FINE-GRAINED PAINTING CLASSIFICATION

BY

MANAV KEDIA

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Adviser:

Associate Professor Svetlana Lazebnik

ABSTRACT

A lot of progress has been made in the domain of image classification in the deep learning era, however, not so much for paintings. Even though paintings are images they are very different from photographs and classification of paintings requires in-depth domain knowledge compared to classifying an object. This makes the task of fine-grained classification of paintings even harder. In this thesis, we evaluate the classification of paintings into its various styles, genres, artists and formulate the problem of dating paintings as a classification problem. We experiment with the standard networks available as baselines and then improve the classification models via multi-task learning. We also propose a novel architectural addition to the VGG network to do fine-grained classification. Our models beat the existing state-of-the-art classifiers by a big margin.

ACKNOWLEDGEMENTS

I would like to thank my advisor for her constant support and guidance throughout the duration of my thesis.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

Over the last few years, there has been rapid advancement towards digitization of fine-art collections such as paintings, sculptures, etc [16-21]. These have also been made available to the public for online viewing. These works span from classical to modern and now contemporary paintings. With the growing number of digital artworks, there is a strong emerging need to build systems optimized for the domain of paintings that can automatically classify these works of art into their respective styles, genres, artists, and date them. This is possible since most of these artworks come with the associated metadata. By building classification systems that can accurately classify paintings into the above mentioned classes, our system can easily be extended to build recommendation systems which return visually similar paintings that a user might like to purchase.

Deep learning has made tremendous leaps in a number of image processing tasks such as image classification [22], object recognition [23], scene recognition [24], etc. in natural photographs/images. This has been made possible since cameras have become a mainstream device. However, little attention has been paid to the task of classification of fine-art paintings or if the models built for natural image classification [22, 25] can be extended to the domain of paintings. This may be attributed to the lack of digital datasets of these fine-art paintings as compared to the datasets for natural images of which there are several (Imagenet [28], Pascal VOC [26], CIFAR-10 [27]). On the other hand, very few fine-art datasets are available to extensively evaluate. Of the few available, Khan et al. [14]'s dataset has only 4,266 paintings. Only recently, a dataset was provided by Saleh et al. [3] namely the 'Wikiart paintings' dataset which consists of 80,000 paintings and is considered as the main benchmark for evaluating

models in this domain. Another big paintings dataset was provided by [12] called the 'Your Paintings' dataset and consists of 210,000 oil paintings collected from [21].

We believe the task of fine-arts classification is a more challenging problem compared to object recognition such as cats or dogs. Any individual could look at an artwork and say whether it is figurative (i.e. where the subject of the painting is discernable) or abstract (where it is mainly color, shapes and lines). However, there is so much more to an artwork than meets the eye. It takes a person with strong domain knowledge to appreciate art and to be able to accurately identify the style, artist, genre, era of the artwork. The task of fine-grained classification is even more challenging which we also target in our work.

In this thesis, we first present a detailed analysis of fine-tuning a vanilla CNN architecture namely VGG [25] to the domain of paintings and analyze their performance for the task of large-scale style, genre, artist classification using the Wikiart paintings dataset [3]. To the best of our knowledge, ours is the first work to date paintings using deep learning. We formulate the problem of dating as an n-way classification problem similar to [1].

While our aim is to improve fine-arts classification we also explore really fine-grained fine-arts classification by increasing the number of classes considerably. We then look into the multi-task learning for the same tasks described above which lead to better results. Our models beat the existing state-of-the-art classifiers [2, 3] by a considerable margin. Finally, we present a novel fine-arts specific modification to the VGG architecture which shows promising results over the existing state-of-the-art classifiers.

CHAPTER 2: RELATED WORK

On the subject of paintings, traditional image processing techniques have provided art historians with useful tools [30]. Classifications of paintings has thus far mainly involved low-level features such as color, shadow, texture and edges. Lombardi et al. [31] used these features for artist classification on a small set of artists. Brushstrokes [32] have also been used to identify the artist. [33] used SIFT features within BOW pipeline for artist classification. We use deep learning for fine-grained classification of 194 artists. Artist classification at this scale has not been done before.

Bar et al [34] used features from pre-trained CNN [22] for style classification; however they used a small dataset. Karayev et al [7] explored hand crafted features vs deep learning features for style classification; however their model extracted CNN features and used various classification algorithms on top of it thus not being end-to-end. Saleh et al [3] created the Wikiart dataset and used features ranging from low-level to high-level semantic features with metric learning approaches for the task of style, artist and genre classification. The closest to our work was done by Tan et al. [2] where they fine-tuned a VGG network on the same Wikiart dataset [3] to improve classification scores. However, our models not only improve the classification scores significantly from all the above works but also work quite well on the task of fine-grained classification where we consider 194 artists labels instead of 23.

Our models are based on transfer learning. It has been shown [35] that transferring well-learnt knowledge from a source domain to a target domain leads to improvement in accuracies. Crowley et al. [12] used transfer learning to find objects in paintings using features extracted from a CNN pretrained on ImageNet [22]. Their results showed that transfer learning from the

domain of natural images to paintings is feasible. Crowley et al. [13] later also showed the problem of domain adaptability, where fine-tuning on a dataset of paintings and using the fine-tuned features led to better retrieval of objects in paintings compared to fine-tuned features extracted from a CNN trained only a dataset of natural images. This provides our motivation in fine-tuning/training models on paintings in all our experiments.

To the best of our knowledge, ours is the first work that dates paintings. We formulated the problem of dating paintings similar to [29] who tried dating datasets of cars and vintage clothing images. Ginosar et al. [1] also followed a similar approach where she dated portraits of female students obtained from school yearbooks. Lee et al. [10] used visual data mining techniques to model the changes in visual style across time. Palermo et al. [8] dated historic color photographs using features which capture temporally discriminative information based on the evolution of color imaging processes over time.

Multi-task learning is a useful tool. Toshev et al. [4] use multi-task learning to simultaneously predict the positions of the various human body parts in an image and then combines the losses for back propagation. We refer the reader to [15] for a survey on multi-task learning in deep-learning. In our work, we used multi-task learning to jointly learn various combinations of styles, artists and dates such as style-artist, style-artist-date, etc. Our models trained to optimize losses for two or more categories generally saw a huge improvement in classification accuracy.

Gatys et al [5] worked on artistic style transfer by combining the content of one image with the painting style of another image. He defined a style reconstruction loss which was computed from the Gram matrices of the feature maps from all the convolutional layers. We use the same formulation, however our end-goal is not generating an image with similar style,

instead we use the Gram matrices from the feature maps for classification. A more recent work in this area by Johnson et al. [6] used a separate pre-trained VGG for computing the style loss to reduce the computational time.

Fine-grained classification tasks such as identifying the category of a bird or the kind of an airplane is challenging. Lin et al. [9] proposed a bilinear CNN model which takes the outer product of the two CNN models to form a bilinear vector which is then used for classification. While we did not use a bilinear CNN model in our work, we use the outer products of our features to consider pairwise interactions to aid in fine-grained classification. Our results show that classification using gram matrices produced results comparable with existing state-of-the-art, thus revealing new promising directions of research. Gao et al. [11] came up with compact bilinear pooling methods which efficiently compute outer product using less computational power while still achieving similar results.

CHAPTER 3: DATASET

The dataset collected by [3] is used in this thesis. Here, we briefly describe the dataset. The Wikiart paintings dataset was built from [16] and consists of 81,449 fine-art paintings from 1,119 artists ranging from fifteenth century to the present contemporary times. The paintings have 27 different styles (Abstract, Cubism, Impressionism, etc) and 45 different genres (Portrait, Landscape, etc). However, we do not include all the paintings for each task due to the limited number of samples for some of the classes. To put these into numbers, we used genres with more than 1,500 paintings which came to 10 out of the 45 genres for genre classification. For style classification, we used all the 27 styles. For artist classification, we used artists with more than 1,000 paintings. Only 23 artists met this criterion to form the classes for artist classification. Table 3.1 lists the set of style, genre and artist labels. Figures 3.1, 3.2 and 3.3 illustrate some examples of paintings for the different styles, artists and genres respectively.

For dating paintings, we used several different temporal groupings of the years ranging from 1-year intervals (fine-grained classification) to 100-year intervals (coarse classification). The dataset had nearly 65,000 paintings with date labels. More details on the dating division are included in Section 4.4. We also experimented with fine-grained artist classification where we relaxed our criterion to include artists with more than 1,000 paintings to artists with more than 100 paintings. This increased the number of classes for artists from 23 to 194 and made the classification really challenging. We call this new set of artists as expanded artists. The list of expanded artists is included in the supplementary materials table A.1.

Table 3.1: Table of the different classes of styles, artists and genres

Task type	Classes
Style	Early Renaissance; High Renaissance; Mannerism (Late Renaissance); Northern Renaissance; Baroque; Rococo; Romanticism; Realism; Impressionism; Post Impressionism; Pointilism; Symbolism; Fauvism; Expressionism; Cubism; Analytical-Cubism; synthetic cubism; Art Nouveau Modern; Abstract Expressionism; Color Field Painting; Action-Painting; New Realism; Naive Art Primitivism; Pop Art; Contemporary Realism; Minimalism; Ukiyo-e;
Genre	Abstract Painting; Cityscape; Genre Painting; Illustration; Landscape; Nude Painting (nu); Portrait; Religious Painting; Sketch and Study; Still Life
Artist	Albrecht Durer; Boris Kustodiev; Camille Pissarro; Childe Hassam; Claude Monet; Edgar Degas; Eugene Boudin; Gustave Dore; Ilya Repin; Ivan Aivazovsky; Ivan Shishkin; John Singer Sargent; Marc Chagall; Martiros Saryan; Nicholas Roerich; Pablo Picasso; Paul Cezanne; Pierre-Auguste Renoir; Pyotr Konchalovsky; Raphael Kirchner; Rembrandt; Salvador Dali; Vincent van Gogh;

Figure 3.1: Paintings from the styles of Impressionism, Abstract Expressionism and Pop-art (in order from left to right)

Figure 3.2: Paintings by Claude Monet, Pablo Picasso, Vincent Van Gogh and Camille Pissarro (in order from left to right)

Figure 3.3: Paintings from the genres of Portrait, Landscape and Still life (in order from left to right)

CHAPTER 4: FINE-TUNING VGG

The network used in our experiments is VGG [25] which performed very well in the ImageNet challenge in 2014. VGG has five convolutional layers (conv1-5), three max pooling layers (max1-3) and three fully connected layers (fc6-8). We used VGG-16, the architecture of which is shown in Figure 4.1.

It has already been shown that transfer tasks work quite well from CNNs pre-trained on natural images to paintings [12, 13]. We use VGG network pretrained on the ImageNet dataset [28] for our classification experiments with varying degrees of fine-tuning. For each of the 4 tasks – style classification, artist classification, genre classification and date classification, we fine-tune:

1. VGG-fc8 : Only the final fully-connected layer (fc8)
2. VGG-fcs : All the fully connected layers (fc6-8)
3. VGG-full : The entire VGG network

The last fully connected layer of VGG (fc8) has 1000 neurons corresponding to the 1000 classes in ImageNet. We change the number of neurons in the last layer to match the number of classes for that task. For example, style has 27 neurons in the last fully connected layer.

Since we change the number of neurons in the last fully connected layer, we also need to initialize the new weights associated with the layer. The weights were initialized from a zero-mean Gaussian distribution with a standard deviation of 0.01. The biases were initialized to 0.

The last layer (fc8) is followed by softmax activation. We minimize the cross entropy loss as our objective function. We use cross-entropy loss since it works quite well for multi-class classification problems. All our tasks are a form of multi-class classification. Cross-entropy loss

also works well with unbalanced datasets. While our dataset is not particularly skewed, it nevertheless helps with better training. For a given input (X), the model produces scores(S) for all the classes(C). The cross-entropy loss for a given target class (T) is calculated as follows:

$$loss(X, T) = -\log \frac{e^{S[T]}}{\sum_{j \in C} e^{S[j]}} \tag{4.1}$$

Saleh et al. [3], the creator of the Wikiart dataset was the first to work on style, artist and genre classification in paintings. In their work, they used features ranging from low-level to semantic level clubbed with various metric learning techniques to learn optimal similarity metrics. They used CNN features extracted from AlexNet [22] having 1,000 dimensions corresponding to the 1,000 categories in ImageNet as the semantic visual feature. Our baseline models discussed in this section beat their best performing visual feature and metric.

