Describing Artworks Using Convolutional Neural Networks

Manikanta Kotaru
Stanford University
mkotaru@stanford.edu

Varun Vijay Kumar
Stanford University
varun3@stanford.edu

Abstract

In this paper, we describe an approach based on Convolutional Neural Networks (CNNs) to classify artworks by artistic technique, time period, genre, and produce image captions. We assemble a large dataset of high-resolution images (40,000) and use it to train several neural network models, achieving promising results. We also propose to use the features we learn to illustrate various categories.

1. Introduction

The internet has made it incredibly easy to access digital photographs of artworks, from masterpieces in museum collections to new productions uploaded to a blog. Further, these images are often implicitly indexed by the textual data present alongside them, which might include mentions of genres or artists, making it possible to query artworks. Nevertheless, a richer machine understanding of the images themselves would broaden the scope of searches to images that annotations, allow the classification of new artworks to be automated, and enable other exciting applications such as the automatic generation of illustrative examples of techniques and genres.
Our class project will pursue this goal by applying the recent advances in Convolutional Neural Networks (CNNs) to the task of classifying and captioning artworks.

2. Problem Statement

2.1. Image Classification

It is of interest to identify the artist of a particular painitng, the technique used by the artist, the period during which the painting, and emotions conveyed by the artwork. In the first part of our project, we want to investigate the information that can be obtained from the painting itself. Sepcifically, we want to classify images of paintings by several labels, including time period of the artwork, artistic technique and genre. To achieve this task, we will use a database of approximately 40,000 high-resolution images of Western artworks for training a neural network architecture. We will then analyze the relative accuracies obtained on different classification tasks. This work provides very good understanding of the difficulty of identifying various characteristics of the artworks.

2.2. Image Captioning

In museums, we generally observe some text adjoining the painting describing the artwork because it is of interest to know the story behind the artwork and the story conveyed by the artwork. In the second part of our project, we will investigate if we can obtain some description of the painting given an image of the artwork. We will use the same database of high-resolutions images, supplemented by several sentences describing each image. After training, our aim is to generate a description of a new image.

3. Related Work and Data Collection

Our data is drawn from the World Gallery of Art [1], a virtual museum and search-able database of reproductions of Western art. The database provides a catalog of collection in the form of a csv file, with a link to a page for each artwork. Each artwork is accompanied a description of few sentences, and information about the artist, time period, technique and school. We automatically scrape the webpage to retrieve an image of the artwork, artwork information, and a paragraph describing it. There are over 39000 images and the descriptions lengths varied from 0 sentences to couple of tens of sentences.

Previous work in the area of artwork classification has relied on small datasets and handcrafted features. For instance, J. Zukjovic et. al.ụse a dataset of only 353 paintings belonging to 5 genres and implement features Steerable Pyramids and Canny edge detection. While the authors argue that using images of varying provenance and dimensions makes their method robust, we demonstrate robustness by showing results on large validation and test sets ($>$ 2000 images).

The task of classifying and captioning our dataset is made particularly difficult by the variation in frequencies between labels and in the number of sentences per image. For instance, of the 2400 technique labels we analyzed,

1630 labels were found in only a single image, while the most common label, 'oil on canvas', was found over 13,000 times. Similarly, the number of descriptions of an image ranged from 0 to 50, in contrast to the more extensive MS-COCO dataset [2], which supplies five sentences for every image.

4. Generating Image descriptions

4.1. Preprocessing

The images were preprocessed by scaling them to a common size of 224 by 224, which was the input size of the pre-trained models we worked with, and subtracting the mean image.

The preprocessing for the image captioning task was somewha more involved. The goal was to produce a single sentence description of an artwork given an image of the artwork. During training, the input is images of artwork and paragraph descriptions corresponding to the image. The paragraph descriptions are split into multiple sentences. We removed sentences that contain any numeral in them. This occured mostly due to artworks containing information about the year the artwork is made or corresponds to some events of the artist of the arwork. We believe that this information is hard to deduce from the image pixels. And more importantly, adding all the years blows the state space of the words. So, we decided to trim the number of training captions by ignoring the captions which contain a numeral in them. Further, the images are of different input sizes. So, as described in the artwork classification task, we resized all images to 224×224 size. We considered only color images.

