UMUAI This is a pre-print of paper published at UMUAI journal.
Oficial version available online at: https://doi.org/10.1007/s11257-018-9206-9

Content-Based Artwork Recommendation

Integrating Painting Metadata with Neural and Manually-Engineered Visual Features

Pablo Messina · Vicente Dominguez · Denis Parra · Christoph Trattner · Alvaro Soto

This is a preprint of an article published in UMUAI journal. The final authenticated version is available online at: https://doi.org/10.1007/s11257-018-9206-9

Abstract Recommender Systems help us deal with information overload by suggesting relevant items based on our personal preferences. Although there is a large body of research in areas such as movies or music, artwork recommendation has received comparatively little attention, despite the continuous growth of the artwork market. Most previous research has relied on ratings and metadata, and a few recent works have exploited visual features extracted with deep neural networks (DNN) to recommend digital art. In this work, we contribute to the area of content-based artwork recommendation of physical paintings by studying the impact of the aforementioned features (artwork metadata, neural visual features), but in addition we study manually-engineered visual features, such as naturalness, brightness and contrast. We implement and evaluate our method using transactional data from *UGallery.com*, an online artwork store. Our results show that artwork recommendations based on a hybrid combination of artist preference, curated attributes, deep neural visual features and manually-engineered visual features produce the best results. Moreover, we discuss the trade-off between automatically obtained DNN features and manually-engineered visual features for the purpose of explainability, as well as the impact of user profile size on predictions. Our research informs the development of new generation of content-based artwork recommenders which rely on different types of data, from text to multimedia.

Keywords Artwork; Recommender systems; Content-based Recommender; Hybrid Recommendations; Metadata; Visual Features; Deep Neural Networks

P. Messina
IMFD & Pontificia Universidad Catolica (PUC), Chile
E-mail: pamessina@uc.cl

V. Dominguez, D. Parra, A. Soto
IMFD & Pontificia Universidad Catolica (PUC), Chile

C. Trattner
University of Bergen, Norway

1 Introduction

Despite the financial crisis started in 2008 which shook the markets worldwide, the global artwork market has kept growing over the years. For instance, in 2011, art received $11.57 billion in total global annual revenue, over $2 billion versus 2010 [13]. Particularly, online artwork sales are booming mostly due to the influence of social media and new consumption behaviours of millennials [49]. Online art sales reached $3.27 billion in 2015, and at the current grow rate, it will reach $9.58 billion by 2020. Notably, although many online businesses utilize recommendation systems to boost their revenue, online artwork recommendation has received little attention compared to other areas such as movies [4,15] or music [33,8].

Several stores nowadays sell artworks online, such as UGallery[1], Singulart[2], and Artspace[3]. However, finding the right artwork for people's personal taste is a tricky task, as several different properties need to be considered, apart from the prize range. To overcome these issues, recent research [18] has made first steps to help people to find the content they love more efficiently by using recommender systems.

Recommender systems could indeed help in this task, since previous research have been tailored explicitly towards the artwork domain [5,3,43,18]. Most of these works have dealt with recommendation in museum collections using traditional methods and data such as ratings, textual descriptions and social tags [5, 3,43]. The earliest of these works was the CHIP project [5], which implemented well-known techniques such as content-based and collaborative filtering for artwork recommendation at the Rijksmuseum. More recently, He et al. [18] used pre-trained deep neural networks (DNN), combined with collaborative information, for the recommendation of digital art. This is a very promising technique, since the development of deep neural networks has increased by orders of magnitude the performance on visual tasks such as image classification [26] or scene identification [44]. However, they only studied digital art rather than physical artifacts such as paintings or sculptures, which is what most of the aforementioned online art stores sell.

Unlike these works, in this article we address the problem of artwork recommendation for one-of-a-kind paintings. We call a painting *one-of-a-kind* when only one instance is available. If the only user feedback in the datasets are purchases, then there is no chance for computing user co-occurrences, which is needed for methods such as collaborative filtering. For this reason, we address this problem using a content-based recommender, with a focus on different types of content –including metadata, automatically learned features from deep neural networks (DNN) as well as manually-engineered visual features (MEVF)– and also on how to combine them for personalized recommendation.