Tan et al. [2] performed several experiments for style, artist and genre classification on the Wikiart dataset [3]. They tried several variants using VGG namely no fine-tuning, using SVM at the end instead of softmax, fine-tuning the fully connected layers, making changes to the architecture by changing the number of neurons in the fc6 layer, etc. Our fine-tuning models perform better than theirs and our multi-task classification explained in the next section enhances the performances even further. We present the comparisons for each task in the below sections.

We used data augmentation techniques for all our experiments. In the training phase, each image went through image translation where a random square cropping of size 224*224 was extracted from the image of size 256*256. This is followed by random horizontal flips. The image is then normalized by subtracting the mean pixel value over the entire dataset. In each iteration, only one random square crop is chosen and randomly mirrored and the random crop is different across different iterations. In the validation phase, we extract a square crop of size

224*224 size from the center of the image and then subtract the mean pixel value. There is no random horizontal flipping in the validation phase.

All models are trained using stochastic gradient descent (SGD) with momentum and weight decay. The batch size depends on the current task and the size of the training/validation examples of the current task. The momentum value was fixed at 0.9 and the weight was halved every 10 epochs. The learning rate was chosen to be between 0.001 and 0.002 depending on the task. All the experiments were carried out using PyTorch on NVIDIA Tesla K-40.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB imag)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

The 6 different architecures of VGG Net. Configuration D produced the best results

Figure 4.1: The different VGG architectures proposed by Simonyan et al. [25]. We use VGG-16 (in the rectangle) for our experiments. Source: [25].

4.1 STYLE CLASSIFICATION

For the task of style classification, we have 27 style labels. So, the final fully connected layer had 27 neurons followed by a softmax. For this task, the best results were obtained using a learning rate of 0.002. Table 4.1 contains the results (accuracy percentage) of style classification for the different experiments. The first row corresponds to the accuracy percentage of Saleh et al. [3]. We chose their best performing visual feature with their best performing metric for style as a baseline. The second row corresponds to the accuracy percentage of Tan et al. [2]. Again, we chose their best performing experiment for style as our second baseline. The next three rows correspond to the three cases we discussed in Section 4. We found that the style class was a little skewed where Action Painting had only 100 images and Impressionism had over 13,000 images. So, we performed VGG-fcs again with weighted cross-entropy loss (VGG-fcs-weighted) where the weight for each class was calculated according to:

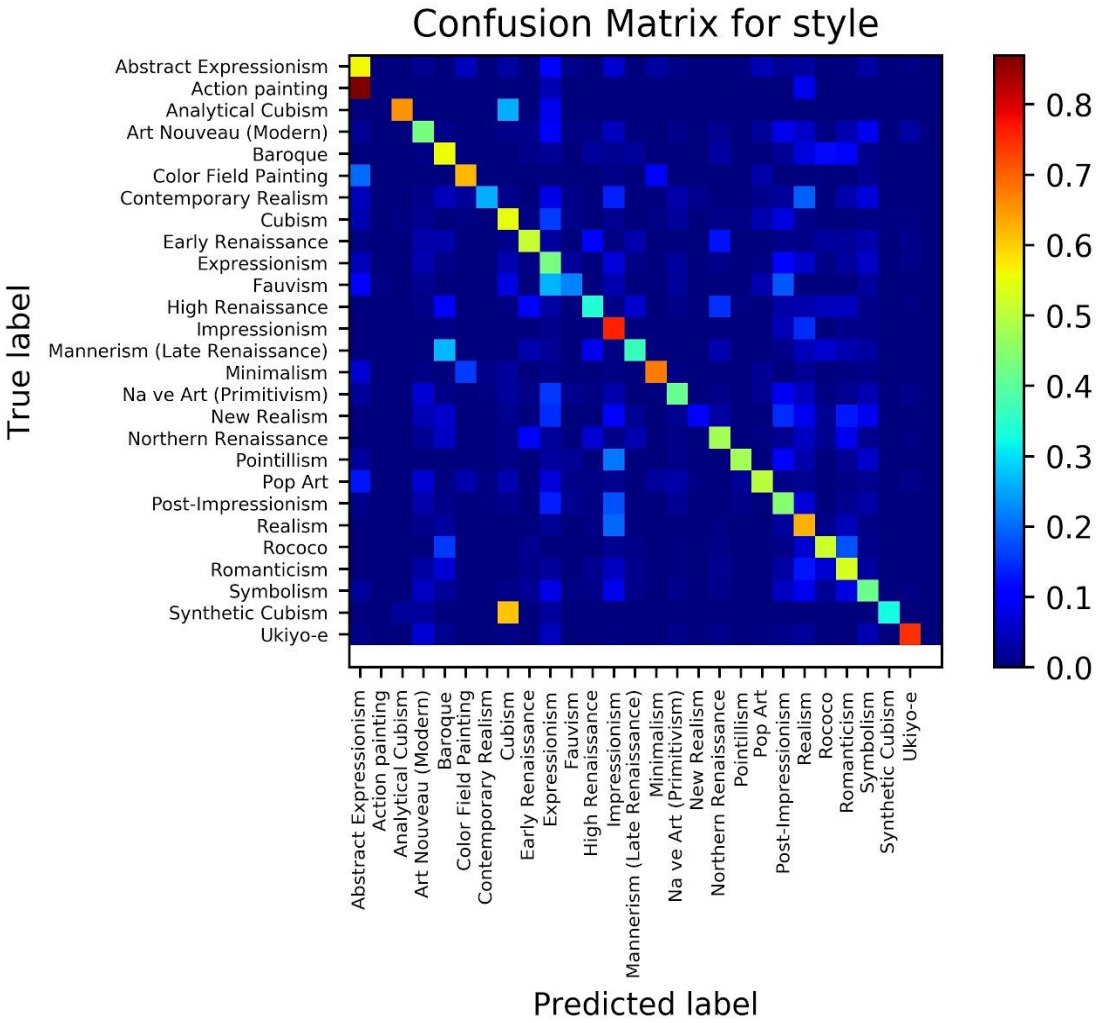
$$W(i) = \text{Total number of images}/(\text{Number of classes} * \text{Number of images in class } i) \quad (4.2)$$

Table 4.1: Accuracy percentage for style classification

Experiment	**Accuracy**
Saleh et al. [3]	45.97
Tan et al. [2]	54.50
VGG-fc8	50.04
VGG-fcs	52.24
VGG-fcs weighted	54.88
VGG-full	65.70

The results show that transfer learning in CNN helps improve the accuracy and the deeper we fine-tune the better the results are. The full-model fine-tuning performed the best and it outperformed the state-of-the-art style classifier by **20.5**%. Our second best model which is the weighted fine-tuning also beats the state-of-the-art style classifier by a small margin. Figure 4.2 shows a visualization of the confusion matrix for VGG-full for style classification. In the matrix red represents higher values. In an ideal classification, only the diagonals should be red.

Figure 4.2: Confusion matrix for style classification using VGG-full



Upon analyzing the matrix, we found that Ukiyo-e style is quite distinctive. This is because Ukiyo-e is a Japanese art form and is very different from all the other styles in the dataset. There is a lot of confusion between Abstract Expressionism and Action Painting. This is expected since Action Painting is a subgenre of Abstract Expressionism. Cubism is confused between Cubism, Analytic cubism and Synthetic Cubism which is also expected since Synthetic and Analytic Cubism are essentially a subgenre of Cubism. Synthetic cubism has more continued usage of collage and pasted papers, but less linear perspective than cubism. There is some confusion between Expressionism and Fauvism which is expected based on art history literature.

4.2 GENRE CLASSIFICATION

For the task of genre classification, we have 10 genre labels. Each genre class has more than 1,500 paintings in it. The final fully connected layer has 10 neurons followed by a softmax. For this task, the best results were obtained using a learning rate of 0.002. Table 4.2 contains the results (accuracy percentage) of genre classification for the different experiments. The first row corresponds to the accuracy percentage of Saleh et al. [3]. We chose their best performing visual feature with their best performing metric for genre as a baseline. The second row corresponds to the accuracy percentage of Tan et al. [2]. Again, we chose their best performing experiment for genre as our second baseline. The next three rows correspond to the three cases we discussed in Section 4.

The results again show that transfer learning in CNN helps improve the accuracy and the deeper we fine-tune the better the results are. Our vgg-full performs the best and beats the existing state-of-the-art genre classifier by **5.7%.**

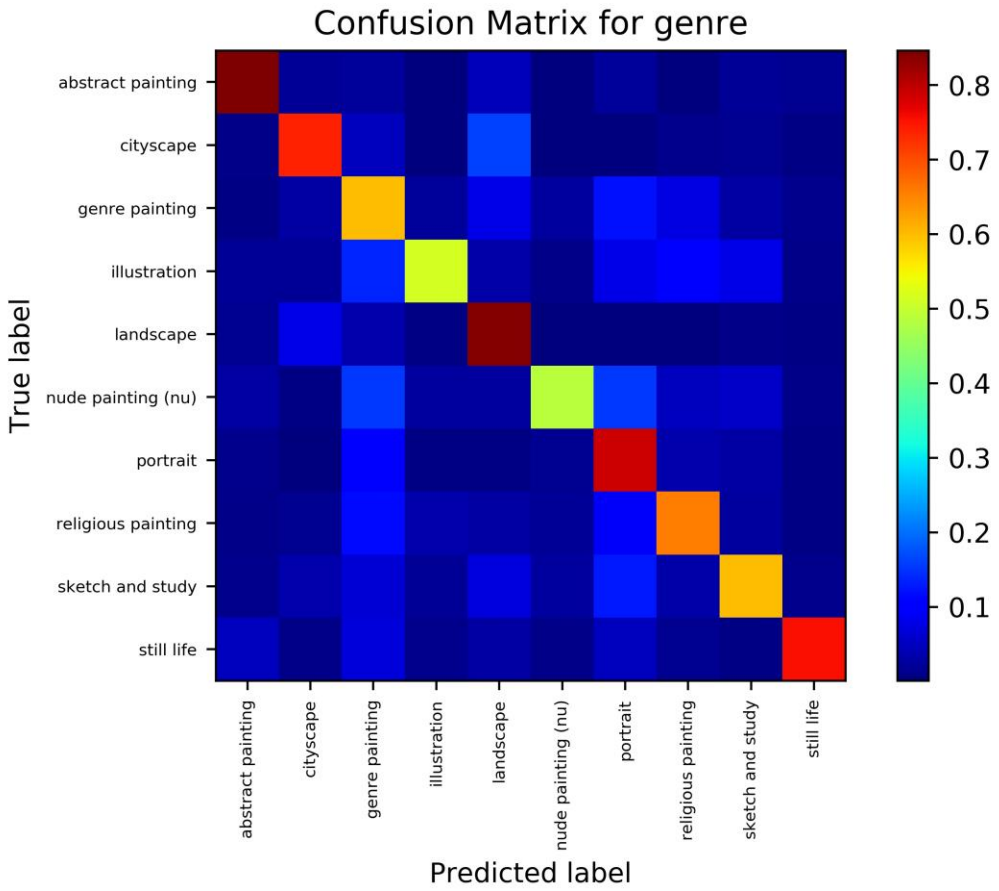
Table 4.2: Accuracy percentage for genre classification

Experiment	**Accuracy**
Saleh et al. [3]	60.28
Tan et al. [2]	74.14
VGG-fc8	70.28
VGG-fcs	72.74
VGG-full	78.37

Figure 4.3 shows a visualization of the confusion matrix for VGG-full for the task of genre classification. In the matrix red represents higher values. In an ideal classification, only the diagonals should be red.

We observe that the Landscape and Portrait are quite distinctive. We attribute this to the fact that the pre-trained CNN works quite well for face detection and scene recognition. There is confusion between Landscape, Cityscape and Genre Painting. This is because there lots of common elements between these three genres. Landscape has rivers, mountains and valleys (no significant figures) which is similar to what Genre Painting has. Genre Painting captures daily life. Cityscape on the other hand uses lots of open space like Landscape. Abstract painting is distinctive because its genre is really quite different from the other genre types which are mostly figurative. Nude painting has confusion with several classes since it has common elements in the different classes to varied degrees.