4.2. Architecture

The description of the artworks is composed of words which describe the objects visible in the artwork and also hidden deeper meanings and emotions conveyed by the artwork. Our insight is that these words correspond to different spatial portions of the image. This places us in the framework described in [3]. So, we adopted the framework in [3] for generating descriptions for the image. At a high level, the system in [3] maps the words and image regions into a common multimodal embedding and learns the embedding representations so that the words and image regions describing a similar object or feeling fall close to each other in the embedding representation. For completeness, we summarize the framework below. We refere to [3] for detailed description.

4.3. CNN Architecture

First, CNN is trained for a particular target class to extract relevant features. For the artworks, we used weights from 16-layer VGGNET trained on ImageNet Detection challenge. The layers in our architecture are as follows:

1) Two convolutional layers with 64 filters of size 3 by 3, followed by a pooling layer
2) Two convolutional layers with 128 filters of size 3 by 3, followed by a pooling layer
3) Three convolutional layers with 256 filters of size 3 by 3, followed by a pooling layer
4) Two sets of three convolutional layers with 512 filters of size 3 by 3, each time followed by a pooling layer
5) Two fully connected layers with 4096 units
6) A fully connected layer mapping the 4096 features to the classes

The VGGNet architecture was chosen because it has been demonstrated to have better performance than competing architectures, such as GoogLeNet, on transfer learning tasks.

4.4. RNN Architecture

A regional Convolution Neural Network built on top of CNN is used to detect different regions of the image. The system uses top 19 detected locations and computes representations for each bounding box using equation 1

$$v = W_m[CNN_{\theta_c}(I_b)] + b_m, \tag{1}$$

where $CNN_{\theta_c}(I_b)$ is the feature vector extracted from the pre-trained CNN. W_m maps these features into embedding space and b_m are biases. Let h be the dimension of vectors in embedding space. Similarly, words are converted into h dimensional vectors using Bidirectional Recurrent Neural Networks. Specifically, BRNN takes a sequence of N words, and performs the following operations to include context of the words.

$$\begin{aligned} x_t &= W_w I_t \\ e_t &= f(W_e x_t + b_e) \\ h_t^f &= f(e_t + W_f h_{t-1}^f + b_f) \\ h_t^b &= f(e_t + W_b h_{t+1}^f + b_b) \\ s_t &= f(W_d(h_t^f + h_t^b) + b_d). \end{aligned} \tag{2}$$

where I_t is indicator function for t^{th} word, W_w is word embedding matrix (we map each word to 512-dimensional space in our experiments), h_t^f and h_t^b are two processing vectors in the forward and reverse directions repectively and the vectors lie in $h-$ dimensional embedding space. The final embedding representation is obtainined by ReLU activations on a function of both forward and backward processing vectors. Note f is standard ReLU activation function.

Then the similarity of words and images in the embedding space is computed by using the dot product between the corresponding vectors in the embedding space. The

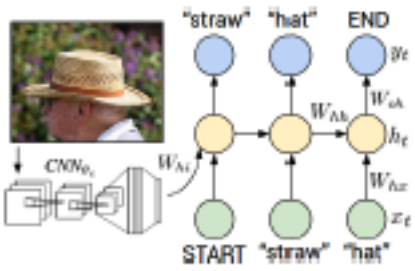

Figure 1. Description of RNN training process. RNN takes a word and context as input and produces likelihood distribution for next word.

words and the image regions are aligned using sum of the similarities across all object-word pairs.

The network is trained by providing the image features frim pre-trained CNN and first word (which is special START token) and the secnd word is desried output. Then the network proceeds forward in time with second word as an additional input and the goal is to predict the third word and so on till the special end of sentence token END is reached. The equaitons describing the process are shown below in Equation 3.

$$\begin{aligned} b_v &= W_{hi}[CNN_{\theta_c}(I)] \\ ht &= f(W_{hx}x_t + W_{hh}h_{t1} + b_h + I(t=1) * b_v) \\ yt &= \text{softmax}(W_{oh}h_t + b_o). \end{aligned} \tag{3}$$

The RNN training process is best illustrated in Figure 5 from [3].

4.5. Implementation

We used weights of 16-layer VGGNET pre-trained on ImageNet detection challenge to extact the features from the image. A 512-length word embedding is chosen to map words to vectors. The CNN was trained using our own framework in Lasange and the combined CNN-RNN was trained in neuraltalk2. The experiments were conducted using NVIDIA GRID K520 GPU.

