearch functions to get it working properly. Both the reshaping and hypertuning led me to learn a lot about two machine learning libraries, and the reshaping got me a lot more familiar with modifying the shape of multidimensional arrays.

I incorporated feedback in the process of enhancing this project by changing the data structure which was used, as well as implementing a custom sorting algorithm. In my original enhancement I used a dictionary to house the various models and did not include the new sorting function. The issue with using a dictionary was that it is a built-in data structure and does not fulfill the objectives of this course. I fixed this by instead creating a linked list to hold the various models in the revised enhancement. This aligned more with the objectives of this enhancement.

In my original enhancement, no sorting algorithm was implemented. I had misunderstood the assignment and believed the greyscale to RGB conversion function would be a suitable algorithm, but this was not the case as it did not follow the architecture of sorting/searching algorithm. In the revised version, I included the previously mentioned sorting function to fix this.

The course objectives I planned to meet with this enhancement were “design and evaluate computing solutions that solve a given problem using algorithmic principles and computer science practices and standards appropriate to its solution, while managing the trade-offs involved in design choices” and “demonstrate an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals”.

I believe I have met the first of these goals using the new custom search function. It is built on the bubble sort architecture and works to effectively sort the new linked list. While other algorithms such as quick sort and merge sort may be faster, for this specific problem of sorting the linked list they were not necessary. The linked list contains less than 10 models, and there is no reason to believe it would later be expanded to include thousands of models. Due to the small size of the linked list, a bubble sort algorithm is an ideal solution as its simplicity is easier to understand by others working on the project and introduces fewer opportunities for error. In a small linked list, the sorting computation time will not be noticeably slower than a sorting algorithm with a faster Big O time complexity. Therefore, the sorting function which was built on existing algorithmic principles helped to accomplish a solution specific to this project’s problem.

Regarding the second of the mentioned goals, I believe this was met through the new linked list and the new models. A linked list is a well-founded data structure and using it to house the models as well as their metrics accomplished the task of determining the most ideal model. By designing each node to house both the model and its metrics, it made it so this data structure could both house the models and easily be sorted based on the desired metric.

In addition, the addition of all the new models helped to accomplish this course objective. Computer vision models based on neural networks are a newer and innovative field of computer science, and implementing a diverse selection of them, both with designing the models by hand as well as utilizing transfer learning, showed the ability to use cutting edge techniques to create an accurate model. The remaining three objectives were accomplished in the other artifact enhancements.