Figure 4.3: Confusion matrix for genre classification using VGG-full



4.3 ARTIST CLASSIFICATION

For the task of artist classification, we have 23 artist labels each of which has more than 1,000 paintings in it. The final fully connected layer has 23 neurons followed by a softmax. For this task, the best results were obtained using a learning rate of 0.002. Table 4.3 contains the results (accuracy percentage) of artist classification for the different experiments. The first row corresponds to the accuracy percentage of Saleh et al. [3]. We chose their best performing visual feature with their best performing metric for artists as a baseline. The second row corresponds to the accuracy percentage of Tan et al. [2]. Again, we chose their best performing experiment for

artists as our second baseline. The next three rows correspond to the three cases we discussed in Section 4.

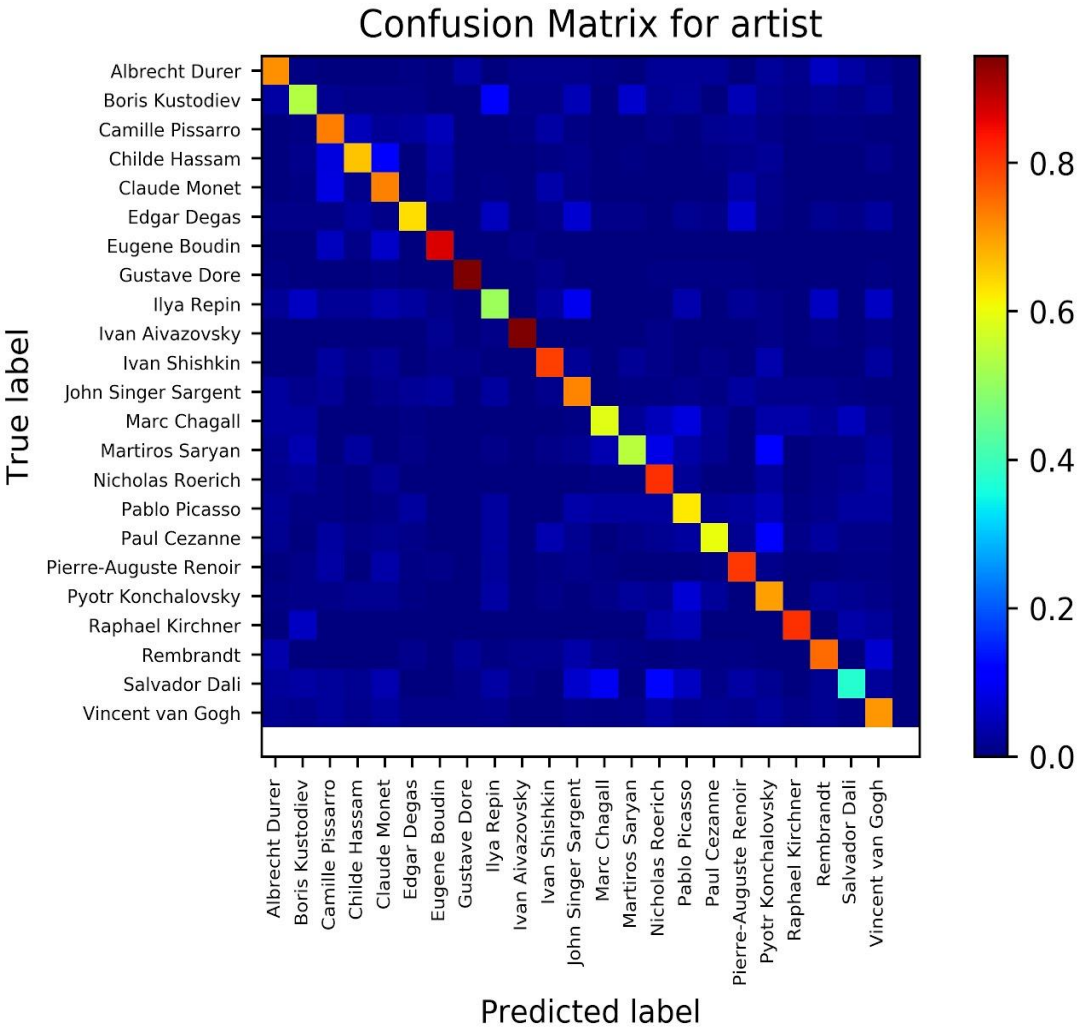
Out of the three tasks of style, artist and genre classification, artist classification seemed to have the highest accuracy. This could be because artists are inherently more discriminative than the other two tasks. The results once again confirm that the deeper we fine-tune the better the accuracy. Our vgg-full performs the best and is better than the existing state-of-the-art artist classifier by **12.74%**. In our later experiments in Section 5, we further outperform our VGG-full baseline.

Table 4.3: Accuracy percentage for artist classification

Experiment	**Accuracy**
Saleh et al. [3]	63.06
Tan et al. [2]	76.11
VGG-fc8	69.14
VGG-fcs	72.22
VGG-full	82.77

Figure 4.4 shows a visualization of the confusion matrix for VGG-full for the task of artist classification. In the matrix red represents higher values. In an ideal classification, only the diagonals should be red.

Figure 4.4: Confusion matrix for artist classification using VGG-full



The artists that have less confusion generally prefer certain techniques or objects in their paintings. From the confusion matrix, we see that Gustave Dore is quite discriminative. Upon inspection we found that he uses engravings, etchings, and lithography which results in greyish paintings. Eugene Boudin on the other hand is also well classified by our model. He mainly indulges in outdoor scenes, and most of his paintings were of marine and seashore. Ivan Shishkin

is marked as discriminative by our model. Shishkin was one of the most popular landscape painters of Russia. His distinctive style involved wooded landscapes.

Salvator Dali was a prominent Spanish artist, however the model failed to classify him. Infact, it is know that Salvator Dali and Pablo Picasso influenced each other's work and hence it is logical to have some confusion between them. There is confusion between Claude Monet and Childe Hassam. Hassam is an American Impressionist who declared himself to be influenced by French Impressionists and painted works similar to Monet.

Claude Monet also seems to be confused with Camille Pissaro by our model. Both of them were Impressionist artists who lived in the late nineteenth century. Art history says they were childhood friends which led to a lot of noticeable interactions between the two and hence the confusion.

Our model also shows confusion between Boris Kustodiev and Ilya Repin. Upon inspection we found that both these realist artists lived in the late nineteenth century and infact shared a master-pupil bond [36] with Kustodiev being the pupil and Repin his master.

4.4 FINE-GRAINED ARTIST CLASSIFICATION

In this thesis, one of our major goals was to experiment with really fine-grained classification. None of the prior works had any fine-grained classification and we believe this is the first fine-grained classifier for the domain of paintings. We increased the number of artists from 23 to 194. The process was similar to the normal artist classification. We selected all artists who had over 100 images in the Wikiart dataset. This amounted to 194 artists which form our expanded artists set. The names of all the 194 artists are included in the Supplementary section in table A.1. Deep learning usually requires a good number of training examples per class to work well but we lowered the threshold to 100 paintings to see how far we can go with our results.

For the task of fine-grained artist classification, we have 194 artist labels each of which has more than 100 paintings in it. The final fully connected layer has 194 neurons followed by a softmax. For this task, the best results were obtained using a learning rate of 0.002. Table 4.4 contains the results (accuracy percentage) of expanded artist classification for the different experiments. The first row corresponds to the baseline accuracy percentage obtained by randomly classifying images into the 194 classes.

Table 4.4: Accuracy percentage for expanded artist classification

Experiment	**Accuracy**
Random	0.51
VGG-fc8	37.23
VGG-fcs	52.10
VGG-full	65.42

We observe that the deeper we fine-tune the better the accuracy as with all the previous tasks. Our vgg-full performs the best. We perform more extensive experiments with the expanded artists set in Section 5.

4.5 DATE CLASSIFICATION

Dating of the content in the images has been attempted before by Ginosar et al. [1] where she dated portraits of female students from school yearbooks. The yearbooks spanned a period of 83 years. They formulated the problem of dating portraits as an 83-way classification problem and fine-tuned the last layer of the VGG network (fc8) with the new set of 83 classes. Their accuracies were almost double the existing state-of-the-art accuracy. For evaluation, they used

both the classification accuracy and L1 distance between the predicted year and the actual year as metrics. We only use the classification accuracy as a metric in this work. Another work in this area was done by Vittayakorn et al. [29] where they dated photographs of vintage cars and Flickr clothing dataset. They formulation was similar to [1], however their temporal resolution was a decade instead of a year. They used L1 distance as their evaluation metric. Lee et al. [10] used visual data mining techniques to model the changes in visual style across time.

We formulate our problem similar to the setting in [1] where the problem of dating paintings is transformed into the problem of an n-way classification. To the best of our knowledge, this is the first work that does dating within the domain of painting. The ability to estimate when a painting was made would be very useful for categorization of paintings by museums and to help art historians analyze paintings.

Like with artists, we also want to explore really fine-grained classification with dates. To this end, we conducted several experiments with different temporal resolution of our classes. We performed 6 experiments with different temporal resolutions ranging from 1 year to 100 years. The range of dates in our dataset was between the year 1400 to 2012. Table 4.5 shows the number of classes for each temporal resolution obtained after removing all classes which had less than 100 images.

Table 4.5: Number of classes for each temporal resolution

Temporal resolution	**Number of classes**
1 year	145
5 year	87
10 year	54

Table 4.5 (cont.)

Temporal resolution	**Number of classes**
20 year	28
50 year	13
100 year	7

For all the 6 experiments, the best results were obtained using a learning rate of 0.002. Table 4.6 contains the results (accuracy percentage) of date classification for the different experiments. The rows correspond to the different levels of fine-tuning as seen in the previous tasks of style, artist and genre classification. The columns correspond to the different temporal resolutions for this task. We used 6 different temporal bins – 1 year, 5 year, 10 year, 20 year, 50 year and 100 year intervals.

Table 4.6: Accuracy percentage for date classification

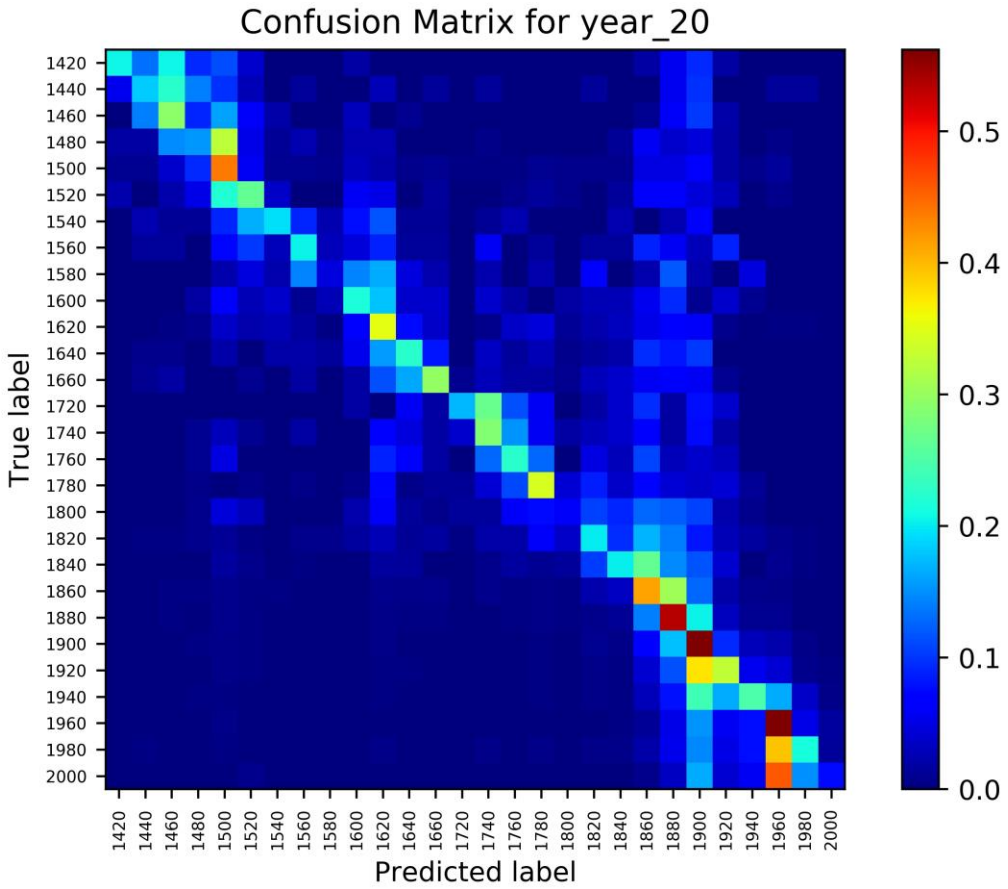
Experiment	**1-year**	**5-year**	**10-year**	**20-year**	**50-year**	**100-year**
VGG-fc8	3.42	9.32	15.83	27.57	42.91	55.37
VGG-fcs	9.37	19.25	28.68	41.85	62.40	72.09
VGG-full	4.51	11.37	19.28	32.29	50.09	62.67

From the above results, we see that with decrease in temporal resolution, the accuracy starts increasing which is expected and intuitive since a 1-year resolution classification is a naturally more difficult task than a 100-year resolution classification. An interesting point to note here is that fine-tuning the entire VGG model leads to decrease in performance contrary to what we have been seeing so far. By fine-tuning the entire model we might be over-fitting the data for

the task of date-classification which explains the decay in performance. Further experiments are required to validate this claim. Our vgg-fcs performs the best for all 6 experiments. In Section 5, we use multi-task training to significantly improve on our scores.

