Labeling Paintings with Thematic and Emotional Content

Bardia Beigi & Soroosh Hemmati
Stanford University
450 Serra Mall, Stanford, CA 94305
bardia@stanford.edu shemmati@stanford.edu

Abstract

In this manuscript, we explore three different Convolutional Neural Network (CNN) models on a dataset comprising over 35000 paintings taken from WikiArt. We explore different models, preprocessing procedures, and hyper-parameters to maximize accuracy. Using our customized model, we were able to reach a validation accuracy of 60.2% and a test accuracy of 61.4 % on a group of 6 thematic labels. Given these results, the dataset is examined more carefully and suggestions for improving both the dataset and the classification model are given.

1. Introduction

This project involves labeling a dataset of images of paintings taken from WikiArt. The images are labeled based on theme, artist, style, and genre. There are 35750 images in total in the dataset. The main goal of this project was to develop a neural network capable of reaching high accuracies in theme classification. Three major models were implemented and a number of preprocessing procedures were explored to arrive at significant accuracy enhancement in this project.

This project is motivated by the authors' deep appreciation of visual arts and their attempt to find a way to use deep learning to classify paintings. Due to the rapid digitization of available forms of visual arts, including paintings, there is a need for researchers and art enthusiasts to have access to an automatic system capable of accurately classifying paintings. This is the authors' attempt at such system. Additionally, this project could be beneficial in recommending paintings to users based on their interests.

This manual will explore all measures taken to properly adjust and use the dataset for the learning tasks at hand. We will first explore related work in this area which shed some light on the fronts taken to explore this problem and the current state of the art models used to enhance classification.

2. Related Work

The methods used for this task can be broadly put into two categories. First, there are deep neural network models and second, there are more traditional models using support vector machines and related algorithms in conjunction with manual feature selection to enhance classification.

In the first category, researchers have explored the use of deep convolutional neural networks to identify different aspects of paintings through various techniques such as training from data or retraining pre-trained models on datasets. With the help of CNNs, [8] achieves an 88% recall on artist classification, [1] and [10] obtain state of the art accuracy through extracting regional features as well as global ones. [2] improves the genre classification task's accuracy by using pre-trained models in object and texture classification. [14] uses adaptive weighted matching of the convolutional layers to perform well on identifying similar paintings. Finally, Inspired by the fact that correlation between feature maps describes image texture, [4] employed deep correlation features for paintings style classification.

On the other hand, older papers explored the problem differently. [15] and [5] took a more traditional approach and used MATLAB to extract features of paintings and fed it to a traditional neural network to classify their genres on a much smaller dataset. [12] employed a multi-task dictionary learning approach in conjunction with traditional machine learning tools to classify styles of paintings. [6] managed to obtain a good representation of texture through modeling of local binary pattern operator through traditional image processing techniques.

In [3], the authors utilize general neural networks and Artificial Intelligence techniques to enhance bridge paintings. [13] takes on the task of classifying images with emotional content. [7] employs dual-tree complex wavelet transform, Hidden Markov Tree modeling, and Random Forest classifiers to classify styles of images. [9] obtains 101 high resolution digital versions of paintings from the

Figure 1: The Family of the President by Fernando Botero

Figure 2: Teatime (Self-Portrait with Raphael Sawye) by David Burliuk

Van Gogh and Kroller-Muller museums to help them with traditional artist classification. Finally, [11] uses both global as well as local features to rate paintings on an aesthetic level with labels coming from art-inclined human observers.

3. Dataset and Features

Our dataset consists of over 35000 paintings taken from WikiArt. Each image is labeled with four pieces of information: artist, genre, style, and theme. Our task was to classify images based on theme. Throughout training, we used nearly 80% of the suitable data for training, 10% for validation, and 10% for testing.

There are over 60 theme classes with a highly non-uniform distribution of images per class. Figure 3 illustrates the skew in the number of thematic labels. This dataset contains a lot of peculiar images and even more peculiar labels. For instance, figures 1 and 2 are two examples of images which have the same theme, "male portrait", yet look vastly different and do not necessarily represent a male portrait.