Design Choices: A major design choice was in constructing the vocabulary. Given the nature of the dataset, there are lots of proper nouns describing persons and places. Also, a huge portion (more than 30 precent) of the words occur lonly once. Mapping all the words that occur less than a threshold number of times, say 5, to a special UNKNOWN word token, then we will run into case where a sentence full of UNKNOWN tokens may achieve smaller loss which is undesirable. Mapping all unique words to different tokens blows up the state space and the network cold train. Say, we attempted two approaches - one, using a small dataset of 1000 images and mapping all unique words to differnet tokens and two, take a large dataset of 10000 images but map all the words which occur less than 6 times to the UNKNOWN token.

Another major design choice was whether to train CNN specific to artworks dataset and with a target class specific to artworks like artist, or should we use CNN pre-trained on some other standard dataset with a standard data class. We decided to use pre-trained CNN.

Our training infrastructure also had to allow for the fact that the entire training dataset could not fit in memory. Instead, we loaded and preprocessed images on the fly while sampling batches.

We also decided to use the Adam update rule throughout as it has been empirically shown to produce the best results and requires less hyperparameter tuning. The main feature of the update rule is its use of first and second order momentum.

4.6. Experiments

4.6.1 CNN

Our training procedure for the CNN was as follows. Beginning with the highest learning rate that did not cause the loss to increase to infinity, we halved the learning rate whenever the loss began to stagnate. When it became apparent that the network was overtraining, we added batch normalization to a single CNN layer (starting from the first layer) and repeated the process.
The results are summarized in the table above. We see that adding batch normalization helps considerably and greater performance gains could perhaps be achieved by continuing the process. We also note that our maximum validation accuracies are competitive with those achieved by J. Zujovic et al.
Our test accuracies were 0.42 on the school labeling problem and 0.49 on the technique labeling problem, indicating that our model performs quite well on unseen data.
In order to analyze the mistakes made by our model, we list the most commonly confused pairs of techniques:
True label: Oil on panel; Prediction: Oil on canvas
True label: Oil on wood; Prediction: Oil on canvas
True label: Fresco; Prediction: Oil on canvas
True label: Oil on oak panel; Prediction: Oil on canvas
True label: Oil on copper; Prediction: Oil on canvas
True label: Fresco; Prediction: Manuscript
We note that most of these pairs contain oil painting techniques on various materials, which are naturally difficult to distinguish. However, we might expect our model to perform better on some of these examples. For instance, frescos should be clearly distinguishable from manuscripts.
The confusion matrices show a similar trend. The mistakes

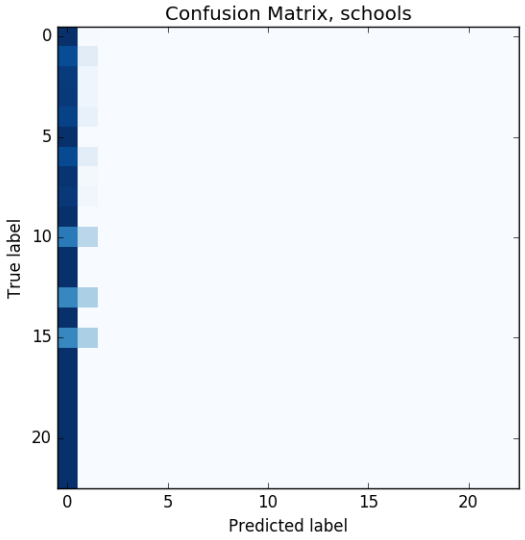

Figure 2. Confusion matrix for schools.

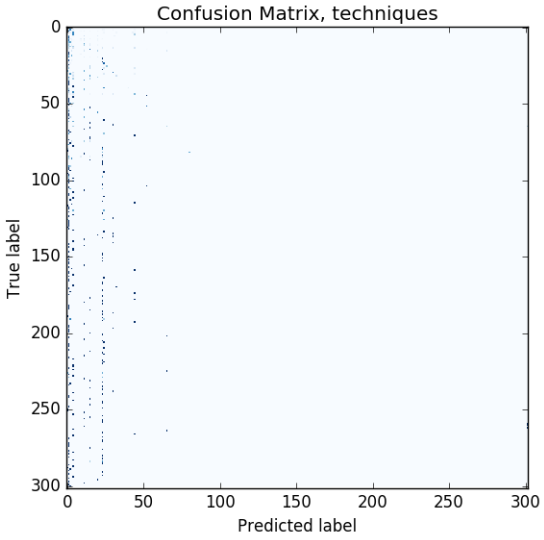

Figure 3. Confusion matrix for schools.

