Learning to rank approach for refining image retrieval in visual arts

Tetiana Yemelianenko, Iuliia Tkachenko, Tess Masclef, Mihaela Scuturici, Serge Miguet

Learning to rank approach for refining image retrieval in visual arts

Tetiana Yemelianenko, Iuliia Tkachenko, Tess Masclef, Mihaela Scuturici, Serge Miguet
Univ Lyon, Univ Lyon 2, CNRS, INSA Lyon, UCBL, LIRIS, UMR5205
F-69676 Bron, France
{tetiana.yemelianenko, iuliia.tkachenko, tess.masclef, mihaela.scuturici, serge.miguet}@univ-lyon2.fr

Abstract

Modern content-based image retrieval systems demonstrate rather good performance in identifying visually similar artworks. However, this task becomes more challenging when art history specialists aim to refine the list of similar artworks based on their criteria, thus we need to train the model to reproduce this refinement. In this paper, we propose an approach for improving the list of similar paintings according to specific simulated criteria. By this approach, we retrieve paintings similar to a request image using ResNet50 model and ANNOY algorithm. Then, we simulate re-ranking based on the two criteria, and use the re-ranked lists for training LambdaMART model. Finally, we demonstrate that the trained model reproduces the re-ranking for the query painting by the specific criteria. We plan to use the proposed approach for reproducing re-rankings made by art history specialists, when this data will be collected.

1. Introduction

In recent years, a large amount of visual art collections has become available thanks to the digitalization of the big museums' and art galleries' collections. Among them are collections from the Metropolitan Museum of Art, which consists of more than 490k images from more than 224k classes, Images D'Art with around 500k works from French museums, the National Gallery of Art [19] with collection of more than 150k paintings, sculpture, decorative arts, photographs, prints, drawings and others. A list of existing artwork datasets can be found in the review of She and Cetinic [28] with comments about the size of collections and specific tasks for which datasets were designed or most often used. The large number of digitized artworks let researchers implement for analyzing these collections approaches based on Deep Neural Networks (DNN), which are commonly used in computer vision tasks. Art history specialists could be very interested in automated tools for searching connections between different artworks because a manual approach could be very time-consuming, and also sometimes using automated search lets find hidden connections in big art painting collections. A variety of approaches was proposed last years, a lot of them use deep learning models pre-trained on big photographic datasets for feature extraction and then use these features in art paintings classification or art retrieval tasks. The problem is that these models are trained to search for generalized visual similarity of the paintings, but very often experts can have their own criteria of similarity and their opinions can differ among different groups of specialists, such as art history specialists and semioticians for example. According to this, the same list of objects proposed by art retrieval system can be evaluated as very relevant to one group of specialists and less relevant to another. It is interesting to find a way to take into account the experts' criteria, which can improve art retrieval results according to these criteria. This task is rather common for information retrieval problem, and details of using deep learning for content-based image retrieval tasks can be found in the survey of Wei Chen *et al*. [5].

The main idea of our work is to find a way to refine the list of retrieved similar paintings according to the specific criteria of art history specialists or regular museum visitors. Possibly, this approach can reduce the so-called "intention gap" between the users' desired results and the actual retrieval outcomes. When the criteria are rather obvious, such as texture similarity, for example, the re-ranking task can be accomplished by the machine based on the calculated characteristics of each painting. However, when an expert has specific preferences or wishes to retrieve paintings based on more complex criteria, such as composition similarity for example, the automatic re-ranking possibilities are limited because such criteria cannot be easily formalized for automated tools. Nonetheless, we can train a model to mimic expert behaviour through a supervised learning approach, utilizing a training dataset consisting of previously collected expert re-rankings. In this regard, we propose to train the Learning To Rank (LTR) model based on a simulated ex-

pert criterion training dataset. To the best of our knowledge, our work is the first attempt at using LTR models in the art retrieval task.

This work is a part of a interdisciplinary project that involves the participation of specialists in computer vision and art history.

The paper is organized as follows. Section 2 deals with related works. The proposed approach is presented in Section 3. Section 4 describes the datasets used, shows the experimental setup, and the results obtained. Section 5 concludes the paper and outlines directions for further research on the topic.

2. Related works

There are two lines of previous studies related to our work: the research on the study of art retrieval approaches and LTR models.

2.1. Art retrieval

Fine art retrieval task can be formulated as the problem of finding list of images of fine art objects from art dataset which are similar to the query image. Paintings can be considered as similar not only in terms of visual similarity but also can be stylistically or semantically similar. To obtain a representation of the image in latent space, we need to extract the features of the image. One of the most popular approaches is using pre-trained DNN for feature extraction, among these networks, the best-performing are DenseNet [14] and ResNet50 [13].

