Style Classification in Posters

Tim Löhr
Friedrich Alexander University
Pattern Recognition Lab
tim.loehr@fau.de

***Abstract*—Digitalization, Big Data and data collection is the keyword of many companies nowadays. Even museums digitize their artworks. A local museum from Bayreuth photographed 17786 posters showing invitations to exhibitions or advertising for specific events. Our department received this data to find patterns in it. More specifically, one shall order the posters in clusters to sort them logically. Furthermore, if there is a new poster, the clustering should present all posters with similar attributes. Since how to do it was relatively open, I tried four different methods to cluster the posters. First, I used a supervised approach with a neural network and another dataset from WikiArt to perform Transfer Learning. The following two approaches are based on the text printed on the posters. I extracted the text and first performed clustering with the BERT Topic Modeling approach, then with the LDA Topic Modeling approach. Lastly, I focused on the image by extracting the features with a pre-trained neural network and then performing PCA and K-Means to find a predefined number of clusters. The three unsupervised approaches produced some output that is workable but is not satisfactory enough.**

***Index Terms*—neural networks, LDA, BERT, classification, poster, wikiart, museum, clustering**

I. Introduction

Data is gathered almost everywhere. For this project, a local museum in Bayreuth took 17786 posters showing different kinds of museum exhibitions. The pictures came in different sizes and with some of them showing only the poster itself, but with some, the background or tables are also visible. The providers of the dataset aim to sort the posters in some way. It shall therefore be researched if the images are sufficient enough to cluster and group them. For this project, I used different techniques to analyze the potential clusters of these images. Figure 1 illustrates an example poster from the dataset. Some of the problematic properties of the images can already be seen in this picture.

Fig. 1. Sample image from the poster dataset from the Bayreuther museum. It can be seen that we have a white frame on the right side of the image and black frames on the top and at the bottom. Also, both the German and the English language is present as text in the image. Luckily the rotation is correct, but that is only for 25% of images the case.

II. Problem Statement

The thousands of posters come in different shapes and qualities. Not only is this problematic for training a neural network, which requires a fixed-size input, but the posters also come in random rotation. For us as humans, it is easily detectable in which direction the poster has to point to be correct, but not for the neural network. Furthermore, the images amount to a total of 7 Gigabytes, and this requires a lot of computing power to train neural networks or other kinds of machine learning algorithms. This project aims to find clusters within these posters to somehow search through the 17786 pictures or possibly find similar posters given a new poster. At least 50% of the posters contain more than one word, so a text-based clustering or searching algorithm could be sufficient. Furthermore, some posters like Figure 1 are designed in a specific style, so maybe transfer learning and classification could do the trick. Therefore, I want to find out if the poster images can be ordered somehow, but the approach needs to be evaluated and tested.

The research question for this project can therefore be

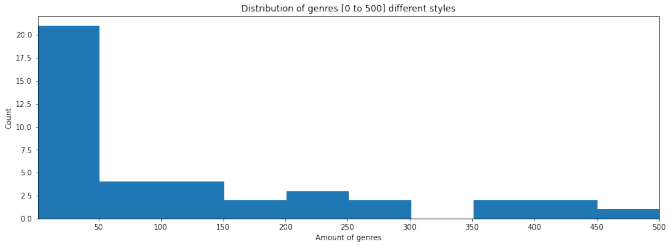

Fig. 2. Genre Distribution of the Wikiart dataset

concluded into

Is it possible to cluster the posters and find similar posters given a new image?

III. Methods

Since the research question binds not to a particular approach, I used four different approaches to find similar posters given a poster. The first method aimed to perform classification with Transfer Learning; the other three approaches perform unsupervised learning and identify an unknown number of clusters. Since the problem statements point out different pinpoints, different methods need to be tested to find the best working one. Each of these approaches had its pros and cons, but most of them required a lot of computing power. For this project, I used an Nvidia RTX 3080 Ti.

A. Style Classification

For this approach, I used data from WikiArt [1] to train a Transfer Learning neural network. Since the poster dataset is not labeled, it seems like a good approach to perform Transfer Learning. This technique allows using a trained network from a similar domain to predict the labels it was trained on. A survey of Transfer Learning points out that for my specific problem, this could be a good approach if another dataset from a similar domain is used [Zhuang et al.,].

Wikiart is an open platform that provides images of famous painters to download. The pictures from WikiArt show the ancient painting, but there is also an extensive description of the year, style, and genre provided. I used the tool from lucasdavid [2] to download the data from WikiArt. In total, I downloaded 17.2 Gigabytes of WikiArt images. The idea is that these paintings can infer the style of the posters provided by the museum and therefore implicitly cluster the images based on their predicted style. If the neural network would learn how, e.g., *Expressionism* in a painting looks like, it could identify all posters that advertise for expressionism exhibitions. Originally the genres of the WikiArt images were distributed as shown in Figure 2. It can be seen that most images can be clustered into 20 to 30 different genres.