Layers	Accuracies
0	(37, 38)
1	(38, 54)
2	(42, 56)
3	(48, 55)

Table 1. Number of layers with batch normalization and the maximum validation accuracy (on schools, on techniques) achieved.

Figure 4. Jean Hey's Nativity. The painter (originally Flemish) worked in France. Our model misclassified the work as Italian.

Figure 5. Jan Bruegel the Elder's Animals Entering the Ark. The image was incorrectly identified as an oil on canvas work. It is actually oil on copper.

made by both the school and technique models are clustered around zero on the x-axis but spread evenly along the y-axis. The clustering is due to our procedure for assigning numeric values to labels, which was to iterate over the data and allocating the next integer to previously unseen labels. The most common labels were likely to be seen early and hence to be assigned numbers close to 0. The actual assignments confirm this explanation The first and second technique labels are 'Fresco' and 'Oil and canvas' respectively, while the first and second school labels are 'Italian' and 'Dutch.' These are also the most common classes. Therefore, the clustering is simply due to the fact that the most mistakes were made in the most frequent labels.

We also present instances of artworks that were misclassified by school and technique.

4.7. Experiments for generating image descriptions

We initially conducted experiments with large dataset of 10000 images. We chose a training-validation split of 80-20 i.e., we randomly chose 8000 of the 10000 images for training purposes and the rest 2000 for validation. Multiple captions are generated for each image using the paragraph descriptions scraped from website. However, for training purposes, we uniformly sampled five captions of those available for each image. Words occurring five times or less are assigned a special token called UNK (or UNKNOWN) to make sure the number of word vectors is reasonable. We trained the network in batches of size 16. The validation loss as the training progressed is illustrated in the Figure 6.

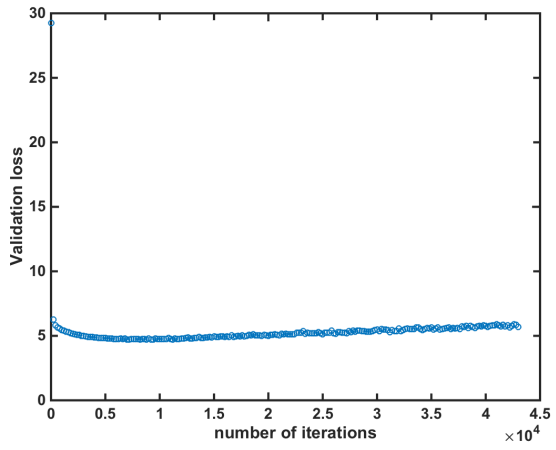

Figure 6. Validation loss for RNN trained on large dataset of 10000 images.

We observed that the training loss flattened within few thousands of iterations. We then finetuned the 16-layer VGGnet in order to reduce the training loss. We started with the network obtained from previous experiments for initialization of weights and trained the CNN and RNN together in this step. We trained in batches of size 6 for this experiment. We plot the validation loss in Figure 7. However, we observed that the validation loss essentially stagnated. We believe that the primary reason for the stagnation is the presence of large number of UNKNOWN tokens in the captions making it hard for the network to learn.

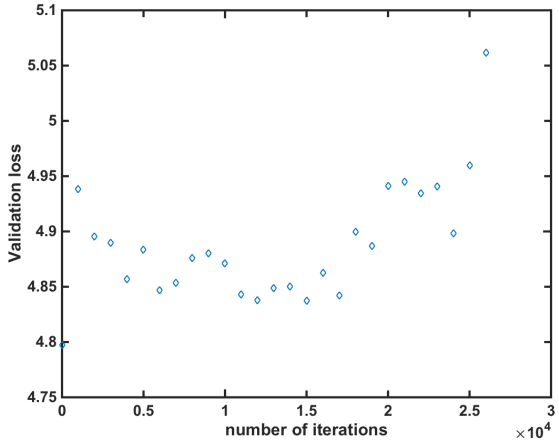

Figure 7. Validation loss for combined CNN-RNN training on large dataset of 10000 images.