One of the first works in which DNN were used for artworks retrieval was introduced by Crowley and Zisserman [6], authors proposed utilize object classifiers learned using Convolutional Neural Networks (CNN) features from natural images to retrieve paintings containing searched object. Feature vectors were generated using CNN, and then classifiers were trained with a Linear-SVM. Later the variety of approaches were proposed which were focused on visual [4, 27, 29] and content similarity [18], visual recognition of a style [11, 7, 32], and other criteria, among them, composition similarity [20, 21] and the pose similarity [30, 16, 26].

Last years, there has been a growing interest in studying multi-modal retrieval approaches. Garcia and Vogiatzis [10] addressed semantic art understanding as a multi-model retrieval task, where relevant images (texts) are retrieved based on the input artistic text (or image). Garcia *et al*. [9] extracted visual features using fine-tuned ResNet50 network, these visual features were enhanced with contextual data. The authors proposed two models which were evaluated in classification and art retrieval tasks. Yankun *et al*. [34] used approach based on generative model for multi-modal retrieval. Their method relies on generating synthetic images generated with Stable Diffusion and computing CLIP [24] image embeddings to obtain content and style embeddings of paintings in content and style spaces. Efthymiou *et al*. [8] proposed the multimodal architecture, which consists of Graph Neural Networks and Convolutional Neural Networks. These networks were jointly trained on visual and semantic artistic representations. Yang *et al*. [36] proposed adaptive multi-task learning method that weights multiple loss functions based on Lagrange multiplier strategy. The authors simultaneously learned multiple objectives and evaluated their model on art classification and art retrieval problems.

All of these approaches either focus on generalized artworks retrieval based on features extracted using DNN or on more specific criteria, such as style, pose, composition similarity, or their combination. For the first group, it is not possible to consider specific preferences of experts. For the second group, it is possible to choose desired criteria for art retrieval, but these criteria must be fixed in advance, and the model needs to be trained accordingly. In such cases, adding or changing criteria without retraining the model is not feasible. Retraining the model can be time-consuming due to the requirement of large datasets when using these models.

Our approach differs because we train the DNN on a large artistic dataset only once, and then we aim to train LTR models to learn specific experts' criteria using relatively small datasets constructed based on the experts' relevance evaluation of previously selected similar paintings. In this paper, we did not use experts' re-ranking, instead, for training we used datasets which are constructed on simulated re-rankings using two selected criteria.

2.2. Learning-to-rank problem

LTR is a class of machine learning algorithms, typically supervised, that is applied to solve the ranking problem in information retrieval (IR) task. LTR is a technique used to re-rank the highest N obtained objects (documents or images) by utilizing trained machine learning models. To perform LTR, a dataset with training data should consist of queries, lists of found similar objects, and the relevance scores of these objects. LTR is commonly employed to improve retrieved results based on user preferences. The LTR task differs from classification and regression tasks. In a classification problem, we aim to predict labels of objects, while in a regression problem we aim to predict a continuous value based on the input variables. However, in the LTR problem the goal is to sort the list of objects according to their relevance to the query object such as the most relevant objects are in the top of the list. Therefore, in the LTR problem the primary focus lies in the relevance order of the objects rather than their individual scores of similarities to the query object.

According to [31] the LTR task can be formally defined in the following way. The training data contains a set of queries Q, for a single query $q \in Q$ we have a list of similar objects $S^q = \{s_1^q, s_2^q, \ldots, s_N^q\}$, each object of S^q is presented as a vector of features. And for the query q we have a list of relevance scores $R^q = \{r_1^q, r_2^q, \ldots, r_N^q\}$, where r_i^q is the relevance score for s_i^q. Thus the training data is represented as

$$T = \{(S^q, R^q) \mid q \in Q\}.$$

The goal is to train the ranking model $f(S^q)$ to predict for a query q the relevance scores for similar objects. Further, we omit superscript q for conciseness.

The existing LTR approaches can be divided in 3 main groups according to used loss function: the pointwise, pairwise and listwise approaches.

In the pointwise approach we evaluate the similarity of query and similar objects one object at a time. The methods in this group train classification or regression models to predict the relevance score of similarity for each individual object compared to the query object. Classification methods are trained to define for the similar object the class to which this object belongs, for example "very similar", "similar", "dissimilar". Regression methods are trained to predict similar function values for similar objects. The model learns a function that for given query-object produces the relevance score for the pair of query-similar object. The loss function measures the accuracy of the prediction for each single object compared to the ground truth label. The final ranking is obtained by sorting the result list based on the scores of similar objects. For the pointwise approach, the score for each object is independent of the other objects, and as the result the position of an object in final ranking is invisible to its loss function. For the pointwise LTR approach, all standard classification and regression algorithms can be used.

In the pairwise approach, a pair of objects is evaluated at a time. The algorithms of this group aim to determine the optimal order for each pair. The loss function only considers the relative order between two objects, and the algorithms attempt to minimize the number of cases where the pair of results are in the wrong order compared to the ground truth. Therefore, the loss function minimizes the number of inversions in the ranking. In practice, pairwise approaches usually outperform pointwise approaches.

