Project Overview

Our team sought to analyze the effectiveness of three machine learning libraries for image classification. Our goal was to compare the differences in classification accuracy among the three models we created, as well as a pre-trained model called “MobileNetv2”. The images used in the classification were art datasets from nine distinct styles, or time periods.

Comparison Overview

ScikitLearn Comparison

The results that are shown here were derived from the standard vector machine in SKLearn. The accuracy of this model …... A few of the challenges that our team faced were resizing and loading truncated images for processing. We also had difficulty running tests on our datasets, since we had multiple directories that we needed to read in. In order to combat these issues, we utilized “SKimage.io import imread\_collection,” which allowed iteration through multiple directories. We also used “SKimage.transform” for our scaling issues. In addition, we also needed to wrap all of our data in “rgb2gray” to reduce the number of dimensions of certain images. The dimensions need to match for every image in order for the model to run correctly.

Tensorflow Comparison

The results that are shown here were derived from the convolutional neural network in TensorFlow. The accuracy of this model was approximately 75 percent for training and 65 percent for validation accuracy. The challenges we faced in building this model were dataset size, training and validation split, overfitting, and debugging the model for maximum efficiency. By adding “dropouts,” which delete a specified amount of data, and shuffling we were able to decrease the amount of overfitting to get more accurate results. The other issues were combatted by moving data from training to validation in a 70 to 30 split. This increased our model efficiency and allowed for more accurate reporting.

Keras Comparison

The results that are shown here were derived from the convolutional neural network in Keras. The accuracy of this model is….. Our team found that Keras was the most user-friendly in terms of building, compiling, and running the model. Having dealt with the challenges in TensorFlow, we were more prepared when working with Keras. This resulted in our team not facing many, if any at all, challenges when using this library. We also found that this machine library seemed to provide more functionality in terms of image processing.