CS230

CS230 Project report: Chronological determination of artwork

Yunchong (Richie) Wang
Physics Department
Stanford University
ycwang19@stanford.edu

1 Problem description

Artists (painters) born and working in a certain period in history usually establish or possess specific styles of work, e.g. impressionism, medieval oil paintings, modern abstract paintings etc. Every historic period is usually dominated by a few leading artists of a certain style (e.g., Vincent van Gogh and Claude Monet for impressionism in the 19th century) that vastly influences painters in the same age, artworks created in certain chronological periods might be clustered in style space. Therefore in this project I will investigate whether training a convolutional neural network with paintings of known composure time can make accurate predictions for the composure period of an unknown painting.

2 Baseline model

In the Milestone report, I discussed in detail the training, testing, and performance of the baseline model for this project. The baseline model implemented weights from hidden layers of RESNET-18 [1] [1] pretrained on ImageNet, and conducted image classification using the pretrained weights on a small set of training images. The images were divided into roughly 100-year chronological periods, totaling 9 choronological classes. The 1000-class fully-connected output layer was replaced with a 9-class fully-connected softmax output layer for my classification problem. Since the aim was to establish a simple baseline model that does better than a random guess of the image period (11%), only ~ 200 training images were used. In the end, the baseline model achieved a training accuracy of $\sim 75\%$ and validation accuracy of 45%, already doing better than a pure random guess. In the following, I will discuss in more detail how I improved the model upon the baseline. I will mainly focus on reducing model bias by freeing up the hidden units and retraining the entire network with my selected dataset, while exploring different hyperparameters to improve training accuracy. I will also demonstrate how I reduce model variance by training on a larger dataset and adding regularization to the model.

3 Dataset and prepocessing

The parent dataset that I have chosen is the Painter by numbers [2] dataset from Kaggle, which contains ~ 80000 paintings from the 1000s all the way to the post 2000s. Since each individual RGB image has a different resolution dimension, the training and validation set images are randomly cropped to 224×224 in terms of resolution to fit properly into RESNET-18 as input and testing images. Random horizontal flip as data augmentation is applied, while data normalization is performed with mean and scatter of [0.485, 0.456, 0.406] and [0.229, 0.224, 0.225] as the default settings of RESNET-18.

[1] https://pytorch.org/hub/pytorch_vision_resnet/

[2] https://www.kaggle.com/c/painter-by-numbers/data?select=train.zip

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

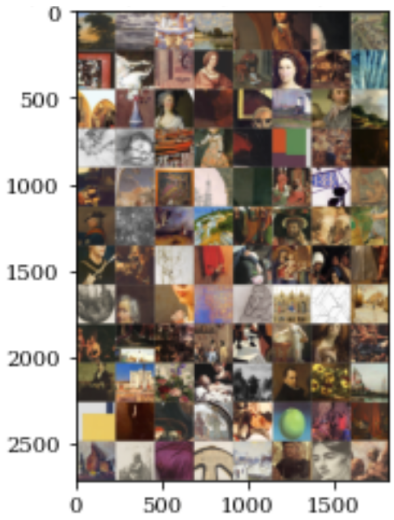

Figure 1: This is a typical set of input for the ResNet at training stage, i.e. an example collection of 96 training images in one mini-batch. Each RGB (3 channels) image is randomly cropped into 224×224 pixels with random flipping applied. The x and y axes labels are the cumulative pixel counts in the two directions for this mini-batch.

For the initial baseline model, I selected ~ 200 images from the parent dataset and divided them into 9 chronological periods, i.e. 1000-1300, 1300-1400, 1400-1500, 1500-1600, 1600-1700, 1700-1800, 1800-1900, 1900-2000, and 2000-now. Specifically, this includes 9 training images for 1000-1300, 16 training images for 1300-1400 and post-2000s, and 24 training images for the other 6 century-long periods each, making a total of 185 training images. In the following, I will refer to this training set as **TrainingSet-1**. The validation set was also kept to a small number, with 3 test images for 1000-1300, 4 test images for 1300-1400 and post-2000s, and 6 test images for the other 6 century-long periods each, totaling 47 test images. The specific model parameters As mentioned above, training with only the fully-connected layer freed up on this dataset resulted in $\sim 75\%$ training accuracy and $\sim 45\%$ validation accuracy, which is a high-bias-high-variance model.