In the listwise approach, the entire list of all ranked objects is considered, and a listwise loss is computed. The ranked lists of objects are treated as instances, and a ranking function is trained through the minimization of a listwise loss function defined on the predicted list and the ground truth list [35]. Algorithms in this group are more complex and computationally costly than pointwise and pairwise approaches, but the listwise approach allows for solving ranking problems in a more natural way. Generally, the listwise approach outperforms pointwise and pairwise approaches [3]. Fig. 1 illustrates the three different approaches that can be used for art paintings re-ranking.

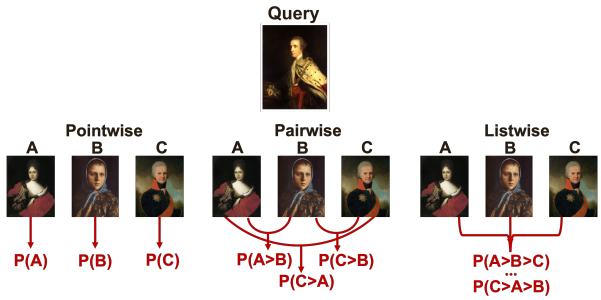

Figure 1. Existing LTR approaches: pointwise, pairwise, listwise.

The commonly used measures for evaluation of trained LTR models are Mean Average Precision (MAP), Normalized Discounted Cumulative Gain (NDCG), and Mean Reciprocal Rank (MRR). MAP considers the number of relevant objects in the ordered list, making it useful in tasks with binary relevance, where objects are considered either relevant (score 1) or irrelevant (score 0), NDCG assigns higher weights to objects at the top of the ordered list compared to those at the bottom. MRR only takes into account the position of the first relevant object. For our purpose, the most useful measure is NDCG since we want to evaluate how well the model learned to re-rank the given list of similar objects according to expert's criteria. The value of NDCG measure is calculated as follows:

$$NDCG@n = \frac{DCG@n}{maxDCG@n},$$

where $DCG@n = \sum_{i=1}^{n} \frac{2^{rel_i}-1}{log_2(i+1)}$, rel_i — is the relevance score of object number i in the ordered list, $maxDCG@n$ — DCG of the list with the objects ranked in the most relevant order, the notation $NDCG@n$ means that only n top objects are taken in consideration.

Fig. 2 illustrates the calculated values of the NDCG measure for two different rankings. The ground truth ranking is the expert's ranking, and the most relevant object has the relevance score 2 in this ranking. The paintings in the rankings 1 and 2 are sorted by their relevance to the query painting according to each of these rankings. The value on the right side of the painting indicates the relevance score of this painting in the ground truth ranking. The NDCG measure lets define which of two rankings is the most similar to the ground truth expert's ranking.

Most of the recently proposed LTR algorithms are based on using neural networks, but despite the impressive performance of neural network models in a variety of other machine learning tasks, such as computer vision and natural language processing, their effectiveness in traditional LTR problems has yet to gain widespread recognition. Qin *et al.* [23] showed that gradient-boosted decision trees out-

Figure 2. Calculated NDCG metrics for two rankings, expert's ranking is the ranking with objects ranked in the most relevant order according to the expert's criterion, 2, 1, 0 — the relevance scores rel_i in the ground truth ranking.

perform other approaches for the LTR problem. According to [23] a classic LambdaMART algorithm [33] implemented within a LightGBM[1] library [17] outperformed recent neural ranking algorithms and a version of LambdaMART implemented in a RankLib[2] library, by a large margin. In a recent paper Osman and Eman [15] showed that their neural ranking algorithm outperforms LambdaMART, but it was not mentioned with what implementation of LamdaMART LightGBM or a weaker RankLib it was compared. A comprehensive survey of neural ranking models can be found in [12]. Anyway, LambdaMART still stays one of the state-of-the-art algorithms in LTR tasks, especially in tasks with small datasets and hand-crafted features. Models based on neural networks require a big amount of training data, which is not possible in our case, because only a small number of experts will participate to the re-ranking task, so among all the models LambdaMART was chosen, because we need a model which will not overfit on a small dataset.

LambdaMART is the boosted tree version of LambdaRank, which is based on RankNet. Typically, the RankNet algorithm is based on neural networks, but it is possible to use any underlying model for which the output of the model is a differentiable function of the model parameters [2].

LambdaMART combines Multiple Additive Regression Trees (MART) and LambdaRank and this algorithm is based on a gradient boosting for combining an ensemble of weak prediction models into a single prediction. On each training iteration of gradient boosting a cost function derived from LambdaRank is performed.