Objective. In this paper, we study the impact of different features for online content-based recommender systems of physical artworks. In particular, we investigate the utility of artwork metadata (curated attributes and artist), neural (DNN) and manually engineered (MEVF) visual features extracted from images as well

[1] www.ugallery.com
[2] www.singulart.com
[3] www.artspace.com

as user transactions from the online store *UGallery*[4]. We address the problem of artwork recommendation: (i) with positive-only feedback (user transactions) over *one-of-a-kind* items, i.e., items that go out of stock with the first purchase, (ii) by utilizing item metadata and visual features from images, (iii) by implementing a content-based recommender method that recommends the top-n most relevant artworks to a user, and (iv) by performing an on-line evaluation with expert curators to validate the off-line results obtained with user transactions.

Research Questions. In this article, four questions drive our research, considering the problem of one-of-a-kind artwork recommendation employing content-based methods:

- *RQ1*. What is the performance of the different kinds of features for content-based artwork recommendation when used individually? Since we have several types of features we answer this question by splitting the analysis within two subgroups:
  - *RQ1.1* Which is the best metadata-based feature?
  - *RQ1.2* Which is the best visual feature?
- *RQ2*. How do different sets of features (metadata vs. visual) compare?
- *RQ3*. Is there an optimal way of combining features, by hybrid methods, to maximize recommendation performance?
- *RQ4*. To what extent is an off-line evaluation consistent with an expert user validation?

Contributions. (1) In general, the work outlined in this article makes a contribution to the yet sparsely explored problem of recommending physical artworks to people online. To make this happen, we study and compare the utility of several sources of information (content metadata, visual features), typically available in online galleries. We do this by running an extensive set of simulated experiments with real-world data provided by a large online artwork store based in CA, USA called *UGallery*. (2) Furthermore, our work contributes to the one-of-a-kind recommender system problem – i.e., items that go out of stock with the first purchase – by using a content-based approach. Also (3) we introduce a hybrid artwork recommender which exploits the aforementioned features. Finally, (4) we conduct an evaluation with UGallery curators in order to tell if the off-line results are mirrored when tested on real people. To the best of our knowledge, we believe we are the first to study the utility of pre-trained DNN visual features and how these compare to manually-engineered visual features and metadata for artwork recommendation.

Outline. Section 2 presents a formal definition of the content-based artwork recommendation problem. In Section 3 we survey relevant related work in the area. Section 4 presents the *UGallery* dataset. Then, in Section 5 we provide details of our recommendation methods, following section 6 with our evaluation procedure. Section 7 presents the results, we discuss them in Section 8, and finally section 9 concludes the article and presents ideas for future work.

[4] http://www.UGallery.com/

2 Problem Statement: Content-Based Recommendation of Artworks

Based on the formulation of the recommendation problem by Adomavicious et al. [1], we formalize our content-based recommendation problem with the following definitions.

Let U be the set of all users and I be the set of all items (physical artworks) available in the inventory. Let s be a function which measures the utility of an item i to a user u, $s : U \times I \rightarrow R$, where R is a totally ordered set (e.g., non-negative real numbers within a certain range). In other words, a utility function s which given a user $u \in U$ and an item $i \in I$, returns a predicted utility score r. Now, our end goal is to identify the set R_u of "top k items" $\{i_1..i_k\}$ which maximize the utility of the user u, i.e., the list of recommended items:

$$R_u = \underset{\{i_1..i_k\}}{\operatorname{argmax}} \sum_{j=1}^{k} s(u, i_j) \tag{1}$$