Figure 4.5 shows a visualization of the confusion matrix for VGG-fcs for the task of date classification for a temporal resolution of 20. In the matrix red represents higher values. In an ideal classification, only the diagonals should be red. The confusion matrices for temporal resolution of 10 and 100 are presented in Figure A.1 and Figure A.2 respectively in the supplementary section for the interested reader.

Figure 4.5: Confusion matrix for date classification (temporal resolution of 20) using VGG-fcs



Upon analyzing the figure, we can see that there is a lot of confusion between an interval and its neighbors on both sides. This is expected since date prediction is a difficult task and it is easy to mix the current interval with the neighboring intervals. Some of the nineteenth century intervals are however discriminative such as 1880, 1900, 1960. Many of the intervals have been confused with the intervals in the late 1900s. This is perhaps due to the bias in the dataset which has considerably more number of paintings from the twentieth century than the other years.

Figure 4.6: Confusion matrix for date classification (temporal resolution of 50) using VGG-fcs

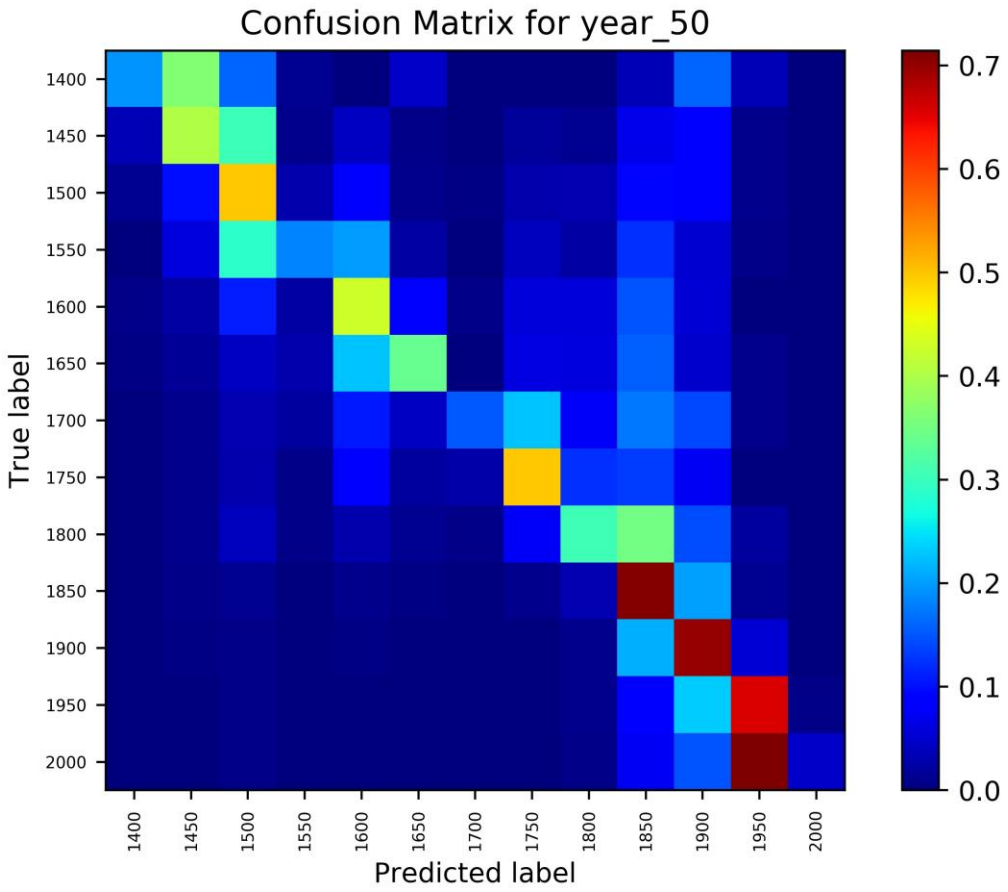


Figure 4.6 shows a visualization of the confusion matrix for VGG-fcs for the task of date classification for a temporal resolution of 50. We can see from the figure that the confusion is

less due to wider intervals as compared to the Figure 4.5. The confusion between neighboring years can also be seen in this figure. However, from this figure we observe that the late nineteenth century, twentieth century paintings are quite discriminative which could be due to combining the different intervals into a big interval thereby making the task easier.

4.6 ART PERIOD CLASSIFICATION

The oldest known painting is at the Grotte Chauvet in France [37], which is believed to be 32,000 years old. We have come a long way since then. The entire history of paintings can be divided into art periods where each such period is a phase in the development of the work of an artist, groups of artists or art movement. The paintings in the Wikiart dataset range from 1400 to 2012. The art periods that existed between this time and their rough dates are presented in Table 4.7. The table also lists the styles that each time period contains from the Wikiart dataset. The Japanese art even though it has overlapping time periods with the Modern Art is listed separately since its origin is from Japan and is very different from all the Modern Art styles.

Table 4.7: Art periods and their dates for the paintings in the Wikiart dataset

Art period	**Dates**	**Styles included**
Early Renaissance	1400-1600	Early Renaissance; High Renaissance; Mannerism (late renaissance); Northern Renaissance
Post Renaissance	1600-1800	Baroque; Rococo;
Beginning of Modern Art	1800-1860	Romanticism; Realism; Impressionism; Post Impressionism; Pointilism;
Modern Art	1860-1940	Symbolism; Fauvism; Expressionism; Cubism; Analytical-Cubism; synthetic cubism; Art Nouveau Modern;
Late Modern Art	1940-1970	Abstract Expressionism; Color Field Painting; Action-Painting; New Realism; Naive Art Primitivism;

Table 4.7 (cont.)

Art period	**Dates**	**Styles included**
Contemporary Art	1970-now	Pop Art; Contemporary Realism; Minimalism;
Japanese Art	1600-1860	Ukiyo-e;

We perform the task of art-period classification using the art periods above. We have 7 art period labels and we perform the same experiments as done for the other tasks discussed previously. For this task, the best results were obtained using a learning rate of 0.002. Table 4.8 contains the results (accuracy percentage) of art period classification for the different experiments. Our vgg-full performs the best. Figure 4.7 shows a visualization of the confusion matrix for VGG-full for the task of art period classification. In the matrix red represents higher values. In an ideal classification, only the diagonals should be red.

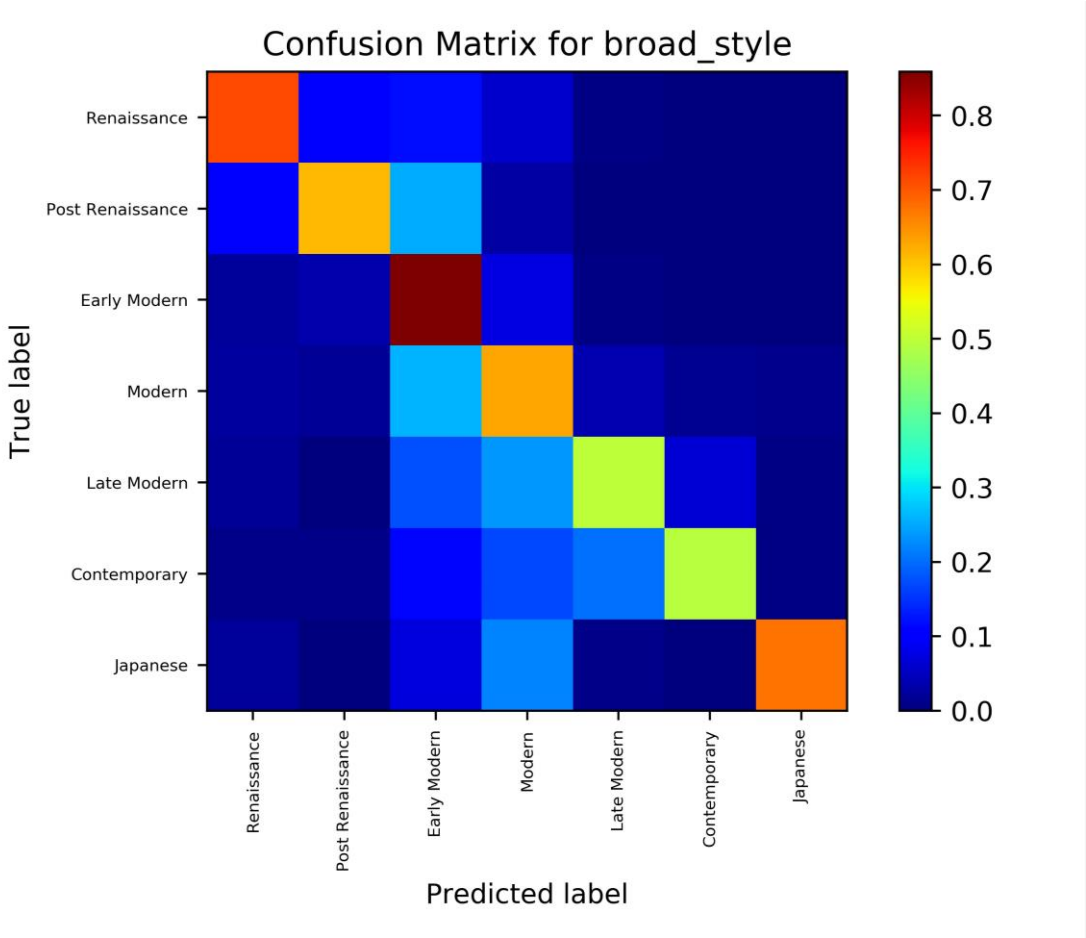
Table 4.8: Accuracy percentage for art period classification

Experiment	**Accuracy**
VGG-fc8	68.35
VGG-fcs	73.02
VGG-full	79.68

The figure shows that our model can recognize the different art periods well enough. The Japanese art period is well recognized which we attribute to the fact that the Japanese style Ukiyo-e is quite distinctive from all the other styles in our dataset. Early Modern is also a distinctive period from the figure. There is confusion between the transitioning of one art period to another which is expected since the end of an art period and the beginning of another is

ambiguous and cannot be given a hard year. For example, there is confusion between Contemporary and Late Modern, Modern and Late Modern and so on.

Figure 4.7: Confusion matrix for art period classification



CHAPTER 5: MULTI-TASK CLASSIFICATION

In the last section, our models beat the state-of-the-art classifiers by simply fine-tuning for a single task. However, we ignored information that might help us do better. This information comes from the training signals of related tasks. We can enable our models to do generalize better on our original task by sharing representations between related tasks. This is multi-task learning. Multi-task learning can be seen as a form of inductive transfer. Inductive transfer helps improve a model's performance [15] by introducing an inductive bias, which causes a model to prefer some hypotheses over others.

In deep learning, multi-task learning is typically done with either hard or soft parameter sharing of hidden layers. In this thesis, we used hard parameter sharing for our multi-task learning. Hard parameter sharing is generally applied by sharing the hidden layers between all tasks, while keeping task-specific output layers depending on the number of tasks. This greatly reduces the risk of over-fitting. The more tasks we learn simultaneously with task-specific output layers, the more our model has to find a representation that captures all of the tasks and overfits less on the original task.