[1]https://github.com/microsoft/LightGBM
[2]https://sourceforge.net/p/lemur/wiki/RankLib

3. Methodological approach

3.1. Proposed pipeline to re-ranking

LTR is a supervised approach, so we need experts' re-rankings to train the chosen LTR model. Expert annotations are expensive and time-consuming, and at this stage of the project, it was decided to verify whether the proposed approach works and can be used when we collect re-rankings made by experts. Therefore, in this paper, we propose to fully simulate expert interactions with retrieved lists of similar paintings using two different criteria: colour palette similarity and similarity according to objects found in the paintings.

To train the LTR model, we constructed the dataset in the following way. For every query painting, we found a list of $N = 15$ similar paintings (N was chosen empirically), then re-ranked this list based on the selected criterion. Objects in the re-ranked list were ordered according to their relevance scores. For each similar painting, the relevance score was calculated based on the distance from the query painting - the smaller the distance, the higher the relevance. We chose the relevance scores in the interval [1, 15], where 15 is the score for the most relevant painting. Next, we used re-ranked lists to train the LTR model using the LambdaMART algorithm and verified how well this model reproduced re-ranking Fig. 3.

In our work, we used ResNet50 for feature extraction of the paintings and Approximate Nearest Neighbour Oh Yeah (ANNOY) [3] algorithm for searching similar paintings. Previously, the ResNet50 was fine-tuned for genre classification, the length of the feature vector is 512. Fig. 4 illustrates the proposed pipeline. First, for query painting q the feature vector is calculated using a pre-trained ResNet50 DNN model. Then, $N = 15$ paintings which are the most similar to the query painting are selected using ANNOY algorithm. Next, using the pre-trained LTR model, the list of similar paintings is re-ranked based on the learned simulated expert's re-ranking.

3.2. Simulated interactions

The first tested simulation criterion was palette similarity. For the extraction of colours from the art paintings we used the ExtColors[4] tool which groups colours based on visual similarities using the CIE76 formula, and for each colour in the palette presented in RGB format the proportion of this colour is calculated. Distances between palettes were calculated by using the minimum colour difference model proposed by Qianqian and Stephen [22]. Additionally, we made a correction based on the proportion of each colour in the compared palettes. For each query painting, we retrieved $N = 15$ most similar paintings using the art

[3]https://github.com/spotify/annoy
[4]https://github.com/CairX/extract-colors-py

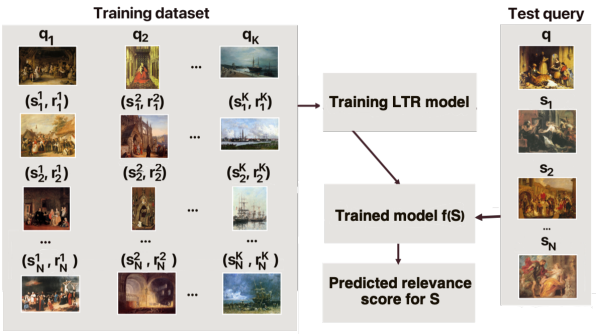

Figure 3. Using of trained LTR model for re-ranking the list of chosen paintings according to the expert's criterion.

retrieval tool, then re-ranked this list according to the distances between palettes. Fig. 5 illustrates calculated palettes and distance between them.

We also tested the simulated criterion which allows to re-rank paintings based on the objects drawn in the paintings. The re-ranking simulation was made in the same way as for the palettes. First, we retrieved $N = 15$ most similar paintings for the query painting, then we defined objects painted in the query painting and each retrieved similar paintings using annotated data from the dataset. Then, we calculated the distances between the query painting and similar paintings considering which objects each pair query-similar painting has in common.

4. Experiments

4.1. Datasets

For the first simulated criterion we used the WikiArt dataset, which is one of the largest online collections of digitized paintings available. The dataset covers the periods between the 15th and 20th centuries. WikiArt contains 81,444 paintings and integrates metadata including 27 different styles (Romanticism, Baroque, Impressionism, etc.), 10 painting genres (landscape, portrait, still life, etc.) and artist names.

The WikiArt dataset was used as the base for the simulated re-ranking criterion for the palette similarity. We randomly chose 2000 paintings as the query paintings from the WikiArt dataset (the number of queries was chosen empirically), for each selected painting we retrieved $N = 15$ similar paintings using the ResNet50 model and ANNOY nearest-neighbours search algorithm. Then for the query and similar paintings, we calculated palettes of 20 colours and re-ranked the list of similar paintings based on the palettes similarity as described in Section 3.2.

For the second simulation of re-ranking, we used the DeArt dataset: Dataset of European Art proposed by Reshetnikov *et al.* [25]. This dataset contains more than 15000 images of paintings between the XIIth and the XVIIIth centuries. Images are manually annotated with bounding boxes identifying 69 classes, more than 50 classes are specific to cultural heritage, among them are classes which reflect imaginary beings, symbolic entities and other categories related to art. This dataset also contains 12 possible poses for boxes identifying human-like objects. We used this dataset only for the simulation of re-ranking the paintings according to their similarity based on the objects drawn on the paintings. So, in this work, we didn't use pose detection and we were not interested in the coordinates of detected objects.