From the baseline model, I first tried out two modifications. The first step is to free up the entire RESNET-18 model and retrain all the hidden layers along with the modified fully-connected layer using my dataset. On **TrainingSet-1** this made negligible difference. However, I kept this architecture of retraining the entire network in all the subsequent explorations, including the final version of my model. The second step is to expand the training set, and I immediately noticed that the periods of 1800-1900, and 1900-2000 had way more paintings than the other periods. Nonetheless, I tried training the model on an unbalanced dataset, with ~ 1000 images for the 1800-1900 and 1900-2000 classes, while keeping the ~ 20 training images fixed for all the other classes. This resulted in much lower training ($\sim 50\%$) and testing ($\sim 30\%$) accuracy, which can be attributed to the network being dominated by the two classes with lots of training samples and making poor guesses for paintings in other sparsely-sampled chronological periods.

With these insights, I created a more balanced training set for exploring hyperparameters of the model. I combined the 1000-1300 and 1300-1400 classes to make a better-populated training class '1000-1400' which reduced the total number of classes from 9 to 8. This resulted in a training set with ~ 200 images for each of the 8 chronological periods (~ 1600 in total) and I will refer to it as **TrainingSet-2** in the following. After an exhaustive exploration of hyperparameters (details in the next section), I was able to overfit to **TrainingSet-2** and achieve a training accuracy of $\sim 95\%$. I was left with a low-bias-high-variance model with validation accuracy of $\sim 60\%$. To reduce model variance, I finally expanded the training set to ~ 10000 images, with ~ 200 images for 1000-1400 and post-2000s, ~ 1200 images for 1300-1400, 1400-1500, 1500-1600, 1600-1700, 1700-1800, and ~ 2000 for 1800-1900 and 1900-2000. This is my final training set which is reasonably balanced between different classes and I will refer to it as **TrainingSet-Final** in the following. The final **validation set** consists of 12 images for the classes of 1000-1400 and post-2000s, and 30 images each for the other 6 chronological periods, totaling 204 test images.

4 Learning architecture and hyperparameters

For the baseline model, I started with fixed hidden layer weights and biases from RESNET-18 pretrained on ImageNet and only freed up the final fully-connected (FC) layer which originally had 1000 outputs. I rescaled the 1000-class FC layer to the initial 9-class FC layer. A softmax function is applied to normalize the output probabilities and determine the inferred class (chronological period) for each image. The model was trained locally on my laptop with CUDA acceleration on Nvidia GTX1650 (4GB GPU RAM). The hyperparameters for the baseline model are as follows:

- Loss function: Cross entropy loss on 9-class softmax output
- Optimizer: Stochastic Gradient Descent with momentum
- Mini-batch size: 4
- Learning rate α:0.001
- Momentum β: 0.9
- Number of epochs: 25
- Learning rate decay: 0.1 for every 7 epochs

From the baseline model, I first freed up the hidden layers of RESNET-18 during training, and saw no significant improvement on **TrainingSet-1**. Then, directly applying this model to **TrainingSet-2** saw a slight improvement, resulting in $\sim 80\%$ training accuracy and $\sim 55\%$ validation accuracy. From this point, I started exploring different hyperparameters trying to overfit to **TrainingSet-2**. I changed the optimizer from Stochastic Gradient Descent (SGD) to Adam with weight decay, with Adam being better suited for a large set of parameters when the entire RESNET-18 is re-trained and also having the conveniently built-in L2 regularization. Using Adam required much smaller learning rates than SGD, and I decreased the learning rate from 0.001 to 0.0003. Below $\alpha = 0.0003$, the training accuracy did not improve significantly, and I kept it at 0.0003. I also decreased the learning rate decay from 7 to 5 epochs, resulting in slightly finer learning steps in later epochs than the baseline model. While training on **TrainingSet-2**, I found that increasing the mini-batch size would improve the training accuracy, while increasing the weight decay parameter that increases the effect of L2 regularization improves the validation accuracy. As for the number of training epochs, the loss of the training and validation phases usually converge after ~ 15 epochs, and increasing the training epoch to 50 did not improve or worsen the model accuracy in the extended epochs. Therefore I decided to fix the number of training epochs to 25.