The WikiArtimages all had different sizes, so I had to use different data augmentation methods like

- RandomAffine
- ColorJitter
- RandomHorizontalFlip
- RandomVerticalFlip
- RandomResizeCrop

For the augmentation, I used the tool *Kornia* [3]. Kornia is a modern tool augmentation [Riba et al.,] that can easily be connected to the PyTorch architecture. The most important augmentation method is the RandomResizeCrop, because this allows that images of different sizes can be cropped to a similar size (224, 224) and be fed into the neural network.

The entire deep learning pipeline was built with PyTorch [4] and PyTorch Lightning [5]. Furthermore, I used a pre-trained Effcient-Net V2 and only changed the last layer to adjust the 20 different labels to predict. The Efficient Net V2 [Tan and Le V, 2019] is one of the latest neural network architectures for reasonable classifications. It can be easily implemented from the Github repository from lukemelas [6] by allowing to download a pre-trained model based on PyTorch, and therefore the last layer can be easily changed.

To not wrongly classify, I threw away all images from WikiArt that were not in the most common 20 labels.

B. Clustering on Image Text with LDA

The second approach to cluster the images is based on a completely different technique. Since the posters mainly advertise exhibitions, the majority of the images are text present. I extracted the text with easyOCR [7] and clustered the images based on topic modeling. The two most common approaches for Optical Character Recognition (OCR) are easyOCR and TesserOCR. Smelyakov and colleagues have already conducted a comprehensive survey [K. Smelyakov et al., 2021] and let to the decision for using easyOCR for my specific poster problem.

The major problem of this approach was that the posters were in a random position. To solve this issue, I needed to predict the text with easyOCR for every image four times, because of the rotations (0, 90, 180, 270) degrees. Even with the use of my Nvidia GTX 3080 Ti and easyOCR supporting PyTorch GPU, the entire process took 24 hours. Furthermore, the images are sometimes blurry and/or the text is not straight but diagonal and in different fonts.

Since easyOCR also finds funny words even if the poster is wrongly rotated, I needed to write an algorithm that detects the most likely rotation angle. For that, I used a language detection package and counted the number of german and English words found in every rotation and selected the most likely by the most occurrences of actual words.

After extracting all the corrected texts from the posters, I used a basic natural language processing pipeline to remove

[1] www.wikiart.org
[2] https://github.com/lucasdavid/wikiart
[3] https://kornia.readthedocs.io/en/latest/index.html
[4] https://pytorch.org/
[5] https://www.pytorchlightning.ai/
[6] https://github.com/lukemelas/EfficientNet-PyTorch
[7] https://github.com/JaidedAI/EasyOCR

stopwords, numbers and unnecessary characters. Also, standard techniques like lemmatization have been conducted.

For topic modeling, there exist different approaches. For my first approach, I used the well-studied LDA (Latent Dirichlet Allocation) dimensionality reduction method by Ng and colleagues [Blei et al., 2001]. The python package gensim offers an easy-to-use implementation of this method. The drawback is that the number of topics must be given as a parameter to the algorithm. Since I have zero expert knowledge about possible topics, it is hard to estimate the right amount of topics.

C. Clustering on Image Text with BERT

Since I also wanted to use a more state-of-the-art approach for topic modeling, I used the BERT implementation from Google [Devlin et al., 2019]. This is a transformer-based neural network created by Google. The python package BERTopic leverages the transformer BERT and also word embeddings to perform the topic modeling. The package is very straightforward to use. The corpus must be provided as a list predicting the right amount of topics, so not even a fixed number of topics must be defined in advance, other than the LDA-based approach.

The number of clusters can be minimized later, but it is a huge advantage to have a broad idea about the number of topics the algorithm found in comparison to LDA. To evaluate a new image with the found clusters, I predicted to every image the topic. The extracted text from the new image can then be put in one of the clusters.

D. Clustering on Image with K-Means

The third approach is also completely different than the first two. Here I investigated if it is possible to cluster the posters based on their image content. For this, I used a pre-trained VGG network to extract the CNN features out of the posters and then cluster all of the features with the commonly used K-Means algorithm. Li and colleagues have already conducted a comprehensive study [Li and Wu, 2012] that shows that K-Means is the right choice for my problem statement.

Extracting the features from nearly 17.000 images also takes much time. Furthermore, the drawback is again that K-Means requires the number of clusters it shall find as a parameter.

The cluster of a new image can be found by extracting the features of the new image with the same VGG model and then calculating the distances to all centroids of the K-Means, and then selecting the centroid with the closest position.