4.2. Training setup used

We used LightGBM implementation of the LambdaMART algorithm, the hyperparameters were fine-tuned using the Optuna framework [1]. For using the LambdaMART algorithm in LightGBM a parameter 'objective' is defined as 'lambdarank', and LambdaMART is the boosted tree version of LambdaRank, a parameter 'boosting_type' is defined as 'gbdt'. As optimization metric we chose NDCG.

According to the official documentation of LightGBM, we chose the set of parameters which was fine-tuned:

- 'n_estimators' parameter controls the number of boosting rounds that will be performed, for LambdaMART this parameter could be considered as the number of trees, more trees are used, more stable is the prediction, but the too big value of this parameter may cause the overfitting. For this parameter, we chose a possible range between 200 and 500.
- 'learning_rate' parameter controls how much each tree contributes to the final prediction. The chosen possible range was between 0.001 and 0.1.
- 'num_leaves' parameter controls the complexity of the tree model. The range was between 2 and 128.
- 'max_depth' parameter is used to limit the tree depth explicitly. We used the range between 1 and 7.
- 'min_data_in_leaf' parameter defines the minimum of data points that must be present in a leaf and could be used to prevent over-fitting in a leaf-wise tree. The possible range was between 1 and 50.

The set of parameters which gave us a maximal NDCG (0.8852) score on the colour palette test data is: 'number of estimators': 368, 'learning_rate': 0.0194, 'num_leaves': 103, 'max_depth': 7, 'min_data_in_leaf': 6.

The set of parameters which gave us a maximal NDCG (0.9552) score on the objects found on the paintings data is: 'number of estimators': 324, 'learning_rate': 0.0593, 'num_leaves': 56, 'max_depth': 7, 'min_data_in_leaf': 22. We used 20% of re-ranked query paintings for the test set

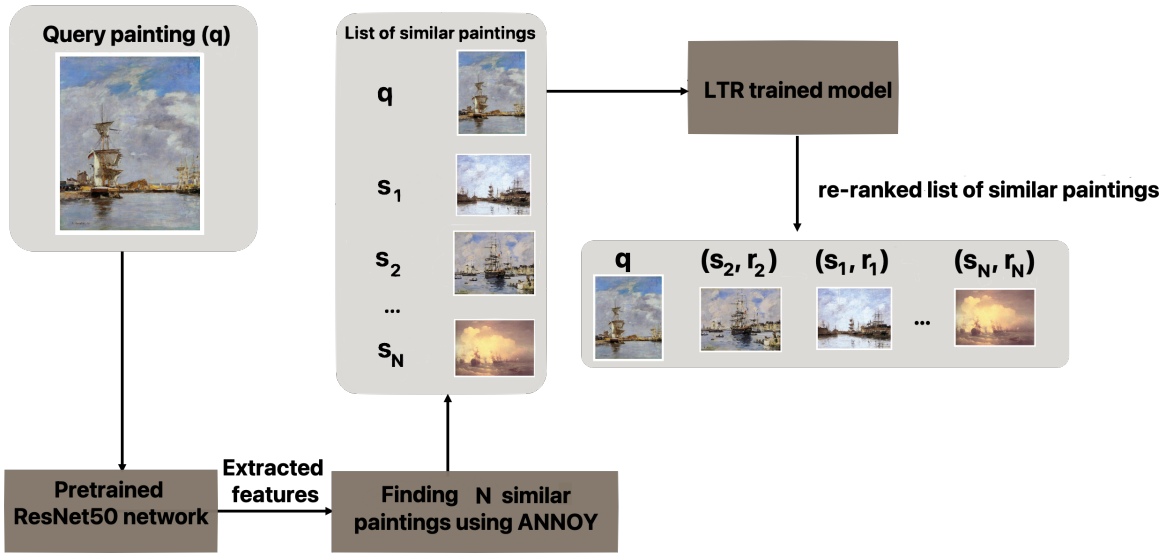

Figure 4. Proposed pipeline to re-rank retrieved similar paintings according to specific criterion, q — query painting, $s_1, s_2, \ldots, s_N$ — N retrieved similar paintings, $r_1, r_2, \ldots, r_N$ — relevance scores of re-ranked similar paintings.

Figure 5. Colour palettes for the query painting (a) and retrieved similar painting (b), here, distance between palettes is 0.89.

and from the rest 80% of re-ranked queries we used 20% for the validation set and all other re-rankings for the train set.

4.3. Results

We evaluated the proposed approach using two simulated re-ranking approaches. The first one uses palette similarity, and the second one the similarity which is based on the objects drawn in the paintings. We repeated each of these two experiments 5 times with random splitting of the dataset on train, test, and validation sets. The performance is evaluated by NDCG measure for relevance scores ranging from 1 to 15, where the relevance score 15 signifies that the painting is the most similar to the query painting according to the expert's re-ranking. The NDCG let us compare the relevance of the paintings retrieved by a search engine (the model based on the ResNet50 network and ANNOY) to the relevance of the painting that would be proposed by an expert (simulated expert criterion in our case).