With these hyperparameter explorations, I achieved $\sim 95\%$ training accuracy and $\sim 60\%$ validation accuracy on **TrainingSet-2** with mini-batch size of 64 and weight decay parameter of 1×10^{-5}. For the final model trained on **TrainingSet-Final**, I tuned the hyperparameters further to achieve better training and validation accuracy by increasing the mini-batch size to 96 and weight decay parameter to 2×10^{-4}. A demonstration of 96 input images in one mini-batch is presented in Fig. 1. The final model hyperparameters are summarized here:

- Loss function: Cross entropy loss on 8-class softmax output
- Optimizer: Adam (with weight decay) [3]
- Mini-batch size: 96
- Learning rate α:0.0003
- Number of epochs: 25
- L2 regularization weight decay: 2×10^{-4}
- Learning rate decay: 0.1 for every 5 epochs

As a final note, I also explored a more complicated model, i.e. RESNET-34, and saw improvements of $\sim 5\%$ over RESNET-18 both trained on **TrainingSet-2**. I wish I could have tested out RESNET-34 on **TrainingSet-Final**, but I was limited by the GPU RAM on my laptop and could not carry out the experiment. This might be worth further exploring as future improvements to the current model.

[3] `https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html#torch.optim.AdamW`

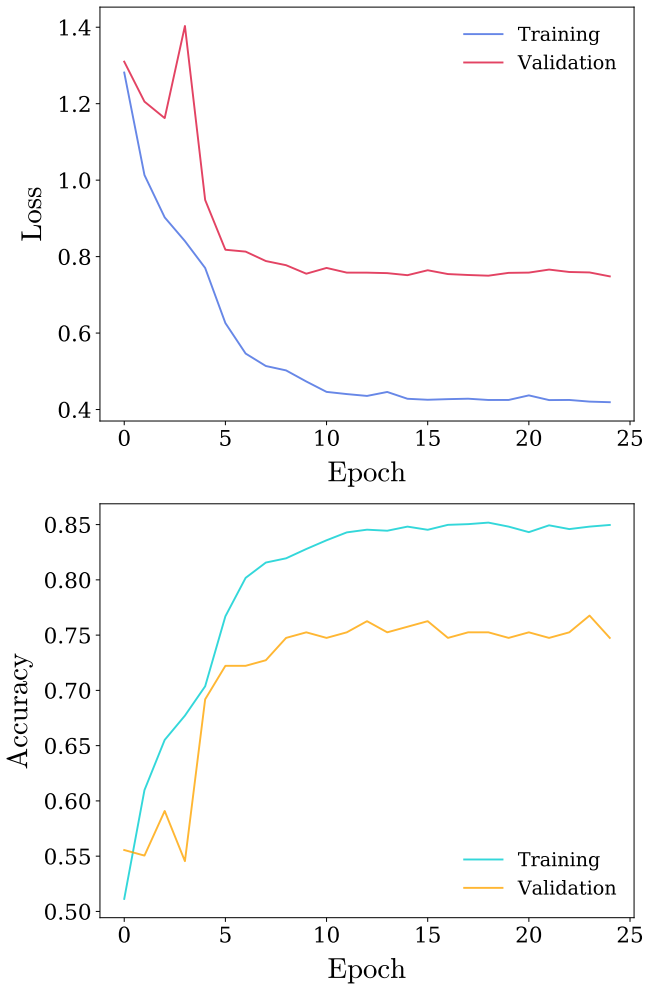

Figure 2: *Top panel*: The loss value for the training and validation datasets as a function of training epoch. Bottom panel: Training and validation accuracy as a function of training epoch.