The skew in the distribution of images poses a challenge to any CNN-based learning model. As a result, a fair amount of preprocessing was necessary. We explored a number of options in this quest.

3.1. Preprocessing

For the baseline model explained in the following section, we simply used the dataset as is, with the exception of using only the top 12 most populated classes for theme classification. As expected this is not a suitable long-term plan and as a result, we decided to up our preprocessing game.

Due to the mediocre results of the baseline model on the naive preprocessing we did above, we decided to equalize the number of examples we used for each class. As a result, from the top four classes we selected only 2000 to be represented in the training and validation sets. Furthermore, we subsampled the images in the less represented classes multiple times to get 2000 images for each of those as well. It is noteworthy that all images in the dataset have different sizes. Therefore, we focused only on images of size at least 256×256 and for subsampling, started from taking the center, to the corners, and off-center 256×256 sub-images. Although this perfectly removed the skew in the data, due to the nature of tasks at hand, it posed some challenges which will be explained later.

Due to the challenges posed, we decided to first resize all images to have size equal to $256 \times x$ or $x \times 256$, where x is the larger dimension, and then take the middle 256×256 sub-image for training. This meant having to take an amount equal to the number of images in the smallest of the 12 classes for all classes used in our training. This proved more useful as for most pictures in the dataset, one requires seeing most of the image for classification. For instance,

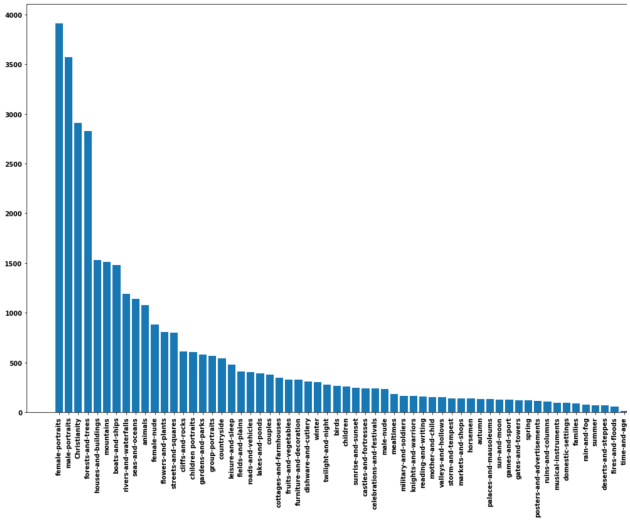


Figure 3: Distribution of thematic labels in the dataset.

two of the top 12 theme classes are ”male portrait” and ”female portrait”. It is clear that a subsample of the picture that does not contain the face or the body of the person would be completely useless in classification. Therefore, resizing is indeed the right way to go.

Finally, since having 12 classes meant having to pick a very small number of images per class (equal to the number of images in the smallest class from figure 3) and many classes are too hard to distinguish, such as ”male portrait” and ”female portrait”, we decided to further reduce our number of classes by combining a few classes into one, such as the aforementioned classes into ”portraits” or completely delete some classes to end up with 6 final classes.

4. Methods

Our approach mainly consisted of establishing a proper baseline model for the tasks, exploring and analyzing the dataset to find a useful plan of action, and to finally, optimize the best models for each task.

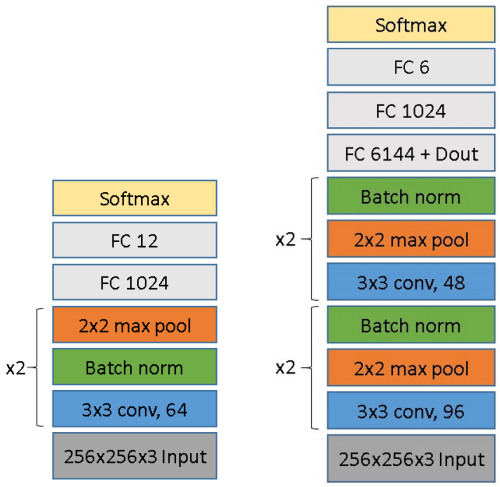
The experimentation mainly occured in the form of removing/adding layers, the number of filters at each layer, as well as tuning regularization, weight decay, learning rate, and dropout probability. Due to the limited amount of resources available, we also explored CNN’s that already worked well on classification tasks, mainly VGGNet and AlexNet.