To illustrate received results, for each painting from the

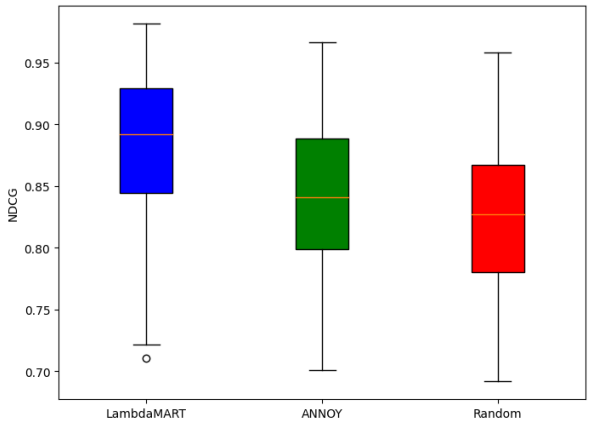

Figure 6. NDCG measure for the initial ranking received by DNN model and ANNOY algorithm (ANNOY in green), random re-ranking (Random in red) and learned re-ranking based on the palette similarity (LambdaMART in blue)

test dataset, we re-ranked the list of retrieved similar paintings using the trained LambdaMART model and calculated the value of the NDCG measure. Then, compared these values of NDCG measure with the calculated values of NDCG for the initially retrieved lists of similar paintings obtained using the ResNet50 and ANNOY algorithm (ANNOY). For comparison purposes, we also included the NDCG measures calculated for each painting from the test dataset with a randomly generated re-ranking of the similar paintings (Random). The results of the comparison are shown in Fig. 6. NDCG measure for the initial ranking obtained by the DNN model and the ANNOY algorithm (ANNOY) is in green, random re-ranking (Random) is in red, and learned

Figure 7. a) initial ranking received by ResNet50 model and ANNOY algorithm; b) paintings re-ranked by learned LambdaMART model. Paintings are ordered according to their relevance scores from the most relevant on the left to the less relevant on the right.

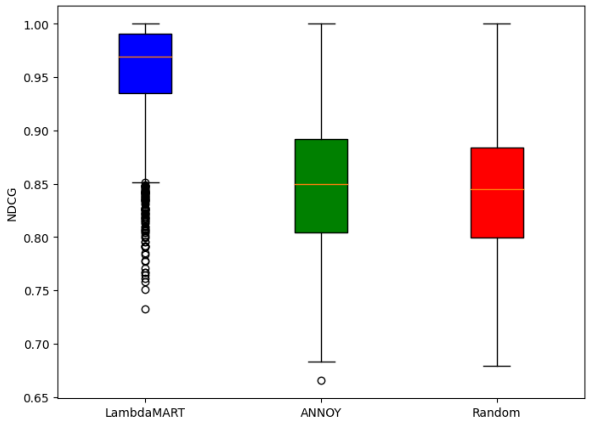

Figure 8. NDCG measure for the initial ranking obtained by the DNN model and the ANNOY algorithm (ANNOY in green), random re-ranking (Random in red) and learned re-ranking based on the type of drawn objects on the paintings similarity (LambdaMART in blue).

re-ranking (LambdaMART) is in blue. The results of the first experiment are introduced in Table 1. First, we calculated NDCG@n values for all re-rankings in test dataset, next, averaged it by the quantity of re-rankings. Then, we averaged these NDCG@n values for all of 5 experiments. For all values of n trained LambdaMART model lets better reproduce learned re-ranking by simulated criterion.

Fig. 7 illustrates the example of comparison of ground truth ranking based on the palette similarity with an initial list of similar paintings retrieved by the combination of the ResNet50 model and ANNOY and the list of similar paintings re-ranked by the learned LambdaMART model. It's worth mentioning, that we calculated the palette similarity only for the simulation of experts' interaction with the retrieved list of similar paintings, and only for creating the dataset which was used for the training LTR model. The value of the NDCG measure received for the LamdaMART model equals 0.9584, and for the initial list equals 0.8974.

