**Problem Statement**: Can images of paintings be classified according to style or artist using a computer model?

Paintings have been historically classified according to their art style (based on period, school or movement) and over the rich history of visual art over the last few centuries, dozens of such styles have emerged from across the world. However, this can be overwhelming for the layperson interested in learning more about a particular painting or the field in general. A classification tool can help them to learn the artist or style of a painting they like and continue their engagement with art. Hence, it can be used as a tool to promote public education and awareness of art. In more practical terms, the tool in a more developed form can be used by art professionals, such as art dealers and appraisers, to help automate or double check painting processing and fraud detection.

**Data Source**

Several institutions have recently made available images of historical paintings available for public viewing and downloading on their websites. WikiArt is one the earliest and largest databases. However, gathering these images would require scraping these websites. The Kaggle competition Painter- by-Numbers (<https://www.kaggle.com/c/painter-by-numbers>), carried out this activity from WikiArt and compiled into a large dataset of 103,250 unique images (about 37 GB in size). Each image is available in jpeg format in widely varying sizes and resolutions (all color). Also included is a csv file listing each image with file name, title, artist, style, genre, pixels and date.

**Preprocessing and EDA**

Details are in provided file ‘Capstone\_EDA.ipynb’. The primary route of interacting with the data structure was through the csv file containing all the image metadata. Several of the columns, particularly for date and genre, had null values. For the two main classification targets, style and artist, rows containing null values in the corresponding column were removed (a small portion of total rows). This was done separately for each target, as different rows had different null columns.

Next, rows with excessively large pixel resolution or file size were also discarded. Finally, the distribution of unique classes and their image count was plotted. First for style, there were 136 unique styles present (Figure 1). However, over half of these had very few example images less than 500 in number. Image classification learning typically is said to require in the thousands for good learning and can be quite computationally taxing. Furthermore, a large number of classes makes modelling difficult. Therefore, it was decided that only the top 20 categories in terms of example images available would be modelled for classification. This resulted in the 20 styles shown in Figure 2.

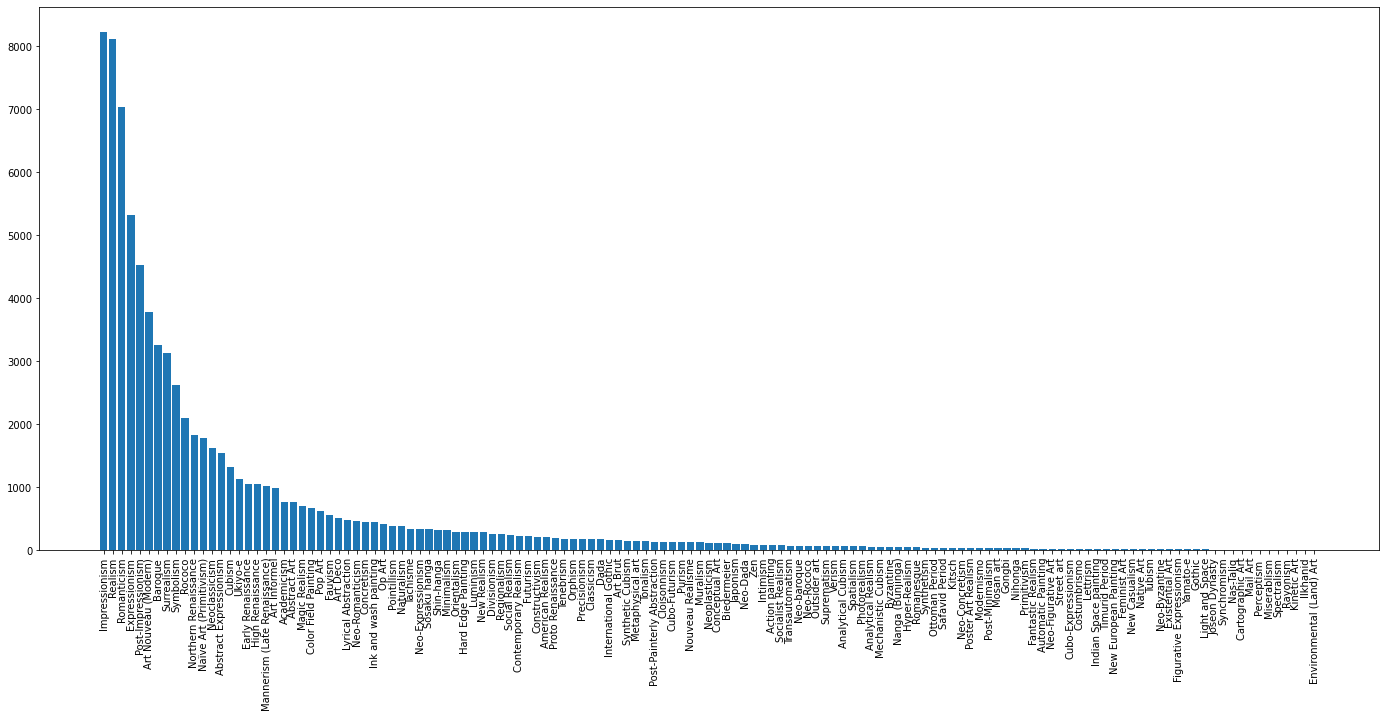
As can be seen, there is still a great deal of imbalance, but it is not practical to utilize thousands of images for the more numerous classes. Therefore, an equal number of images was randomly drawn from each style, thus removing imbalance issues as well as computational practicality ones. Beyond this, as the raw data was in form of pixel values for each image, not much preprocessing or data wrangling was possible other than resizing of the images (all were resized to equal square dimensions of at first 224x224 and then 320x320). A further filter issue was images with extreme dimension ratios (very long and narrow) which might not perform well after being reshaped into a square. However, after filtering for this (where discrepancy was not beyond 1:3), there was little change in model accuracy observed.