Multi-task learning works because of a number of reasons. It does implicit data augmentation by increasing the sample size that we are using for our model. Some features which are difficult to learn for task A might be easy to learn for task B. It biases the model to prefer representations that both tasks prefer. It also acts as a regularizer by introducing an inductive bias. If one of the tasks is noisy or the data is limited, multi-task learning can help the model to focus its attention on those features that actually matter as other tasks will provide additional evidence for the relevance or irrelevance of those features.

Figure 5.1 shows our hard parameter sharing multi-task architecture for VGG. The diagram depicts two classification heads for VGG for learning two tasks simultaneously. Both tasks share the convolutional layers (conv1-5), and have their own classification layers (fc6-8). The model is trained end-to-end for both tasks. We take VGG[25] pretrained on the ImageNet dataset [28] and fine-tune the entire network for this multi-task scenario. The total loss of the model is computed as the sum of losses of all the classification heads computed independently using the shared layers and then back-propagated appropriately to train the model.

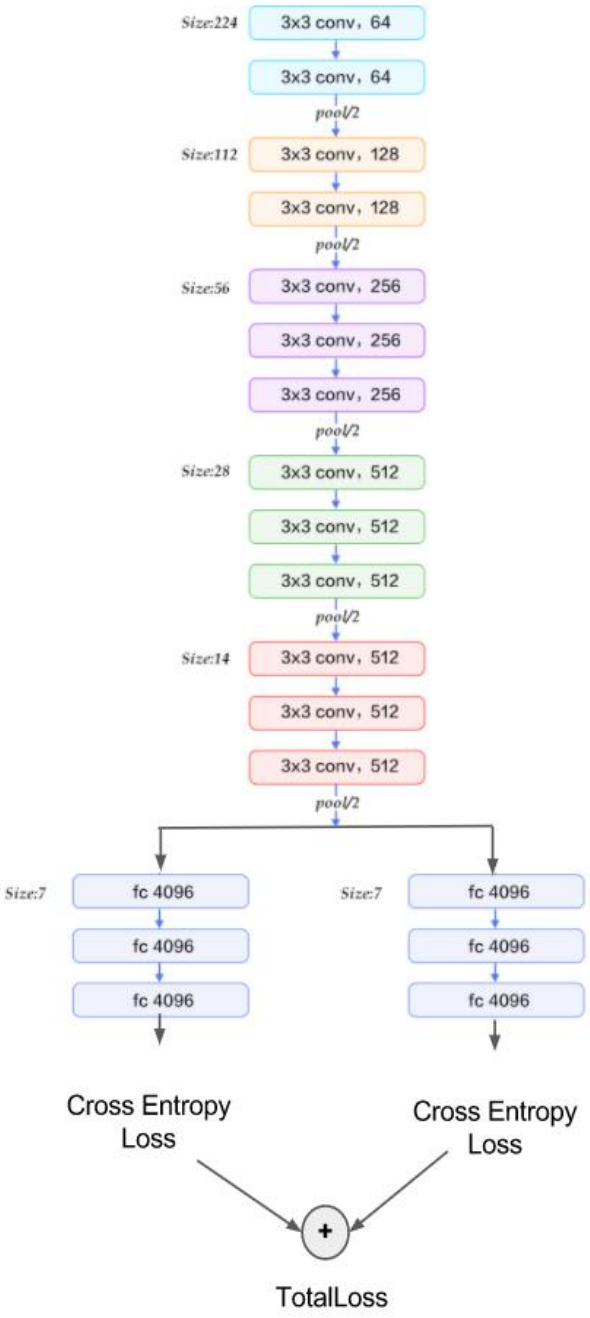
We used multi-task learning to jointly learn styles and artists, and styles, artists and dates. Our models trained to optimize losses for two or more categories generally saw an improvement in classification accuracy. Like in Section 4, the last fully connected layer (fc8) of the tasks was initialized with the respective number of class outputs for that task and their weights were initialized from a zero-mean Gaussian distribution with a standard deviation of 0.01. The biases were initialized to 0. The data augmentation techniques used in Section 4 was also used for multi-task classifications.

In the last section, our models outperformed both Saleh et al. [3] and Tan et al. [2] on the tasks of style, artist and genre classification. In this section, we show that our multi-task learning models outperform all our previous fine-tuning models. We also show improvements in accuracy on the tasks of fine-grained artist classification and date classification compared to our baseline model.

All models are trained using stochastic gradient descent (SGD) with momentum and weight decay. The batch size is smaller than what is was for learning a single task. The momentum value was fixed at 0.9 and the weight was halved every 10 epochs. The best results

were obtained by using a learning rate of 0.001. All the experiments were carried out using PyTorch on NVIDIA Tesla K-40.

Figure 5.1: VGG network modified to support multi-task learning. The convolutional layers are shared by the different tasks. Each task has its own classification head.



5.1 ARTIST CLASSIFICATION REVISITED

We classify artists and styles of paintings together using the multi-task model described above. In section 4.3, 23 artists who had more than 1,000 paintings in the dataset were chosen to form the classes for artist classification. We use the same set of paintings of the above artists for this experiment. Thus, each painting now has an artist who made it and a style to which it belongs. These form the two tasks which we want to learn simultaneously for each painting.

Table 5.1 contains the results (accuracy percentage) of artist classification for the different experiments. The first two rows correspond to the accuracy percentage of Saleh et al. [3] and Tan et al. [2] for artist classification. The third row corresponds to our best baseline model obtained by fine-tuning VGG for artist classification and the last row correspond to the accuracy obtained by multi-task learning which we call VGG-multitask.

Table 5.1: Accuracy percentage for artist classification using multi-task classification

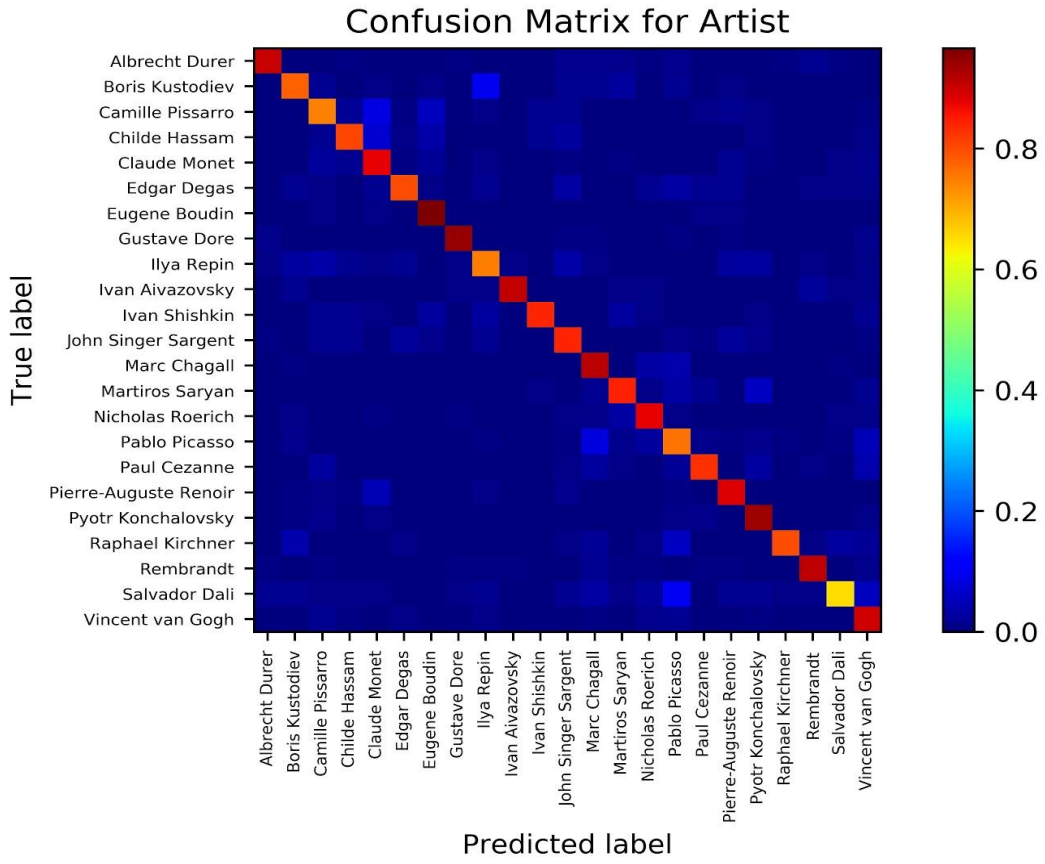
Experiment	**Accuracy**
Saleh et al. [3]	63.06
Tan et al. [2]	76.11
VGG-finetune	82.77
VGG-multitask	85.64

Our multi-task model shows an **3.46**% improvement over our previous fine-tuning model which was already better than the previous state-of-the-art model for artist classification. Thus, multi-task learning greatly helped in boosting the performance of the model. Our model is now able to capture those features which are more relevant for classifying artists than our previous baseline model. The style classification accuracy was 83.39%, however the paintings included in

this task only covered 16 of the 27 styles in our dataset and is hence, not directly comparable to the other style classification results.

Figure 5.2 shows a visualization of the confusion matrix for VGG-multitask for the task of artist classification. In the matrix red represents higher values. In an ideal classification, only the diagonals should be red.

Figure 5.2: Confusion matrix for artist classification using VGG-multitask



This confusion matrix is much more close to ideal than the one in Figure 4.4. The artists are more discriminative. The analysis done in Section 4.3 still holds. Even though Salvator Dali is better classified by our mode, he is still confused with Pablo Picasso who is believed to have

influenced his work. There is confusion between Claude Monet and Childe Hassam, and between Claude Monet and Camille Pissaro (they were childhood friends). Boris Kustodiev and Ilya Repin are also confused. Refer to earlier discussion in Section 4.3.

5.2 FINE-GRAINED ARTIST CLASSIFICATION REVISITED

We classify expanded artists and styles of paintings together using the multi-task model. Fine-grained classification of artists was introduced in section 4.4, where we increased the number of artists from 23 to 194 by including all artists who had more than 100 paintings in the dataset. This significantly increases the complexity of the problem because of two reasons – Less samples per class (as low as 100) and large number of classes leading to more confusion in classifying the artists. We use the paintings of these 194 artists for this experiment. Each such painting has an artist label and a style label. We learn both these tasks jointly with each task having its own classification head.

Table 5.2 contains the results (accuracy percentage) of expanded artist classification for the different experiments respectively. The first row corresponds to our best baseline model obtained by fine-tuning VGG for the same task and the last row correspond to the accuracy obtained by multi-task learning for expanded artists and styles.

Table 5.2: Accuracy percentage for expanded artist classification using multi-task classification

Experiment	**Accuracy**
VGG-finetune	65.42
VGG-multitask	68.80

Table 5.3 contains the results (accuracy percentage) of style classification for the different experiments respectively. The first two rows correspond to the accuracy percentage of Saleh et al. [3] and Tan et al. [2] for style classification. The third row corresponds to our best baseline model obtained by fine-tuning VGG for style classification and the last row correspond to the accuracy obtained by multi-task learning.