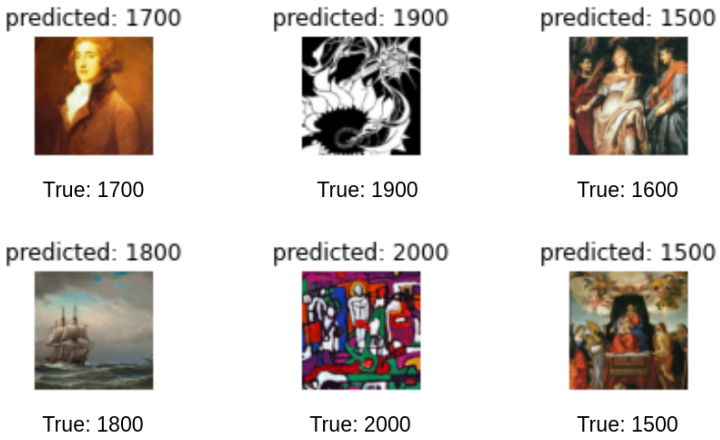

Figure 3: These are 6 example test images from the validation set. The label of each image stands for the starting year of its chronological class (e.g., 1500 stands for 1500-1600 and 2000 stands for 2000-now). Five out of the six images have their chronological period correctly predicted by the model, and the wrongly predicted sample is only off by one chronological class (one century), demonstrating the robustness of my model.

5 Results and discussion

The total loss (upper panel) and model accuracy (lower panel) during training and validation as a function of training epoch is shown in Fig. 2. The loss and accuracy of the model is well-converged after $\sim$ 15 epochs. The final model achieved a maximum training accuracy of $\sim 85.2\%$ and validation accuracy of 76.8%, which is a reasonably low bias and low variance model. The decrease training accuracy and significantly improved validation accuracy compared to the model trained on **TrainingSet-2** is a combined effect of expanded training set and increased regularization effect, which eased the overfitting and reduced the model variance. Therefore, we have arrived at a model that can efficiently determine the composure period of artworks down to the century-level created by different artists in different historical periods, doing way better than a random guess ($\sim 10\%$ accuracy). As a demonstration of the final model's capability, I show six image predictions from the validation set in Fig. 3. Only one image is classified wrong, and the predicted period is off by only one century of its real composure time, justifying the robustness of the final model's predictive power.

6 Summary and outlook

In this project, I have developed a deep learning image classification model that can robustly determine chronological period of individual artworks' composure time down to century-level accuracy. The model is based upon the widely used residual-convolutional neural network RESNET-18, with pretrained weights on ImageNet and fully connected layer modified. The model takes in 224×224-pixels RGB images and outputs 8 century-long chronological class predictions for each image. By retraining the entire RESNET-18 model on ~ 10000 paintings from the 'Painter by Numbers' dataset that span the historical period from 1000s to post-2000s, I have arrived at a low-bias and low-variance image classification model with 85.2% training accuracy and 76.8% validation accuracy.

The final model builds upon the preliminary success of the baseline model as described in the Milestone report. However, it improved significantly both during training and testing compared to the baseline model benefiting from a myriad of model improvements: freeing up all hidden layers of RESNET-18, expanding the training set, refining the chronological classes, improving the optimizer, exploring hyperparameters, and adding regularization.

Last but not least, there are still many aspects of the model worth investigating in the future. For example, I also attempted training using a bigger network, i.e. RESNET-34, but failed to carry out the experiment due to hardware limitations. However, I did find a slight improvement of training accuracy over RESNET-18 during model testing on a smaller training set (**TrainingSet-2**). In addition, I proposed using a different loss function for my classification problem, i.e. the style loss function in neural style transfer frameworks [2]. However, the style loss is currently unsuitable for my classification problem, as the Gram matrix in the loss function is different for every image. Some kind of 'representative' style for all paintings in the same era is required for chronological determination, which could be much more complicated than just a linear combination of the Gram matrix for different paintings. Another outlook for future research is that whether the model developed in this project can make accurate predictions on finer timescales, e.g., 50-year periods, 10-year periods, or even more precisely in 1-year periods.

7 Contribution statement and Acknowledgement

This project is completed by YW independently. YW would like to thank Mauricio Wulfovich (TA) for many helpful suggestions and enlightening discussions throughout the quarter for continuously improving this project.

References

[1] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[2] Gatys, L.A., Ecker A.S. & Bethge, M. (2015) A Neural Algorithm of Artistic Style. *Journal of Vision 16*, pp. 326. Vision Sciences Society Sixteenth Annual Meeting.