Figure : Art Style Classes and Counts

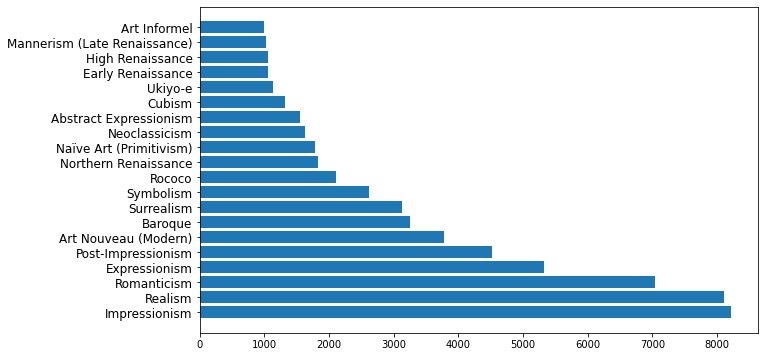


Figure 2: Top 20 Art Style Classes and Counts

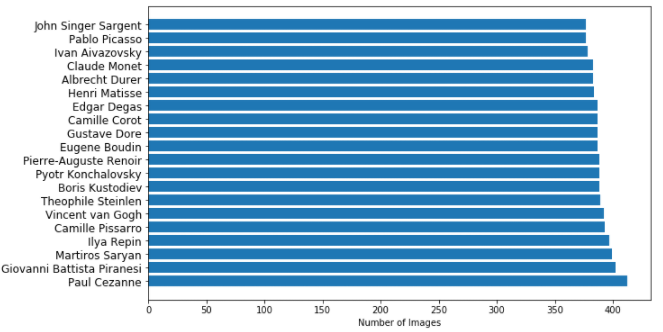


Figure 3: Top 20 Artist Classes and Counts

The same top 20 ranking was carried out for artists, which had much less class imbalance as seen in Figure 3. However, the total number of image examples was much lower and so all images for each class were used in model training.

**Modelling Process**

Convolutional Neural Networks (CNNs) were chosen as the only feasible option to model essentially an image recognition problem. Initial modelling was carried out on local machine (which was able to train models with 7 minute epochs) and a CUDA installation for GPU utilization was also successfully implemented (this was prone to frequent crashes and so abandoned). Eventually Google Colab was opted to process the large image data (the default CPUs were far slower than local machine, switch was made to GPUs on Colab). CNN modeling was done with Tensorflow Keras.

Style Classification

A first baseline model was constructed from scratch with 3 convolutional layers (each followed by max pooling and dropout), two ReLu layers before a softmax final layer. This model increased in accuracy too slowly, so it was decided to do transfer learning with pre-trained CNNs which are suited for datasets with few examples per class. Several nets were tried: ResNet50V2, Xception, NASNetMobile and EfficientNetB0, with two ReLu layers added at the end. The last two produced the lowest or most slowly improving validation accuracies and were not considered further.

Xception performed the best but only reached validation accuracies of about 30% (ResNet having slightly lower accuracy in the 20s), even after unfreezing and retraining of later layers (being careful to unfreeze entire blocks together and not unfreeze batch normalization layers). After much trial and error, it was discovered that swapping the last two ReLu layers with max and global max pooling layers (with no flatten needed) improved ResNet50V2 validation accuracy to now about 40%. However, testing accuracy once again dropped to low 30s.

Further improvements were again not produced by layer unfreezing. Additionally, a cycling learning rate algorithm was tested on the Xception model; this produced drastically lower validation loss but also caused a drop in accuracy. This lower loss model was considered useful for image style transfer applications. Overall, despite adjustments to learning rate and batch sizes, overall accuracy remained disappointing.

Artist Classification

Next, the same process was repeated for artist classification, but with only Xception and ResNet50V2 CNNs. The ResNet model once again outperformed Xception, utilizing the same pooling layers at the end instead of flatten and ReLu. Validation accuracy for the ResNet model reached around 75%, drastically higher than for style classification. This translated to 71% test data set accuracy. Test data top-3 accuracy was about 89%, a respectable number meaning the right artist was in the top 3 guesses 89% of the time. Once again, unfreezing layers and retraining did not provide any benefits.