Table 5.3: Accuracy percentage for style classification using multi-task classification

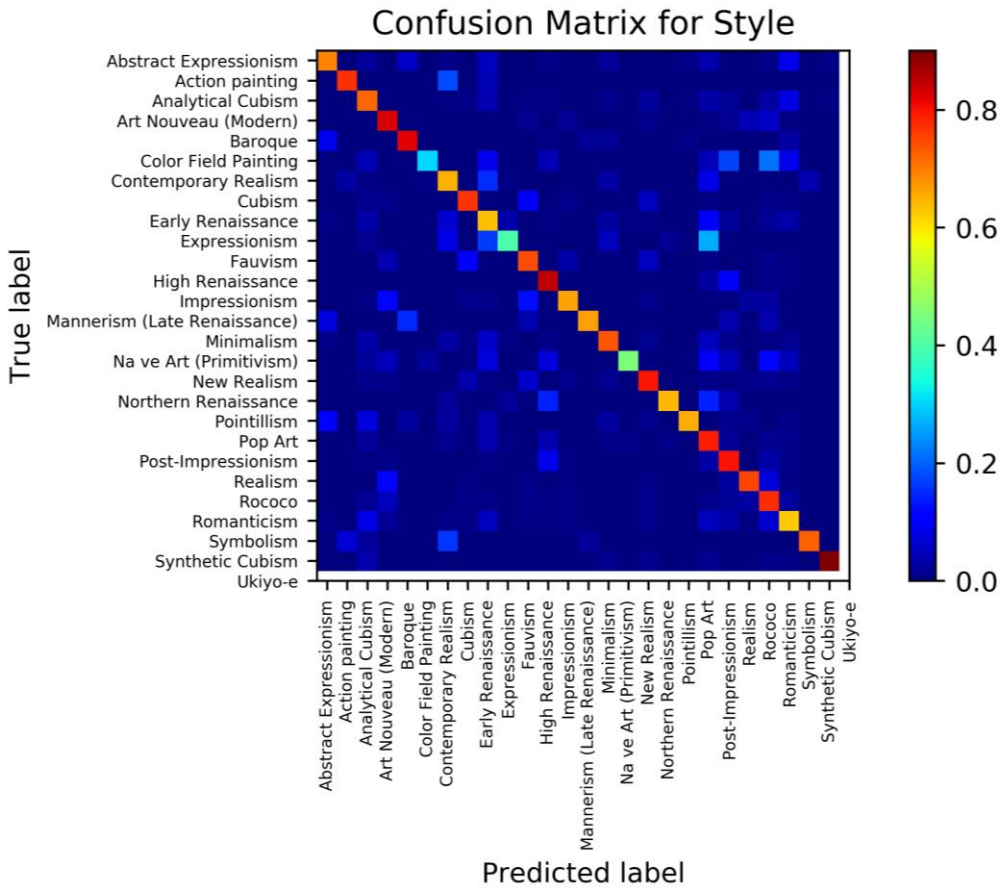
Experiment	**Accuracy**
Saleh et al. [3]	45.97
Tan et al. [2]	54.50
VGG-finetune	65.70
VGG-multitask	76.35

Our multi-task model shows a **5.16**% improvement over our previous baseline model for expanded artist classification and a **16.21**% improvement over our previous baseline model and **28.61%** improvement over the previous state-of-the-art model for style classification. Thus, multi-task learning greatly helped in boosting the performance of the model. Our model not only does well for style classification but also really fine-grained classification in the paintings domain. Figure 5.3 shows a visualization of the confusion matrix for VGG-multitask for the task of style classification.

The confusion matrix shows that the multi-task model learnt the styles better than the previous matrix from Figure 4.2. The model was able to do away with a lot of the confusions seen in Section 4.1. Abstract Expressionism and Action Painting are not confused with each other. All three forms of Cubism (Cubism, Synthetic Cubism, and Analytical Cubism) are also

not confused with each other implying that the model is able to discriminate between sub-genres of styles.

Figure 5.3: Confusion matrix for style classification using VGG-multitask



5.3 DATE CLASSIFICATION REVISITED

The aim of this section is to answer how multi-task learning helps in date classification and how far could we go with our model's performance by increasing the number of tasks learnt simultaneously. We perform two different multi-task experiments as described below:

- *Two-task learning:* We use multi-task learning for learning date and styles together. With this experiment, we would like to see how multi-task learning impacts date classification and style classification. We learnt style and expanded artists before. This experiment would also answer if learning expanded artists with style leads to better models than learning dates with styles.
- *Three-task learning:* We use multi-task learning for learning 3 tasks simultaneously, i.e. date, style and expanded artists. We have previously learnt expanded artists and style together, and date and style together. With this experiment, we would like to see how learning all three together impacts the individual tasks.

We use the same 6 temporal resolutions used in Section 4.5. For a given resolution, the classes were obtained in the same manner by taking all intervals which had more than 100 paintings. From this set of paintings, we learn the date of the painting, the style of the painting and the artist who made it (for three-task learning) together by training the classification heads for each task independently.

Table 5.4 contains the results (accuracy percentage) of date and style classification for two-task learning. The first row corresponds to the baseline model for the different temporal resolutions obtained by fine-tuning a VGG for a date classification only (VGG-finetune date). The second row corresponds to the model scores for the different temporal resolutions obtained by jointly training style and date (VGG-twotask date). The third row corresponds to the model scores for style classification obtained jointly training style and date (VGG-twotask style). The columns correspond to the different temporal resolutions for date classification. We used 6 different temporal bins – 1 year, 5 year, 10 year, 20 year, 50 year and 100 year intervals.

Table 5.4: Accuracies for style and date classification using two-task learning

Experiment	**1-year**	**5-year**	**10-year**	**20-year**	**50-year**	**100-year**
VGG-finetune date	9.37	19.25	28.68	41.85	62.40	72.09
VGG-twotask date	13.77	26.04	37.97	51.25	71.38	81.14
VGG-twotask style	67.54	67.13	67.33	66.36	65.46	66.99

The date classification scores jumps up considerably from the baseline fine-tuning model for all the temporal resolutions. Style classification scores improve by approximately **2.8**% from the previous scores in Table 4.1. However, we observed that there is no considerable difference in style classification scores for the different temporal resolutions implying that the temporal resolution does not play much of a role in style classification. Compared to Table 5.3 where we trained style and expanded artist together, the style classification improvement is less with date than with expanded artists. The best style score with date is 67.54% while with expanded artists it is 76.35%. This implies that training style with expanded artists makes the model learn more discriminative features for style than it does with date.

Table 5.5 contains the results (accuracy percentage) of date, expanded artist and style classification for three-task-learning. The first row corresponds to the baseline model for the different temporal resolutions obtained by fine-tuning a VGG for date classification only (VGG-finetune date). The second row corresponds to the model scores for the different temporal resolutions for triple task learning (VGG-triple date). The third row corresponds to the model scores for style classification for triple task learning (VGG-triple style). The fourth row corresponds to the model scores for expanded artist classification for triple task learning (VGG-triple expanded artist). The columns correspond to the 6 different temporal resolutions for date classification, style classification and expanded artist classification.

Table 5.5: Accuracies for expanded artist, style and date classification using three-task learning

Experiment	**1-year**	**5-year**	**10-year**	**20-year**	**50-year**	**100-year**
VGG-finetune date	9.37	19.25	28.68	41.85	62.40	72.09
VGG-triple date	19.12	33.15	45.34	59.33	77.80	83.21
VGG-triple style	75.61	75.11	74.08	74.36	74.67	75.20
VGG-triple expanded artist	67.95	66.09	65.94	66.23	66.56	66.19

The above results show considerable increase in accuracies for date classification for each temporal resolution compared to both the simple fine-tuned VGG and the multi-task model trained to jointly learn date and style in Table 5.4. This justifies our hypothesis that multi-task learning is beneficial for fine-grained date classification and learning all three tasks together helps boost the accuracy for date classification. The style accuracy scores with two-task learning is lower than three-task learning which implies that training with expanded artists makes the model more discriminative for style features than with date alone.

The style and expanded artist accuracy when trained together were 76.35 and 68.60 respectively from Table 5.2 and 5.3. Here, we observe that they do not improve by training them together with date and remain consistent even across the different temporal resolutions. We can conclude that the features that would make style or expanded artists classification better were already learnt when training the two together and there is no new dimension that dates add to it and hence the constant accuracies.

Figure 5.4 shows the classification results of our model for three-task learning (style, expanded artist and date with a temporal resolution of 20) on some famous paintings. In each of the example, the true labels and the predicted labels are shown. Our model confuses Leonardo da Vinci with Titian. Both these famous artists were Renaissance rivals and their works could have

influenced each other. Date intervals are also likely to be confused with neighboring intervals as discussed previously.

Figure 5.4: Classification results of our model for three-task learning namely expanded artist, style and date with a temporal resolution of 20 on some famous paintings. The name of the painting is included beside the respective paintings.

Mona Lisa

True Labels	Predicted Labels
Style : High Renaissance	Style : High Renaissance
Artist : Leonardo da Vinci	Artist : Titian
Year : 1880-1900	Year : 1880-1900

Starry Night

True Labels	Predicted Labels
Style : Post Impressionism	Style : Post Impressionism
Artist : Vincent Van Gogh	Artist : Vincent Van Gogh
Year : 1500-1520	Year : 1500-1520

Figure 5.4 (cont.)

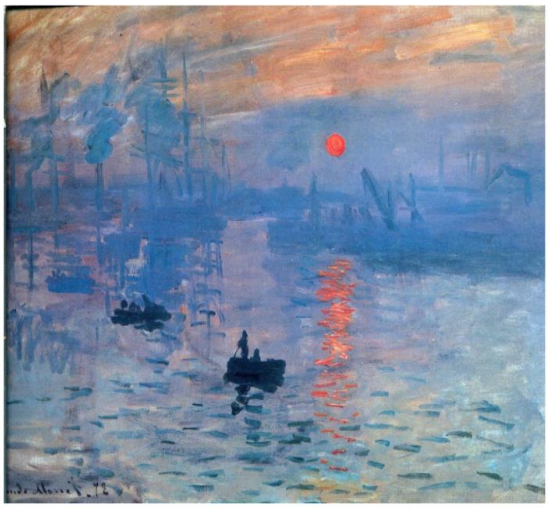
Maid of Athens

True Labels	Predicted Labels
Style : Art Nouveau (Modern)	Style : Art Nouveau (Modern)
Artist : Raphael Kirchner	Artist : Raphael Kirchner
Year : 1900-1920	Year : 1880-1900

Umbrellas

True Labels	Predicted Labels
Style : Impressionism	Style : Impressionism
Artist : Pierre-Auguste Renoir	Artist : Pierre-Auguste Renoir
Year : 1880-1900	Year : 1880-1900

Figure 5.4 (cont.)



Impressionism Sunrise

True Labels	Predicted Labels
Style : Impressionism	Style : Impressionism
Artist : Claude Monet	Artist : Claude Monet
Year : 1860-1880	Year : 1860-1880

The Old Blind Guitarist

True Labels	Predicted Labels
Style : Expressionism	Style : Expressionism
Artist : Pablo Picasso	Artist : Pablo Picasso
Year : 1900-1920	Year : 1900-1920

CHAPTER 6: NEW ARCHITECTURE FOR PAINTING CLASSIFICATION

There has been work (Gatys et al [5], Johnson et al. [6]) on transferring styles from one painting to another. While our work does not involve transferring styles, it does work with the domain of paintings and being able to extract features that capture the style and/or content of a painting would prove to be useful for the task of classification in paintings.

We take inspiration from their works and propose a modification to the batch norm VGG-16 architecture that is more suited to the task of classifications in paintings (batch normalized VGG-16 is used for this section because we observed that the models stop training after a few epochs). As we go deeper in the network, the exact pixel information is not preserved but the content is preserved. The early layers capture some of the finer textures contained within the image, whereas the deeper layers captures more high-level elements of the image's style. Thus, it makes sense to use outputs from the earlier convolutional layers in addition to the final convolutional layer for the task of classification. Gatys et al. [5] found that the best result for style transfer was achieved by taking both shallow and deep layers as the style representation for an image. He defined a style reconstruction loss which was computed from the gram matrices of the feature maps from all the convolutional layers. Gram matrices allows for the consideration of all pairwise interactions similar to a quadratic kernel expansion.

Similar to multi-task learning where we have a separate classification head for each task, we attach classification heads to the Gram matrices obtained from some convolutional layer. There are 5 convolutional blocks (Figure 4.1). The feature maps from the final convolutional layers for each block are chosen for forming the gram matrices. Figure 6.1 shows a diagram of our new architecture. Table 6.1 shows the number of filters for the different convolutional layers

and the size of the Gram matrices. We take the feature maps for each convolutional layer and find its gram matrix. The Gram matrix of a feature map is given by the outer product of the feature map reshaped as c*(h*w), where c is the number of filters, h and w are the height and width of the convolutional layer output. After calculating the gram matrix, we reshape the c*c gram matrix to a c*c 1-dimensional vector, where c is the number of filters in that convolutional layer. We then apply signed square root and L2 normalization (these two steps generally improve performance [9]). The c*c dimensional vector is then fed to a classification head. The classification head consists of 2 to 3 fully connected layers depending on the experiment. The classification heads architectures for three fully connected layer and two fully connected layers are shown in Figure 6.2 and Figure 6.3 respectively. The diagrams are shown for style classification which is why the last layer has 27 as the output dimension since there are 27 styles in the dataset. The fully-connected layers were initialized from a zero-mean Gaussian distribution with a standard deviation of 0.01. The biases were initialized to 0.