For the second series of experiments, we used simulated re-ranking based on the objects drawn in the painting

Ranking	NDCG@15	NDCG@5	NDCG@1
ANNOY	0.8377	0.6213	0.5694
LambdaMART	**0.8848**	**0.727**	**0.6899**
Random	0.8328	0.5975	0.5428

Table 1. Comparison of NDCG measure for the initial ranking (ANNOY), random re-ranking (Random) and learned re-ranking based on the objects drawn on the paintings similarity (LambdaMART)

similarity. We did the comparison for this criterion in the same way as for the first simulation. In Fig. 8 we compared the calculated values of NDCG for the original retrieved list (ANNOY in green), for the list of paintings with the randomly chosen re-ranking (Random in red) and for the re-ranking made by using a trained LTR model (LambdaMART in blue). The calculated NDCG@n values are introduced in Table 2. In this case, the difference between original ranking and learned re-ranking is even more important than for the previous case. A possible explanation is that colour palettes similarity is a low-level criterion, that might already be captured in the latent space of ResNet50, whereas, object similarity involves a higher-level (semantic) interpretation of the painting, more difficult to learn with convolution layer only. Anyway, these two series of experiments let us conclude that there is a possibility to learn the ranking which can be made by an art history specialist or another user for improving the list of retrieved paintings according to this specific learned criterion.

It is worth mentioning, that during data collection for training, all art history specialists must use the same criterion for re-ranking. For different criteria, LTR models must be trained separately for each criterion. A dataset of 2000 query paintings was used to train the LTR model for the palette similarity criterion. This implies that if data were collected from a group of 20 art history specialists, each specialist would need to provide re-ranking for 100 query paintings. With a smaller number of available experts, the dataset size could be reduced, but this would subsequently lead to a reduction in the NDCG measure as well. In the

Ranking	NDCG@15	NDCG@5	NDCG@1
ANNOY	0.8473	0.634	0.5941
LambdaMART	**0.9552**	**0.893**	**0.8677**
Random	0.8397	0.6182	0.5642

Table 2. Comparison of NDCG measure for the initial ranking (ANNOY), random re-ranking (Random) and learned re-ranking based on the type of drawn objects on the paintings similarity (LambdaMART)

case of palette similarity, the average NDCG measure for 5 experiments decreased from 0.8865 to 0.865 for 2000 and 400 query paintings respectively.

5. Conclusions and Perspectives

We have proposed an approach to enhance the performance of content-based image retrieval based on a simulated specific criterion. Our experimental results demonstrate the potential of the proposed approach to improve the retrieved results obtained by the DNN model according to specific criteria.

The next step involves validating the effectiveness of the proposed approach not only on simulated re-rankings but also on real users' requirements using one or more criteria. To achieve this, we are developing a web application for collecting specialists' re-rankings considering art history criteria on the proposed similar paintings' relevance scores for each query painting. Subsequently, we will train a LambdaMART model to recreate the art history specialists' re-rankings for unknown query paintings. Over time, we expect the proposed approach to enable the refinement of metrics employed in searching for similarities between paintings. This will allow us to create a tool that incorporates search criteria from art history specialists, making it accessible to the general public.

6. Acknowledgments

This work was funded by french national research agency with grant ANR-20-CE38-0017. We would like to thanks the PAUSE ANR-Program: Ukrainian scientists support to support the scientific stay of T. Yemelianenko in LIRIS laboratory.

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 2623–2631, New York, NY, USA, 2019. Association for Computing Machinery. 5

[2] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent, 01 2005. 4

[3] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: From pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 129–136. Association for Computing Machinery, 2007. 3

[4] Giovanna Castellano, Eufemia Lella, and Gennaro Vessio. Visual link retrieval and knowledge discovery in painting datasets. *Multimedia Tools Appl.*, 80(5):6599–6616, feb 2021. 2

[5] Wei Chen, Yu Liu, Weiping Wang, Erwin M. Bakker, Theodoros Georgiou, Paul W. Fieguth, Li Liu, and Michael S. Lew. Deep image retrieval: A survey. *CoRR*, abs/2101.11282, 2021. 1

[6] Elliot J. Crowley and Andrew Zisserman. In search of art. In *ECCV Workshops*, 2014. 2

[7] Bhargav Desikan, Hajime Shimao, and Helena Miton. Wikiartvectors: Style and color representations of artworks for cultural analysis via information theoretic measures. *Entropy*, 24:1175, 08 2022. 2

[8] Athanasios Efthymiou, Stevan Rudinac, Monika Kackovic, Marcel Worring, and Nachoem Wijnberg. Graph neural networks for knowledge enhanced visual representation of paintings. *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3710–3719, 10 2021. 2

[9] Noa Garcia, Benjamin Renoust, and Yuta Nakashima. Context-aware embeddings for automatic art analysis. *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, 04 2019. 2

[10] Noa Garcia and George Vogiatzis. How to read paintings: Semantic art understanding with multi-modal retrieval. In *Computer Vision – ECCV 2018 Workshops: Munich, Germany, September 8-14, 2018, Proceedings, Part II*, page 676–691, Berlin, Heidelberg, 2019. Springer-Verlag. 2

[11] Eren Gultepe, Thomas Conturo, and Masoud Makrehchi. Predicting and grouping digitized paintings by style using unsupervised feature learning. *Journal of Cultural Heritage*, 31, 12 2017. 2