4.1. Baseline Model

Our baseline model was a simple 2-layer convolutional model. Figure 4a illustrates this model. This model consists of two layers of convolution - batch normalization - max pooling followed by two fully connected layers for classification. It must be mentioned that for the baseline model, we were doing classification on 12 classes. As a result, the final fully connected layer has a 12-dimensional output.

4.2. Main Models

The models we explored in this project are a small version of VGGNet, AlexNet, and a more complex version of the baseline model. These models are illustrated in figures 5a, 5b, and 4b, respectively. Our approach to optimization was to first run these models until they completely overfit the training data, and then optimize hyper-parameters for best results. In rare cases, the number of filters in the convolution layers were switched to explore how an increase or decrease in the size of the model would enhance accuracy. One point that became evident early on was that it was more useful to have larger convolution layers, in terms of number of filters, come first which is slightly counter-intuitive.



(a) Baseline Model (b) Final Custom Model

Figure 4: Models Used During Training

It is worth mentioning that as opposed to some of the papers mentioned earlier, we decided not to hard-code, or develop ways to extract extra features from the pictures and instead let the neural network do all the learning on its own. This approach proved sufficient given our time frame and expectations, however, to significantly increase accuracy, it is necessary to make use of extra features.

4.3. Principles of Operation

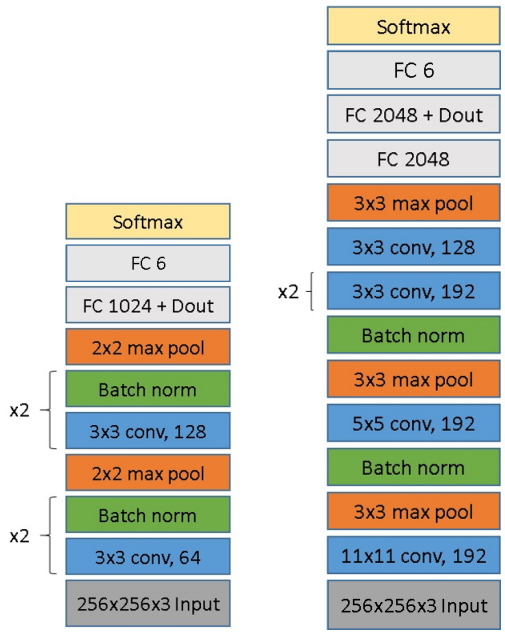
Training and optimization of all these models are similar. All contain a number of convolution, batch normalization, max pooling, and fully connected layers, followed by a final softmax layer which gives the probability vector of the given image belonging to each of the classes. We used an Adam optimizer for training.

5. Results and Discussion

In this section, we will discuss the results we obtained throughout the course of training and explore how those results were optimized.

5.1. Hyper Parameter Tuning

Our general strategy for hyper parameter tuning was to first find a suitable learning rate, then to optimize for weight decay and regularization. In addition, we generally optimized for 10 epochs until the latest stages of development in which we reduced learning rate and ran longer iterations to maximize accuracy. The minibatch size used is 32. This number provides a fair amount of parallelization (and fast processing), yet is small enough for training to not crash due to memory errors. In addition, given the size of our modified dataset, we got enough granularity for our



(a) Mini-VGGNet Model (b) AlexNet Model

Figure 5: Models Used During Training

Model	Validation Accu. (%)	Train Accu. (%)
Custom Model	60.2	65.6
Mini-VGGNet	51.1	56.4
AlexNet	49.6	54.5