Table 6.1: Number of filters and gram matrix size for each convolutional layer

Convolutional Layer	**Number of filters**	**Gram Matrix size**	**Input size to fully connected layer**
Conv-1_2	64	64*64	4096
Conv-2_2	128	128*128	16384
Conv-3_3	256	256*256	65536
Conv-4_3	512	512*512	262114
Conv-5_3	512	512*512	262114

Figure 6.1: Diagram showing our new architecture for improving classifications in paintings. The task is style classification which is why the last layer has 27 neurons

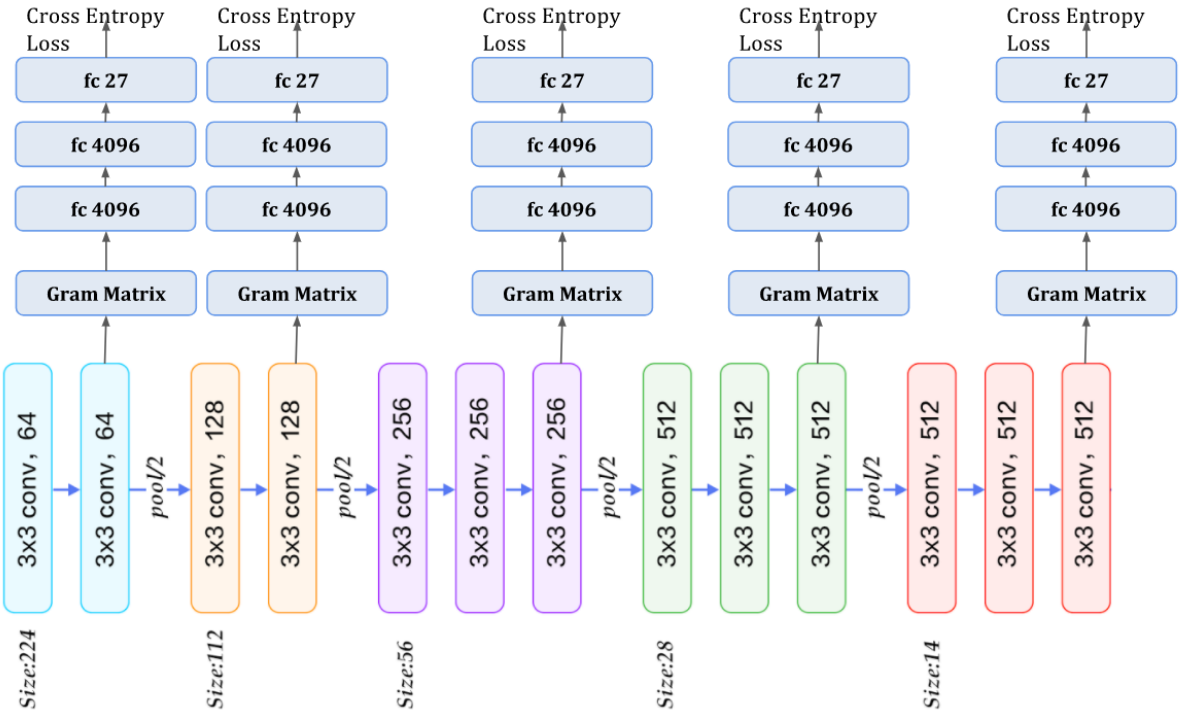


Figure 6.2: Diagram showing the classification head used for 3 fully-connected layer model for style classification

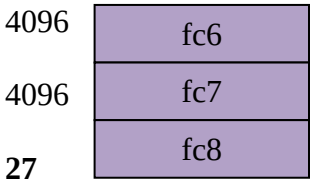
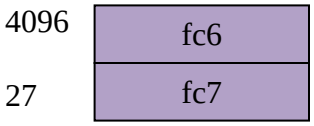


Figure 6.3: Diagram showing the classification head used for 2 fully-connected layer model for style classification



Since the task is a multi-class classification task, we use the same cross-entropy loss as in Section 4. Thus, each classification head has its own loss and its separate accuracy for the test set. We perform two different types of experiments as described below:

- *Style training*: We train several style models each with a single classification head at different convolutional layer positions
- *Expanded artist training*: We train several expanded artist models each with a single classification head at different convolutional layer positions

We used the same data augmentation techniques for both training and testing as explained in Section 4. All models are trained using stochastic gradient descent (SGD) with momentum and weight decay. The batch size was kept small to accommodate the large feature dimensions. The momentum value was fixed at 0.9 and the weight was halved every 10 epochs. The learning rate that gave the best results was empirically found to be 0.001. All the experiments were carried out using PyTorch on NVIDIA Tesla K-40.

Table 6.2 contains the results (accuracy percentage) of style classification for style training described above. The first row corresponds to fine-tuning the 2-layer classification head for the different convolutional layer positions. The second row corresponds to fine-tuning the 3-layer convolutional head for the different convolutional layer positions. The last row corresponds to style accuracies obtained by fine-tuning the entire model for a 3-layer convolutional head. The columns correspond to the different convolutional layers where the classification heads can be attached. Due to the high-dimension of the gram matrices of conv3_3, conv4_3 and conv5_3 we did not carry out the experiments. Techniques to reduce the high-dimensionality of conv3_3, conv4_3 and conv5_3 gram matrices have been discussed in the next section.

Table 6.2: Accuracy percentage for style classification for individual learning using our new VGG architecture

Convolutional Layer	**Conv1_2**	**Conv2_2**
2 layer FC fine-tuning	35.56	35.07
3 layer FC fine-tuning	34.85	33.64
3 layer full fine-tuning	54.98	52.31

From Table 6.2 we see that 2 fc layers in the classification head performs slightly better than 3 fc layers in the classification head. By fine-tuning the entire model, we achieve higher classification scores, a trend that has been seen throughout this thesis. The classification scores obtained by training the entire model using conv1_2 and conv2_2 are comparable with the previous state-of-the-art style classifier [2] which was 54.50%. This suggests that the network is able to learn discriminative style features from the first few layers with the same level of accuracy as that of the previous state-of-the-art.

Table 6.3 contains the results (accuracy percentage) of expanded artist classification. The first row corresponds to fine-tuning the 2-layer classification head for the different convolutional layer positions. The second row corresponds to fine-tuning the 3-layer convolutional head for the different convolutional layer positions. The last row corresponds to expanded artist accuracies obtained by fine-tuning the entire model for a 3-layer convolutional head. The columns correspond to the different convolutional layers where the classification heads can be attached. Due to the high-dimension of the gram matrices of conv3_3, conv4_3 and conv5_3 we did not carry out the experiments.

Table 6.3: Accuracy percentage for expanded artist classification for individual learning using our new VGG architecture

Convolutional Layer	**Conv1_2**	**Conv2_2**
2 layer FC fine-tuning	24.56	24.92
3 layer FC fine-tuning	25.96	25.69
3 layer full fine-tuning	34.66	34.89

From Table 6.3, we see that 3 fc layers in the classification head performs slightly better than 3 fc layers in the classification head which is the opposite of what was observed in style classification. By fine-tuning the entire model, we achieve higher classification scores. The classification scores are considerably less than what the multi-task gave us, however we feel this is an interesting direction and we would be able to make more conclusive arguments about our new architecture with more experiments.

It would be interesting to observe the degree of feature overlapping between the initial convolutional layers and the standard VGG fine-tuning, i.e. do they learn complementary features or the same features and if the new model trained with these combined features be able to beat our best style classifier from Section 5.2. We discuss this further in the next section.

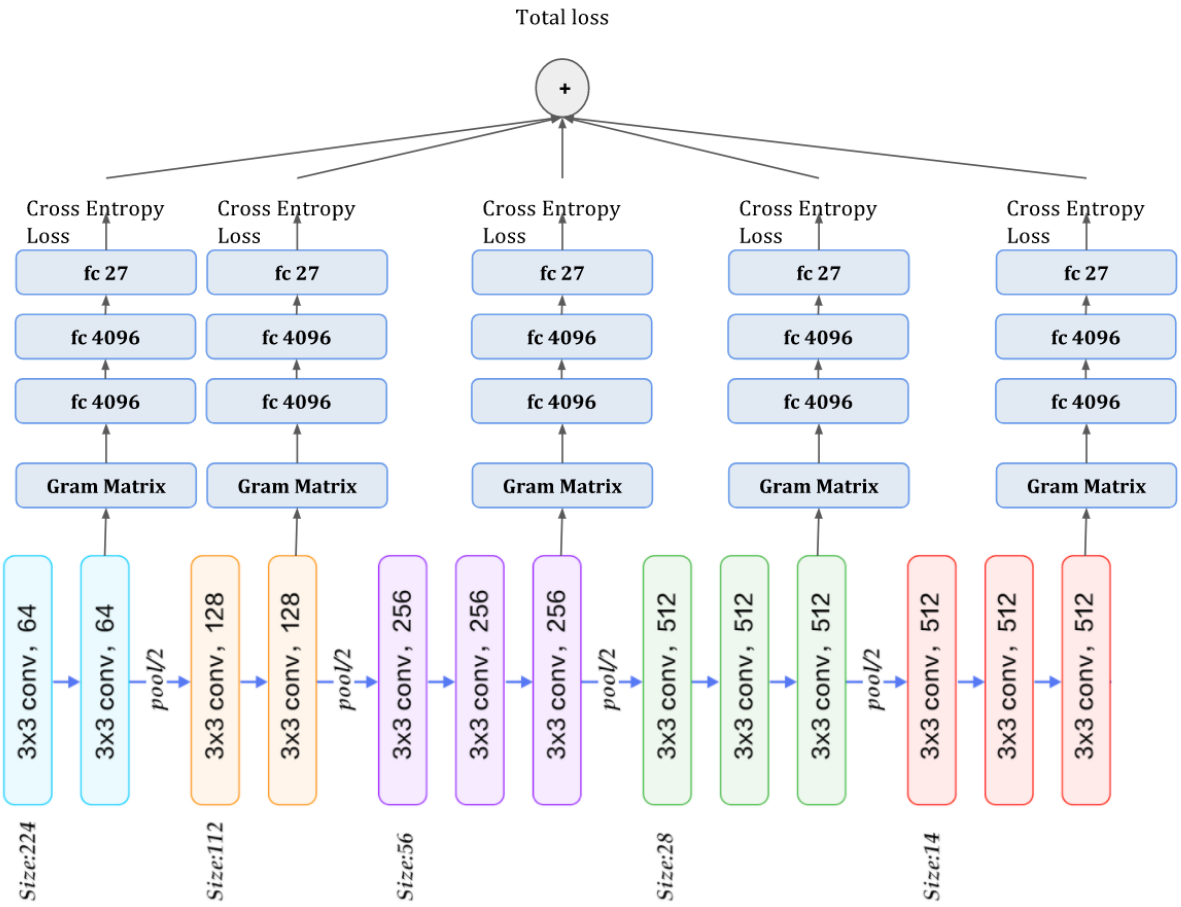
CHAPTER 7: FUTURE WORK

In the last section, we only played around with conv1_2 and conv2_2. However, we did not run any experiments for conv3_3, conv4_3 and conv5_3. The reason for this is because these layers produced a gram matrix of size 512*512 (for conv4_3 and conv_5_3) which when re-arranged as a 1-d vector gives a size of ~260,000. This is a very high-dimensional vector and would 260,000 * 4096 number of parameters in the first fully-connected layer. This makes computation practically infeasible. However, it would be interesting to be able to run the same experiments for these layers to observe any pattern or trend.

Gao et al. [11] came up with compact bilinear pooling methods which do away with the need to compute these computationally expensive outer products while still achieving similar results. With their compact pooling methods, the 260,000 dimensional vector is reduced to only ~10,000 dimensions. This is a drastic reduction in dimension size; while still keeping the accuracy high. We plan to implement this pooling to be able to run experiments on conv3_3, conv4_3 and conv5_3 convolutional layers in the future.