The steps for performing this task are the following:

- Extract features with the VGG net individually
- Perform PCA on all features
- Perform K-Means with fixed-size number of clusters
- Predict new images by beginning with point one again

IV. Results

In summary, the different approaches performed differently good. From my point of view, the best performing method is the BERT topic modeling. It found the right amount of clusters on its own and it also produces excellent visualization to get a glimpse of how the clusters are distributed. For my Streamlit Web App [8], I therefore used the BERT topic modeling model and also, for comparison purposes, the unsupervised image clustering approach with K-Means.

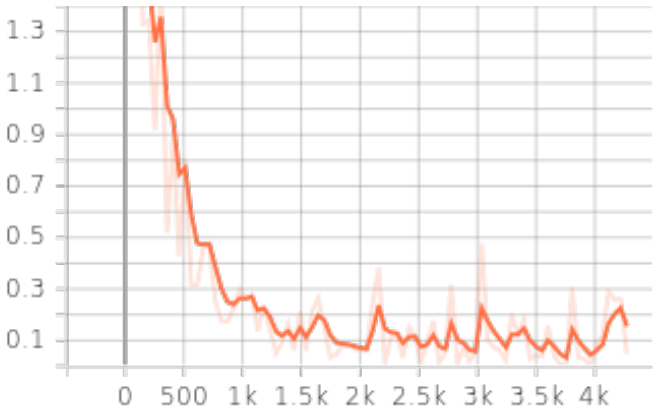

Fig. 3. Tensorboard train loss metric

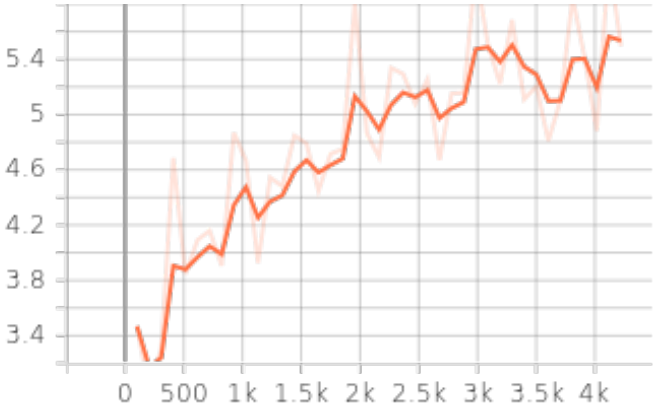

Fig. 4. Tensorboard validation loss metric

TABLE I
Comparison of the four approaches

Approach	Rank (performance)
Classification	4th
BERT Topic Modeling	1st
LDA Topic Modeling	3rd
KMeans Image Clustering	2nd

A. Style Classification

The results of the classification are inferior. The number of genres needed to be limited, due to the stratification of the dataset into train, test and validation set. In Figure 3 and Figure 4 we can clearly see that the model is overfitting towards the training set and the validation loss is increasing again.

The model overfits even though the following tests have been accomplished:

- Different number of labels

[8] https://streamlit.io/

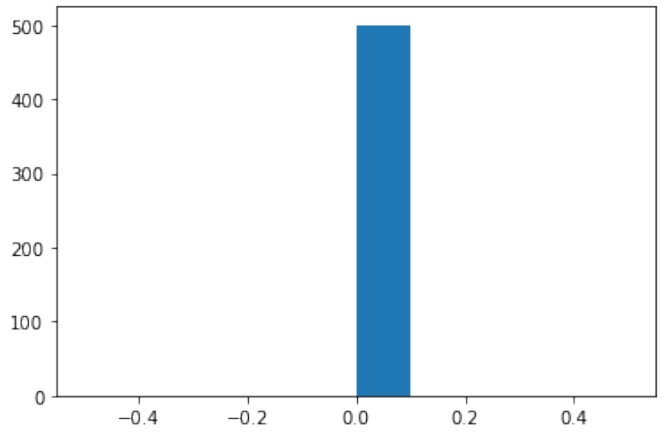

Fig. 5. Model output after training

- Different stratifications of the train, test and validation set
- Many different augmentation techniques
- Different sizes of the Random Crops

All of this was only for training a model on the WikiArt dataset. Apparently, the images in this dataset are not generalizable on a single genre. We can see in Figure 5 that the model only predicts for one genre.

B. Clustering on Image Text with LDA

Focusing on the text of the posters appeared to work significantly better. Although we have no ground truth and therefore cannot compute any metrics about the clustering quality.

We do not know how the LDA clustered the text, but at least we have some examples compared to the classification that did not work. The clusters can be seen in Figure 6.

C. Clustering on Image Text with BERT

The BERT transformer from Google was also implemented into a topic clustering model. This approach appears to be the best working one. After extracting the text from all posters, the most common words are shown in the following Wordcloud in Figure 7.