Table 1: Final Theme Validation and Training Accuracies

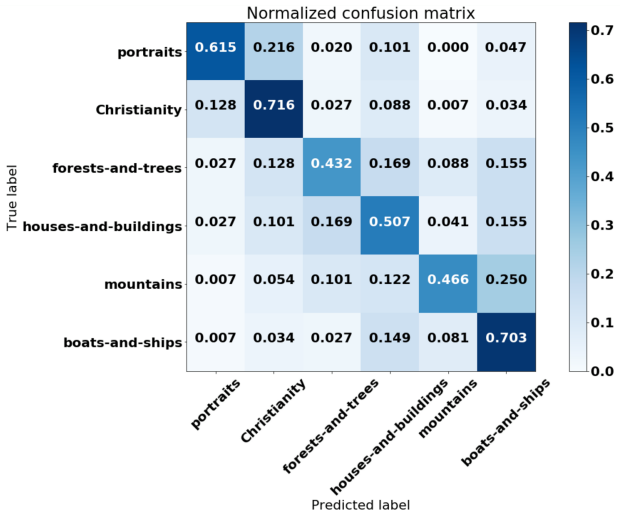


Figure 6: Confusion Matrix for the Custom Model's Output

learning algorithm.

The advantage of using Adam optimizer is its proper use of momentum and its ability to not get stuck at local optima. In addition, Adam optimizer has the ability to pick a direction of change in the parameters that is fairly optimal in terms of global loss minimization and is proven to be a good starting point for all deep learning projects.

Properly adjusting the learning rate is perhaps the first and most crucial step in training any neural network. We carried this task out by first exploring a wide range of learning rates from 10^{-7} to 10^{1}, and then using the bisection method to arrive at a locally optimum learning rate for any first attempt at a given model. After finding the optimal bounds for the learning rate, all future tuning of the hyper parameters was done with minor tweaks of the learning rate, except for cases in which the model was changed enough for the learning rate to require another round of optimization.

Optimization of the other hyper parameters was performed similarly to how learning rate was optimized. However, upon changing the sizes of layers or adding or removing more layers, or changing the model altogether, another complete round of optimization was done.

5.2. Metrics

Our main metric was validation accuracy. However, the results presented here also include precision, recall, and confusion as well. It must be noted that all these metrics served an important purpose in the path we took to complete this project.

As mentioned in the preprocessing section, we first started with classifying the top 12 classes in each category without adjusting for the skew. This resulted in the model getting relatively high train accuracies (above 60%) with an increase in loss. It later became clear, using a confusion matrix, that this was due to the model overfitting the three most populated classes which greatly increased accuracy at the cost of increasing loss for other classes. The confusion matrix can also be used to deduce which classes are too similar to each other in case one decides to hard-code additional features to be used for training.

5.3. Results

Table 1 summarizes the best results achieved using our three models for theme classification. Since the best results came from the custom model, all metrics given hereafter are