[12] Jiafeng Guo, Yixing Fan, Liang Pang, Liu Yang, Qingyao Ai, Hamed Zamani, Chen Wu, W. Bruce Croft, and Xueqi Cheng. A deep look into neural ranking models for information retrieval. *CoRR*, abs/1903.06902, 2019. 4

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 2

[14] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Weinberger. Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2016. 2

[15] Osman Ibrahim and Eman Younis. Combining variable neighborhood with gradient ascent for learning to rank problem. *Neural Comput. Appl.*, 35(17):12599–12610, mar 2023. 4

[16] Tomas Jenicek and Ondřej Chum. Linking art through human poses. In *2019 International Conference on Document*

Analysis and Recognition (ICDAR), pages 1338–1345, 2019. 2

[17] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *NIPS*, 2017. 4

[18] Diana Kim, Jason Xu, Ahmed Elgammal, and Marian Mazzone. Computational analysis of content in fine art paintings. In *Proceedings of the 10th International Conference on Computational Creativity, ICCC 2019*, pages 33–40, 2019. 10th International Conference on Computational Creativity, ICCC 2019 ; Conference date: 17-06-2019 Through 21-06-2019. 2

[19] Matthew Lincoln, Golan Levin, Sarah Reiff Conell, and Lingdong Huang. National neighbors: Distant viewing the national gallery of art's collection of collections, 2019. 1

[20] Prathmesh Madhu, Tilman Marquart, Ronak Kosti, Peter Bell, Andreas Maier, and Vincent Christlein. Understanding compositional structures in art historical images using pose and gaze priors. In Adrien Bartoli and Andrea Fusiello, editors, *Computer Vision – ECCV 2020 Workshops*, pages 109–125, Cham, 2020. Springer International Publishing. 2

[21] Prathmesh Madhu, Tilman Marquart, Ronak Kosti, Dirk Suckow, Peter Bell, Andreas Maier, and Vincent Christlein. Icc++: Explainable feature learning for art history using image compositions. *Pattern Recognition*, 136:109153, 2023. 2

[22] Qianqian Pan and Stephen Westland. Comparative evaluation of color differences between color palettes. *Color and Imaging Conference*, 2018:110–115, 11 2018. 4

[23] Zhen Qin, Le Yan, Honglei Zhuang, Yi Tay, Rama Kumar Pasumarthi, Xuanhui Wang, Michael Bendersky, and Marc NajorkL. Are neural rankers still outperformed by gradient boosted decision trees? In *International Conference on Learning Representations (ICLR)*, 2021. 3, 4

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. 2

[25] Artem Reshetnikov, Maria-Cristina Marinescu, and Joaquim More Lopez. Deart: Dataset of european art. In *Computer Vision – ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part I*, page 218–233, Berlin, Heidelberg, 2023. Springer-Verlag. 5

[26] Stefanie Schneider and Ricarda Vollmer. Poses of people in art: A data set for human pose estimation in digital art history, 2023. 2

[27] Benoit Seguin, Isabella diLenardo, and F. Kaplan. Tracking transmission of details in paintings. In *International Conference on Digital Health*, 2017. 2

[28] James She and Eva Cetinic. Understanding and creating art with ai: Review and outlook. *ACM Transactions on Multimedia Computing, Communications and Applications*, 18:1–22, 05 2022. 1

[29] XI Shen, Alexei A. Efros, and Mathieu Aubry. Discovering visual patterns in art collections with spatially-consistent feature learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9270–9279, 2019. 2

[30] Matthias Springstein, Stefanie Schneider, Christian Althaus, and Ralph Ewerth. Semi-supervised human pose estimation in art-historical images. In *Proceedings of the 30th ACM International Conference on Multimedia*, page 1107–1116. Association for Computing Machinery, 2022. 2

[31] Xuanhui Wang, Cheng Li, Nadav Golbandi, Michael Bendersky, and Marc Najork. The lambdaloss framework for ranking metric optimization. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 1313–1322, New York, NY, USA, 2018. Association for Computing Machinery. 3

[32] Na Wei. Research on the algorithm of painting image style feature extraction based on intelligent vision. *Future Generation Computer Systems*, 123, 05 2021. 2

[33] Qiang Wu, Christopher Burges, Krysta Svore, and Jianfeng Gao. Adapting boosting for information retrieval measures. *Inf. Retr.*, 13:254–270, 06 2010. 4

[34] Yankun Wu, Nakashima, Yuta, Garcia, and Noa. Not only generative art: Stable diffusion for content-style disentanglement in art analysis. In *Proceedings of the 2023 ACM International Conference on Multimedia Retrieval*. ACM, jun 2023. 2

[35] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise approach to learning to rank: Theory and algorithm. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 1192–1199. Association for Computing Machinery, 2008. 3

[36] Bing Yang, Xueqin Xiang, Wanzeng Kong, Yong Peng, and Jinliang Yao. Adaptive multi-task learning using lagrange multiplier for automatic art analysis. *Multimedia Tools and Applications*, 81, 01 2022. 2