

MAIS 202 Deliverable 3: Who's the Artist?

Agnes Totschnig

1 Final Training Results

Since the previous deliverable, I have tried out width and height shifts as well as rotations in the preprocessing step of the data. This seems to increase the final accuracy by a small amount. Furthermore, I have successfully integrated the class weights, where a class with more paintings gets a smaller weight according to the following formula:

$$w = \frac{\text{total}}{\text{nb_artists} \times \text{paintings}}$$

with **total** being the total number of paintings, **nb_artists** the number of artists and **nb_paintings** the number of paintings in a given class. This didn't seem to lead to better training results, but the calculated accuracy is now a better representation of how well the model is actually doing.

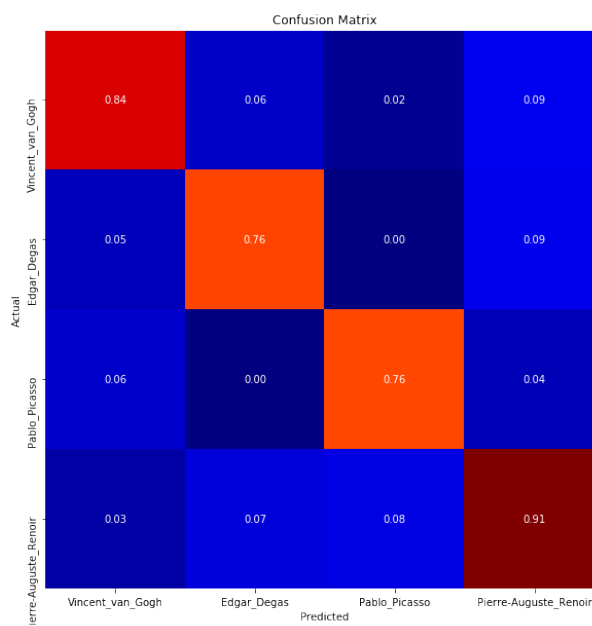
Moreover I tried different batch sizes between 16 and 64. Increasing the batch size by a small amount allowed the model to train and converge faster, but increasing the batch size further lead to lower final accuracy of the model. Thus I decided to work with batches of 32 paintings.

At this point the model was performing too well on the training set, so I tried using **dropout** layers, but this caused the validation accuracy to drop. So I decided to remove them.

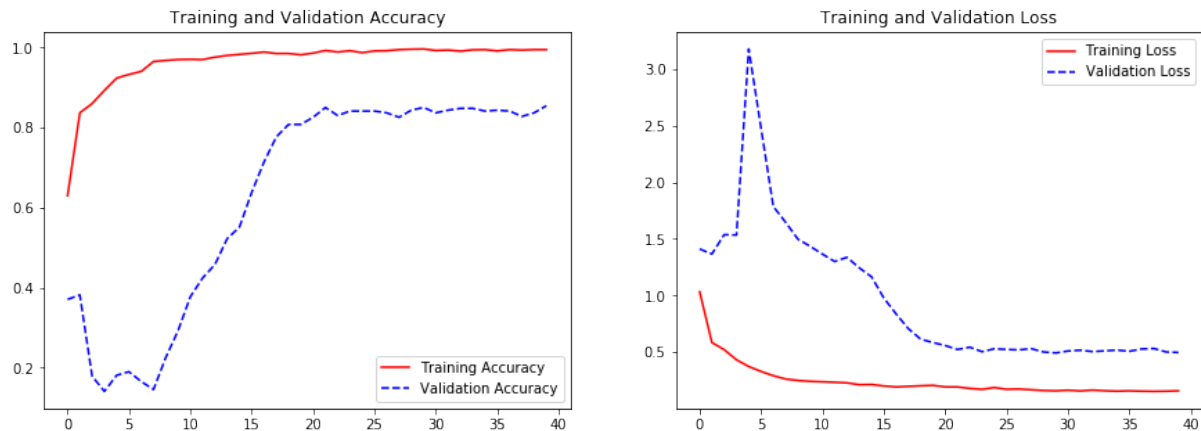
The final model has a prediction accuracy of almost 100% on the training dataset and 82.3% on the validation dataset.

The confusion matrix on the right shows that the model performs the best on paintings by Pierre-Auguste Renoir, while it has the most difficulties with Edgar Degas and Pablo Picasso.

If I had chosen artists with more differentiated painting styles, the results would probably have been better, as it would have been easier to tell them apart.



The plot of the validation loss shown below has a sharp peak at the fifth epoch. I don't know why that is the case. Otherwise, the graphs have the expected shape: the validation accuracy is first very low and eventually rises to a plateau just above 80%. During the training of the model, the learning rate was reduced twice.



The classification report indicates that the model has a better recall for paintings by Renoir, but better precision for paintings by Degas and Picasso. Combining both metrics into an f1-score, we see that the model does comparably well on the four classes.

Classification Report:				
	precision	recall	f1-score	support
Vincent_van_Gogh	0.86	0.84	0.85	128
Edgar_Degas	0.85	0.76	0.81	84
Pablo_Picasso	0.89	0.76	0.82	62
Pierre-Auguste_Renoir	0.80	0.91	0.85	163
accuracy			0.84	437
macro avg	0.85	0.82	0.83	437
weighted avg	0.84	0.84	0.84	437

This concludes the training and hyperparameters fine-tuning of the model proposed in [5], so I saved the weights to integrate them in a website.

2 Final Demonstration Proposal

I will deploy my machine learning model on a landing page type website to showcase my results at the project fair. The user should be able to upload a painting and see the prediction given by the convolutional neural network. This will also allow me to test my model on a random new painting searched on Google. In addition to the name, I would like to have a short biography of each artist.

I will build a static website, so I can host it on Github at no cost. As suggested at the 2nd workshop, I will be using TensorFlow.js to import my trained Keras model in the browser. Moreover I will be looking into Flask applications and Jinja for the frontend.

References

1. *Best Artworks of All Time* database from <https://www.kaggle.com/ikarus777/best-artworks-of-all-time>
2. *Art Challenge - A Quiz Game of Famous Painters* by artchallenge.ru
3. Nitin Viswanathan, *Artist Identification with Convolutional Neural Networks*, Stanford University
4. Alexander Blessing and Kai Wen, *Using Machine Learning for Identification of Art Paintings*, Stanford University
5. Supratim Haldar, *DeepArtist : Identify Artist from Art* on <https://www.kaggle.com/supratimhaldar/deepartist-identify-artist-from-art>