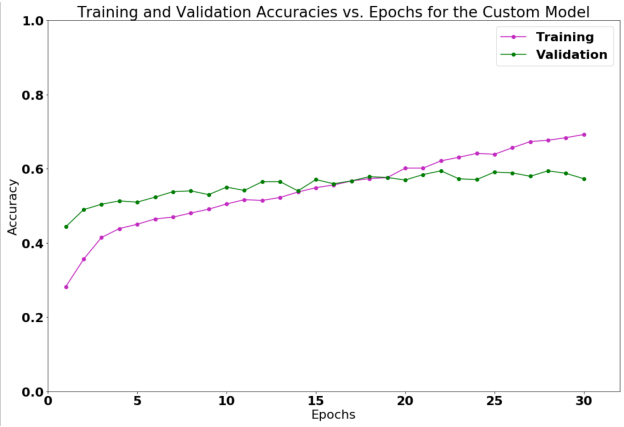


Figure 7: Train and Validation Accuracy of the Custom Model

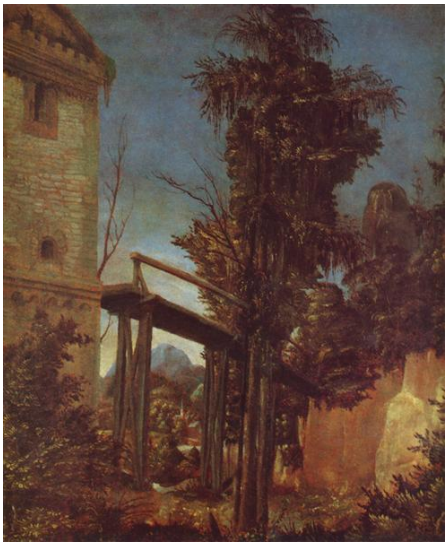


Figure 8: True Label: Forests and Trees. Predicted: Houses and Buildings

taken from the output of that model. Figure 6 is the confusion matrix of the result. These results will be vigorously explained in the next section. Figure 7 illustrates the learning curve of the model under the custom model. Using the model that resulted in table 1, we were able to get a test accuracy of 61.4%.

5.4. Discussion

5.4.1 Outputs of Different Models

Table 1 summarizes the outputs of different models. It is clear that the custom model outperforms the rest in the terms of validation accuracy by about 10%. The reason for this difference is that we spent quite a significant amount of time optimizing this model for the task of classifying themes. Figure 7 shows the learning curve of the model. We used an Adam optimizer with a learning rate of 0.005 and a dropout of 0.9. Clearly, this high amount of dropout led to a low training accuracy early on and it was not until epoch 16 where training accuracy finally caught up with validation accuracy. In addition, it is clear that the rate of increase in validation accuracy was fairly low. Maximum validation accuracy was reached at epoch 21 after which the model began to overfit.

5.4.2 Confusion Matrix

The confusion matrix is given in figure 6. It is evident that the model is fairly successful at identifying ”portraits”, ”Christianity”, and ”boats and ships”. This is also evident in the high recall values we can observe in table 2 for these classes. For the other three classes, the recall values are fairly low which are explained next.

First, let us look at and compare figures 8 and 9. These pictures belong to the ”forests and trees” and ”houses and buildings” classes, respectively. However, in the first picture, there is clearly a building present, and in the second, a fairly large tree. As a result, coming up with the label for these images is first, a highly subjective matter, and second, error-prone for a classification model. This explains the relatively high off-diagonal values in the center of the confusion matrix.

Other areas of concern are the ”portraits” that were misclassified as ”Christianity”, and ”mountains” that were misclassified as ”boats and ships”. Again, the reason for those is fairly similar. A fair portion of ”Christianity” images are portraits of historical religious figures, for instance, figure 10. As a result, those images should belong to both ”Christianity” and ”portraits”. Unfortunately however, only one thematic label was given for each image in the dataset.

Finally, a significant number of ”mountains” -labeled images were classified as ”boats and ships”. Figure 11 is an example of a ”mountains” image. Once again, it is evident that the label is not fully representative of the image and multiple labels should exist for the image. This was a really common incidence in this dataset.

6. Conclusions

In summary, we investigated the success and failure of three CNN models on classifying the theme of a collection of over 35000 images taken from WikiArt. The challenges posed by the dataset included having paintings taken from

Figure 9: True Label: Houses and Buildings. Predicted: Forests and Trees

Theme	Precision	Recall
Portraits	0.76	0.61
Christianity	0.57	0.72
Forests and Trees	0.56	0.43
Houses and Buildings	0.45	0.51
Mountains	0.68	0.47
Boats and Ships	0.52	0.7

Table 2: Recall and Precision Values for Different Classes

a wide array of themes and styles of painting, huge skew in the distribution of images per class, inaccurate labels, and missing labels.