Based on these words, the BERT transformer was applied. In the visualization in Figure 8, we can see that many clusters were found apart from each other.

A random cluster contains the following words:

- yellow
- white
- theme

Another random cluster contains the following words:

- Autorenlesung
- Bibliotheken
- Improtheater

We can clearly see connections of the words even without having the same word stem.

Fig. 6. Visualized Clusters of the LDA approach

Fig. 7. Wordcloud of the most common words of the poster data

D. Clustering on Image with K-Means

The unsupervised clustering approach with PCA and K-Means appears to perform second-best among the four approaches. In Figure 9 we can detect that the clusters show somehow related images. The text of the images was not used for this approach, only the VGG network to calculate CNN features out of the image. Cluster 3 for example, appears to cluster images with much white content, whereas cluster one clusters complex artworks together.

The training of this approach is relatively quick, but many steps were required.

Fig. 8. Visualized Clusters of the BERT topic modeling approach

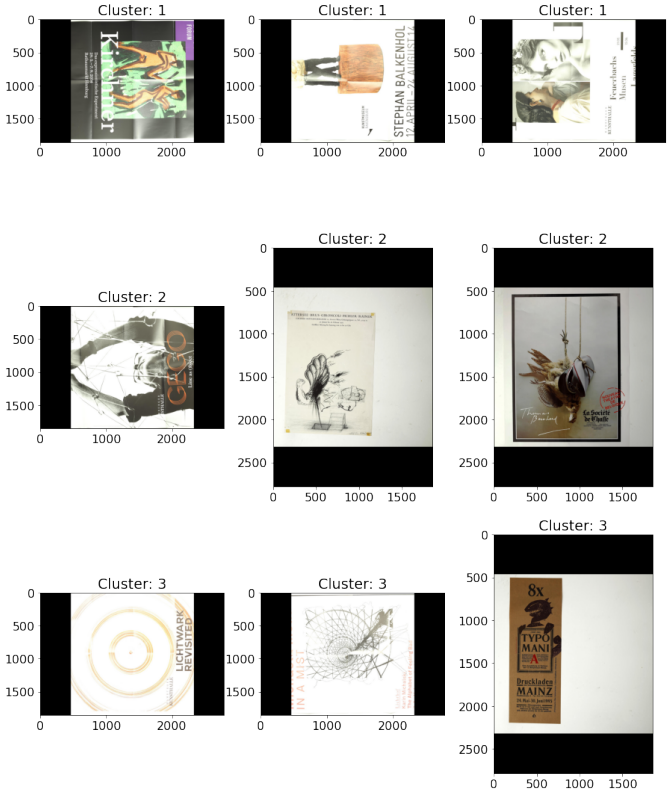

Fig. 9. Output of the Image Clustering model for three different clusters with three images each

E. Discussion

None of the methods was satisfying enough to predict effectively the number of clusters or similar posters. The research question can therefore be answered with no, but there exist many more approaches to try. Since we are dealing with poster images, the necessity of the text is not neglectable, but also the use of the image content. Furthermore, since there is no ground truth, at least for the three unsupervised approaches, we need to define metrics or let humans decide if the predicted clusters make sense for some samples.

F. Future Work

Many things could be considered to improve current models:

- Use a better OCR method
- Try out different parameters like the minimum number of words per poster
- Use other NLP cleaning techniques for the text
- Perform Named Entity Recognition (NER) to extent the word list of each poster
- Impute values if a poster has no words at all
- Combine Image Clustering and Topic Models

I assume that the most impact will have the Named Entity Recognition and the image clustering and topic modeling combination. Since this was a time constraint University IT-Project, I had no more time to investigate these approaches further.

REFERENCES

[Blei et al., 2001] Blei, D., Ng, A., and Jordan, M. (2001). Latent Dirichlet Allocation. volume 3, pages 601–608.

[Devlin et al., 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[K. Smelyakov et al., 2021] K. Smelyakov, A. Chupryna, Dmytro Darahan, and Serhii Midina (2021). Effectiveness of Modern Text Recognition Solutions and Tools for Common Data Sources. In *COLINS*.

[Li and Wu, 2012] Li, Y. and Wu, H. (2012). A Clustering Method Based on K-Means Algorithm. *Physics Procedia*, 25:1104–1109.

[Riba et al.,] Riba, E., Mishkin, D., Ponsa, D., Rublee, E., and Bradski, G. Kornia: an Open Source Differentiable Computer Vision Library for PyTorch.

[Tan and Le V, 2019] Tan, M. and Le V, Q. (2019). EfficientNetV2: Smaller Models and Faster Training. *International Conference on Machine Learning*.

[Zhuang et al.,] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. A Comprehensive Survey on Transfer Learning.