

Our project aims to classify the artist of a painting and transfer the image style using convolutional neural networks (CNN). We developed two web page apps with Gradio: the [Artist Classifier](#) and the [Artist Style Transfer Tool](#). The first app takes an input painting and classifies the artist, while the second app takes an input image and outputs the stylized version based on the artist that the user chooses.

Six artists were chosen for this project: Claude Monet, Vincent van Gogh, Leonardo da Vinci, Rembrandt, Pablo Picasso, and Salvador Dali. We downloaded images of their art from a combination of Kaggle datasets and the WikiArt database. We manually cleaned the dataset by deleting images that are not paintings (sculptures, posters, photos, etc.), as well as ensuring that there were no duplicate images. We then divided the data into three parts: 20% for training, 10% for validation and 70% for testing. The training and validation sets are small because there is a finite, limited amount of paintings by those artists. If we use a large portion for training, it will be very likely that the painting submitted by the user to the art classifier app is one of the training paintings, which defeats the purpose of using machine learning to classify the artist.

For the model training, we tried five pre-trained CNNs from pytorch: EfficientNet, ConvNext, MobileNet, ResNet and VGG16. We fine tuned the model via transfer learning techniques, which drastically reduced the training time compared to training models from scratch. We also used a free GPU on Google Colaboratory to speed up the training process. All models were trained with the same training/validation/test set. They perform similarly for different artists, with da Vinci seemingly the most difficult to classify in all models. Individual model accuracy ranges between 83% - 88%. We implemented a random forest voter model, but found that it failed to increase the performance above that of the top performing CNN model. However, we found that implementing a hard voting model led to an increase in accuracy, up to approximately 90%.

The same techniques used by pre-trained CNNs to classify different images based on identification of features of their artist's style can be used to transfer those features to other images. More specifically, a model can be fine-tuned on an image to learn its style, and then that style can be applied to a new image. A popular tool for this is Magenta, which is available in TensorFlow_Hub and is based on the framework of the MobileNet CNN, and we used it to build the artist style transfer tool.

In the future, we plan to include more artists and art formats. We will also improve our apps by optimizing the run time, adding a "style recommendation" function and developing a mobile version of the app to allow users to classify images directly from the phone's camera or gallery.