The models investigated included a mini-VGGNet model, AlexNet, and a customized model optimized for this task. The customized model was able to reach a maximum validation accuracy of 60.2% and a test accuracy of 61.4 % at a training accuracy of 65.6%. Further investigation showed that the model was fairly successful at classifying ”Christianity” and ”boats and ships” images correctly. However, there was a fair amount of misclassification in ”portraits”, ”forests and trees”, ”houses and buildings”, and ”mountains”. The reason for this behavior was attributed to inaccurate and missing labels and examples were presented.

Moving forward, it will be useful to have multiple proper labels for each image along with models that can properly learn based on multiple labels given for each image. Furthermore, given more computation power, it will be possible to investigate the effect of more complex classification models such as ResNet and explore how adding extra code to independently extract features from

Figure 10: Mary with the Child by Albrecht Altdorfer. Belonging to the Christianity Class.

Figure 11: Sea Bay by Ivan Aivazovsky. Belonging to the Mountains Class.

the dataset could help with learning.

7. Acknowledgements

We wish to thank CS231N staff for their relentless help throughout this project. We also wish to express our sincerest gratitude to Mana Lewis, and Vinay Chaudhry for providing us with the dataset.

References

[1] R. M. Anwer, F. S. Khan, J. van de Weijer, and J. Laaksonen. Combining holistic and part-based deep representations for computational painting categorization. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pages 339–342. ACM, 2016.

[2] E. Cetinic and S. Grgic. Genre classification of paintings. In *ELMAR, 2016 International Symposium*, pages 201–204. IEEE, 2016.

[3] P.-H. Chen and L.-M. Chang. Artificial intelligence application to bridge painting assessment. *Automation in construction*, 12(4):431–445, 2003.

[4] W.-T. Chu and Y.-L. Wu. Deep correlation features for image style classification. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 402–406. ACM, 2016.

[5] R. G. Condorovici, C. Vertan, and L. Florea. Artistic genre classification for digitized painting collections. *UPB Scientific Bulletin*, 75(2):75–86, 2013.

[6] Z. Guo, L. Zhang, and D. Zhang. A completed modeling of local binary pattern operator for texture classification. *IEEE Transactions on Image Processing*, 19(6):1657–1663, 2010.

[7] S. Jafarpour, G. Polatkan, E. Brevdo, S. Hughes, A. Brasoveanu, and I. Daubechies. Stylistic analysis of paintings usingwavelets and machine learning. In *Signal Processing Conference, 2009 17th European*, pages 1220–1224. IEEE, 2009.

[8] K. A. Jangtjik, M.-C. Yeh, and K.-L. Hua. Artist-based classification via deep learning with multi-scale weighted pooling. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 635–639. ACM, 2016.

[9] C. R. Johnson, E. Hendriks, I. J. Berezhnoy, E. Brevdo, S. M. Hughes, I. Daubechies, J. Li, E. Postma, and J. Z. Wang. Image processing for artist identification. *IEEE Signal Processing Magazine*, 25(4), 2008.

[10] S.-G. Lee and E.-Y. Cha. Style classification and visualization of art paintings genre using self-organizing maps. *Human-centric Computing and Information Sciences*, 6(1):7, 2016.

[11] C. Li and T. Chen. Aesthetic visual quality assessment of paintings. *IEEE Journal of Selected Topics in Signal Processing*, 3(2):236–252, 2009.

[12] G. Liu, Y. Yan, E. Ricci, Y. Yang, Y. Han, S. Winkler, and N. Sebe. Inferring painting style with multi-task dictionary learning. In *IJCAI*, pages 2162–2168, 2015.

[13] J. Machajdik and A. Hanbury. Affective image classification using features inspired by psychology and art theory. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 83–92. ACM, 2010.

[14] Q. Wang, F. Gao, Y. Wang, and L.-Y. Duan. Adaptive weighted matching of deep convolutional features for painting retrieval. In *Multimedia Big Data (BigMM), 2016 IEEE Second International Conference on*, pages 194–197. IEEE, 2016.

[15] J. Zujovic, S. Friedman, and L. Gandy. Identifying painting genre using neural networks.