Precision and recall scores, along with a confusion matrix, were generated (Figures 4 and 5).

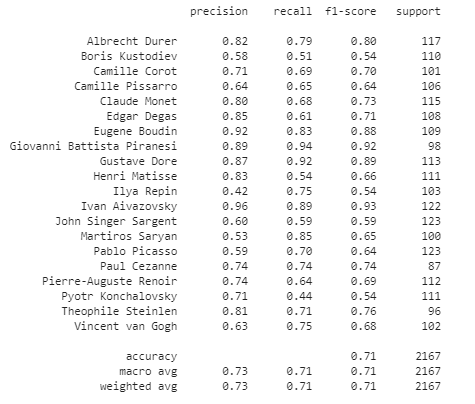
The best classified artists seem to be: Piranesi (sketch like art), Aivazovsky (many water dominated landscapes in realistic style), Dore (etchings) and Durer (distinctive early period). On the other hand, Sargent and Kustodiev have both precision and recall as low values, meaning they are likely difficult to characterize uniquely. Picasso, Saryan and Repin stand out for having much higher recall than precision, meaning they are often being predicted as artists for others' work. This could point to a wider variety in their painting styles (Picasso especially seems to have widely differeing styles on inspection). Matisse seems to have the opposite problem where very few of his works are being classified as such, those that are have fairly high precision. This did indeed seem to be the case where independently downloaded works of Matisse were often predicted to be Picasso works (often using similar color and shape outlines).

Figure 3: Precision and Recall for Artist Classification

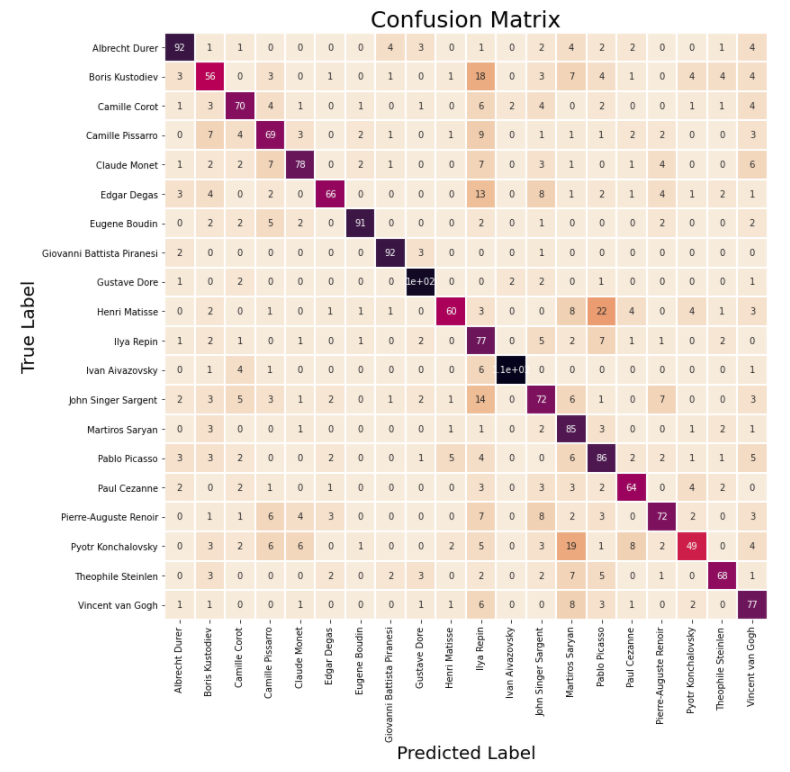


Figure 5: Confusion Matrix for Artist Classification

Art Style Transfer

Another engaging aspect of models trained on paintings is the ability to transfer the style of one image onto another, where the intermediate layers of a CNN model are extracted to transform the image. This is typically done using VGG19 or VGG16 CNNs, which are quite old now and they are used in their pre-trained form with no retraining. A natural improvement would be to use my trained models to extract these layers; as the models had been trained specifically on paintings versus the photograph object recognition training of available CNNs, it was theorized they would perform better in painting style transfer (even with the low accuracy style classifier). However, it was found that the ResNet model layers, after much trial and error, still produced fault images with a deep yellow tinge across them. Literature research suggested ResNets & many complex CNNs are difficult to use in style transfer while VGGs still work best due to their simple sequential architecture.

Summary and Next Steps

* Paintings can be decently classified by artist (accuracy in the 70s) whereas they are quite difficult to do so for style.
* Artist classification can be trained with few images per class (<500)
* ResNets are not suited for image style transfer.

Focusing on artists, a natural extension would be to add more artists, potentially using a knowledge distillation type technique to avoid having to retrain each time & using just a few example images. Combining artist and style prediction into a single multilabel prediction model can also be carried out.