In the last section, the models were trained only using a specific convolutional layer at a time. However, we can potentially train a model using the outputs from all the 5 chosen convolutional layers. This model would have multiple classification heads with the final loss computed as sum of the individual losses of these heads and then back propagated accordingly. The final accuracy would be calculated by using a majority vote from amongst the predictions of the different classifiers. Figure 7.1 shows a diagram of the model we are proposing to train.

Figure 7.1: Diagram showing our proposed architecture where we plan to use all the convolutional layers for training. The task is style classification which is why the last layer has 27 neurons



REFERENCES

[1] Ginosar, Shiry, et al. "A century of portraits: A visual historical record of american high school yearbooks." *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2015.

[2] Tan, Wei Ren, et al. "Ceci n'est pas une pipe: A deep convolutional network for fine-art paintings classification." *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016.

[3] Saleh, Babak, and Ahmed Elgammal. "Large-scale classification of fine-art paintings: Learning the right metric on the right feature." *arXiv preprint arXiv:1505.00855* (2015).

[4] Toshev, Alexander, and Christian Szegedy. "Deeppose: Human pose estimation via deep neural networks." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014.

[5] Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "A neural algorithm of artistic style." *arXiv preprint arXiv:1508.06576*(2015).

[6] Johnson, Justin, Alexandre Alahi, and Li Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution." *European Conference on Computer Vision*. Springer International Publishing, 2016.

[7] Karayev, Sergey, et al. "Recognizing image style." *arXiv preprint arXiv:1311.3715* (2013).

[8] Palermo, Frank, James Hays, and Alexei A. Efros. "Dating historical color images." *European Conference on Computer Vision*. Springer, Berlin, Heidelberg, 2012.

[9] Lin, Tsung-Yu, Aruni RoyChowdhury, and Subhransu Maji. "Bilinear cnn models for fine-grained visual recognition." *Proceedings of the IEEE International Conference on Computer Vision*. 2015.

[10] Jae Lee, Yong, Alexei A. Efros, and Martial Hebert. "Style-aware mid-level representation for discovering visual connections in space and time." *Proceedings of the IEEE International Conference on Computer Vision*. 2013.

[11]Gao, Yang, et al. "Compact bilinear pooling." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016.

[12] Crowley, Elliot J., and Andrew Zisserman. "In Search of Art." *ECCV Workshops (1)*. 2014.

[13] Crowley, Elliot, and Andrew Zisserman. "The State of the Art: Object Retrieval in Paintings using Discriminative Regions." *BMVC*. 2014.

[14] Khan, Fahad Shahbaz, et al. "Painting-91: a large scale database for computational painting categorization." *Machine vision and applications* 25.6 (2014): 1385-1397.

[15] Ruder, Sebastian. "An overview of multi-task learning in deep neural networks." *arXiv preprint arXiv:1706.05098* (2017).

[16] https://www.wikiart.org/

[17] http://arkyves.org/

[18] https://www.artsy.net/

[19] https://www.behance.net/

[20] https://www.artnet.com/

[21] https://artuk.org/

[22] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.

[23] Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.

[24] Zhou, Bolei, et al. "Learning deep features for scene recognition using places database." *Advances in neural information processing systems*. 2014.

[25] Simonyan, Karen, and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition.” arXiv preprint arXic: 1409.1556(2014)

[26] Everingham, Mark, et al. "The pascal visual object classes (voc) challenge." *International journal of computer vision* 88.2 (2010): 303-338.

[27] Krizhevsky, Alex, and Geoffrey Hinton. "Learning multiple layers of features from tiny images." (2009).

[28] Russakovsky, Olga, et al. "Imagenet large scale visual recognition challenge." *International Journal of Computer Vision* 115.3 (2015): 211-252.

[29] Vittayakorn, Sirion, Alexander C. Berg, and Tamara L. Berg. "When was that made?." *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*. IEEE, 2017.

[30] D. G. Stork. Computer vision and computer graphics analysis of paintings and drawings: An introduction to the literature. In Computer Analysis of Images and Patterns, pages 9–24. Springer, 2009.

[31] T. E. Lombardi. The classification of style in fine-art painting. ETD Collection for Pace University. Paper AAI3189084., 2005

[32] R. Sablatnig, P. Kammerer, and E. Zolda. Hierarchical classification of paintings using face- and brush stroke models. 1998.

[33] M. V. Fahad Shahbaz Khan, Joost van deWeijer. Who painted this painting? 2010.

[34] Y. Bar, N. Levy, and L. Wolf. Classification of artistic styles using binarized features derived from a deep neural network. 2014

[35] Yosinski, Jason, et al. "How transferable are features in deep neural networks?." *Advances in neural information processing systems*. 2014.

[36] https://en.wikipedia.org/wiki/Boris_Kustodiev

[37] https://en.wikipedia.org/wiki/Painting

APPENDIX A: SUPPLEMENTARY

Table A.1: List of classes for the expanded artists set

Task type	Classes
Expanded artists	Hans Hofmann; Andre Derain; Hiro Yamagata; Max Pechstein; Paula Modersohn-Becker; John Everett Millais; Leon Bakst; Jacopo Pontormo; Karl Bodmer; Walter Battiss; Dmitry Levitzky; John French Sloan; Jamie Wyeth; Lorenzo Lotto; John William Waterhouse; Domenico Ghirlandaio; Jean Fouquet; Heorhiy Narbut; Morris Louis; Frank Stella; Maria Primachenko; William Blake; Vladimir Makovsky; Theodore Gericault; Vasily Tropinin; Franz Marc; Winslow Homer; Carl Larsson; Edward Hopper; John Constable; Wilhelm Kotarbinski; Fyodor Vasilyev; William H. Johnson; Kitagawa Utamaro; Willard Metcalf; Jean-Francois Millet; Henri Rousseau; Arnold Bocklin; Caspar David Friedrich; Paul Klee; Giovanni Battista Tiepolo; Jan Steen; Guy Rose; Konstantin Bogaevsky; Henri-Edmond Cross; Robert Julian Onderdonk; Sandro Botticelli; Mark Rothko; Kazimir Malevich; Moise Kisling; Diego Velazquez; Mikhail Nesterov; Brice Marden; Guido Reni; Helen Frankenthaler; Anders Zorn; William Hogarth; Sergey Solomko; John Atkinson Grimshaw; Thomas Cole; Theodore Rousseau; Alphonse Mucha; Benozzo Gozzoli; Lovis Corinth; Andrea Mantegna; Maurice Utrillo; Maurice Quentin de La Tour; Hans Holbein the Younger; Jacob Jordaens; Gustav Klimt; Eugene Delacroix; Jacek Malczewski; Hans Memling; Maxime Maufra; Ivan Kramskoy; Pierre Bonnard; Henri Martin; El Greco; Mstislav Dobuzhinsky; Fernando Botero; Gene Davis; Gustave Moreau; Georges Seurat; M.C. Escher; Fra Angelico; Mikalojus Ciurlionis; Karl Bryullov; Raoul Dufy; Antoine Blanchard; James McNeill Whistler; Niko Pirosmani; Anthony van Dyck; Canaletto; Thomas Gainsborough; Edward Burne-Jones; Frans Hals; Michelangelo; Bartolome Esteban Murillo; Pietro Perugino; Arkhip Kuindzhi; Edvard Munch; Hieronymus Bosch; Vasily Perov; Paolo Veronese; Joseph Wright; Raphael; Mikhail Vrubel; Theo van Rysselberghe; Felix Vallotton; Dante Gabriel Rossetti; Juan Gris; Sir Lawrence Alma-Tadema; Leonardo da Vinci; Nikolay Bogdanov-Belsky; Koloman Moser; Gustave Caillebotte; Andy Warhol; Ilya Mashkov; Joshua Reynolds; Edouard Cortes; Viktor Vasnetsov; Valentin Serov; Lucas Cranach the Elder; Vasily Vereshchagin; Jan Matejko; William Turner; Edouard Manet; Tintoretto; Orest Kiprensky; Kuzma Petrov-Vodkin; John Henry Twachtman; Aleksey Savrasov; Giovanni Boldini; Aubrey Beardsley; Berthe Morisot; Ivan Bilibin; Vasily Polenov; Katsushika Hokusai; Konstantin Somov; Titian; Ferdinand Hodler; Gustave Loiseau; Fernand Leger; Lucian Freud; Gustave Courbet; Egon Schiele; Vasily Surikov; Georges Braque; Henri Fantin-Latour; Mary Cassatt; David Burliuk; Thomas Eakins; Konstantin Korovin; Utagawa Kuniyoshi; Sam Francis; Amedeo Modigliani; Joaquin Sorolla; Zinaida Serebriakova; Konstantin Makovsky; Francisco Goya; Peter Paul Rubens; Maurice Prendergast; William Merritt Chase; Ernst Ludwig Kirchner; Henri de Toulouse-Lautrec; Paul Gauguin; James Tissot; Isaac Levitan; Alfred Sisley; Odilon Redon; Camille Corot; Henri Matisse; Raphael Kirchner; Ivan Shishkin; Ilya Repin; Childe Hassam; Eugene Boudin; Ivan Aivazovsky; Paul Cezanne; Martiros Saryan; Salvador Dali; Edgar Degas; Boris Kustodiev; Gustave Dore; Marc Chagall; Rembrandt; John Singer Sargent; Pablo Picasso; Albrecht Durer; Camille Pissarro; Pyotr Konchalovsky; Claude Monet; Pierre-Auguste Renoir; Nicholas Roerich; Vincent van Gogh

Figure A.1: Confusion matrix for date classification (temporal resolution of 10) using VGG-fcs

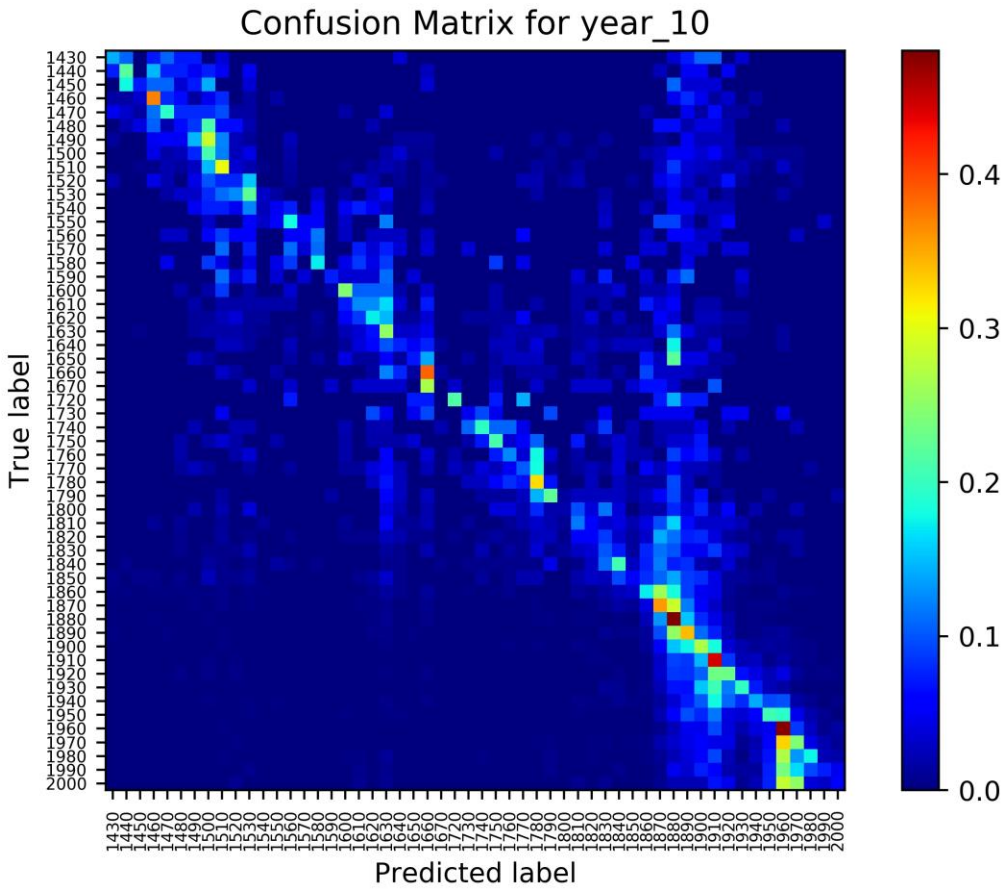


Figure A.2: Confusion matrix for date classification (temporal resolution of 100) using VGG-fcs