We will now look at qualitative performance of image descriptions by examining the captions generated for few images. The validation loss flattened within a couple of iterations and did not change even when the learning rate is reduced or when the training ran for a longer time for half a day. This strongly suggests network stopped learning. This can be seen by same captions generated for all the images. The three most popular captions are 'the picture shows a detail of the central panel of the triptych of the adoration of', 'the painting is signed and dated lower right', and 'this painting is one of the few surviving works by cigoli UNK'. Few representative images for these captions are illustrated in Figures 8, 9, and 10 respectively.

Figure 8. Description generated by the trained network is 'the picture shows a detail of the central panel of the triptych of the adoration of'

We hypothesized that the performance is poor because the CNN has learned features from ImageNet images which may be quite different from images of artwork. So, we backpropagated into the CNN as described above. However, this did not improve the captions generated for a large number of images. However, for few images, the generated description is one among the sentences scraped from website. An illustrative example is presented in Figure 11

We further investigated by playing with a relatively small number compared to the above extensive experiment. We considered a set of 1000 images. However, this time we included all the words that occurred in the captions of these images in the vocabulary. We used a training-validation split of 75-25 i.e., 25 percent of images are randomly selected for validation and rest 75 percent of the images are considered for training. We trained using a batch size of 16. We initially trained only the RNN leaving the weights of the CNN network unchanged from the downloaded model [**?**].

Figure 9. Description generated by the trained network is 'the painting is signed and dated lower right'

Figure 10. Description generated by the trained network is 'this painting is one of the few surviving works by cigoli UNK'

We plot the validation loss in Figure 12.

We observed that the validation loss increased slightly while the training loss decreased. This strongly suggests overfitting. We further finetuned the CNN weights on the small dataset. We show the validation losses in Figure 13.

Figure 11. Description generated by the trained network is 'this painting is one of the most famous UNK of the UNK UNK' which is close to one of the ground truth captions

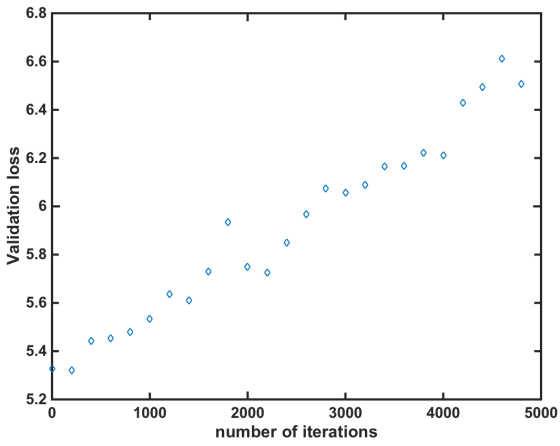

Figure 12. Validation loss for RNN training on small dataset of 1000 images.

This further justified our hypothesis that the model overfits for datasets of size 1000.

Even with overfitting, we observed that the captions generated for different images are pretty similar. The most popular captions are 'this painting is one of the series of the virgin of the', 'the picture shows a detail of the predella', 'the picture shows the right side of the fresco', and 'the picture shows a detail of the predella'. However, working with smaller dataset did generate meaningful captions for images

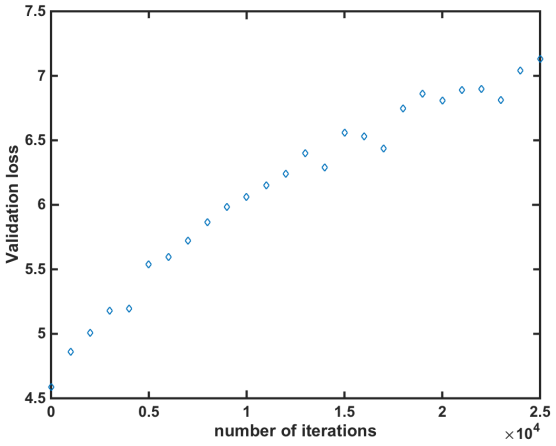

Figure 13. Validation loss for combined CNN-RNN training on small dataset of 1000 images.

of few of the artworks. An example is illustrated in Figure 14.

Figure 14. Description generated by the trained network is 'this painting depicts a stilllife with fruit basket fruits and shellfish' which is close to one of the ground truth captions. The description is also relevant to the information conveyed by the painting which was the ultimate goal of this project.

References

[1] http://www.wga.hu/.

[2] http://mscoco.org/.

[3] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.