Due to the one-of-a-kind nature of our artwork items, we cannot rely directly on co-occurrence methods such as collaborative filtering. Once an artwork item is purchased, it is immediately removed from the system, and for this reason we formulate our utility function as a content-based recommendation problem. In a content-based recommender, the utility function $s(u, i)$ in [1] is defined as:

$$s(u, i) = score(ContentBasedProfile(u), Content(i)) \tag{2}$$

where $score(x, y)$ usually represents a similarity function (such as cosine or BM25 in the case of documents), and $ContentBasedProfile$ of user u and $Content$ of item i can be respectively represented as vectors, such as TF-IDF vectors using the bag-of-words document model. In our case, $ContentBasedProfile(u)$ will be the set of artworks P_u already purchased by user u. $Content(i)$ is a vector representation of the artwork i, its dimensions can represent different features. In this particular research, these features can be: i) manually curated labels, ii) the artist (artwork's creator), iii) visual features extracted with pre-trained DNNs, e.g. VGG and AlexNet, and iv) manually-engineered visual features, e.g. attractiveness features and local binary patterns (LBP).

In Section 5 we will explain in detail which form the function $score(x, y)$ takes depending on the different features used.

3 Related Work

In this section we provide an overview of relevant related work. The section is split into two parts: *Artwork Recommender Systems* (3.1) and *Visually-aware Recommender Systems* (3.2). Both sub-sections are important to better understand our contribution and the problem we are targeting with the paper. A final section *Differences to Previous Research* (3.3) highlights what we add with our work to the already existing literature in the area.

3.1 Artwork Recommender Systems

Within the topic of artwork recommender systems, one of the first contributions in this area was made by the CHIP Project [5]. The aim of the project was to build a recommendation infrastructure for the Rijksmuseum in the Netherlands. The project used several techniques such as content-based filtering based on metadata provided by experts, as well as collaborative filtering based on users' ratings given to artworks of the Rijksmuseum. Another important contribution in the field is the work developed by Semeraro et al. [43]. In their paper, they introduce an artwork recommender system called FIRSt (Folksonomy-based Item Recommender syStem) which utilizes social tags given by experts and non-experts over 65 paintings of the Vatican picture gallery. They focused their research on making recommendations using textual features (textual painting descriptions and user tags), but did not employ visual features among their methods.

More complex methods were implemented recently by Benouaret et al. [7], who improve the current state-of-the-art in artwork recommender systems using context obtained through a mobile application. The particular research question they address is to what extent it is possible to make museum tour recommendations more useful. Their content-based approach uses ratings applied by the users during the tour and metadata from the artworks people have rated, e.g. title or the artists names. They address the artwork recommendation problem in museums, yet their solution cannot be fully applied to the *one-of-a-kind* problem in online stores as we approach it in this research.

Finally, the recent work of He et al. addresses digital artwork recommendations based on pre-trained deep neural visual features [18]. In this case, the experiments were conducted on a virtual art gallery, with the advantage of items always available and explicit user feedback in the form of ratings.

3.2 Visually-aware Recommender Systems

Manually-engineered visual features extracted from images (texture, sharpness, brightness, etc.) have been used in several tasks for information filtering, such as retrieval [40, 27] and ranking [42]. In the latest years, many works in image processing and computer vision such as object recognition [2], image classification [26] and scene identification [44] have shown significant performance improvements by using visual embeddings pre-trained with deep convolutional neural networks (Deep CNN) such as the AlexNet introduced by Krizhevsky et al. [26] or VGG [45]. These are examples of transfer learning methods, i.e., visual embeddings trained for specific tasks (e.g. image classification) which perform well in other tasks (e.g. image segmentation) and have been adopted for the recommendation problem.

Motivated by these results, MacAuley et al. [35] introduced an image-based recommendation system based on styles and substitutes for clothing using visual embeddings pre-trained on a large-scale dataset obtained from Amazon.com. Recently, He et al. [19] went further in this line of research and introduced a visually-aware matrix factorization approach that incorporates visual signals (from a pre-trained DNN) into predictors of people's opinions. Their training model is based on Bayesian Personalized Ranking (BPR), a model previously introduced by

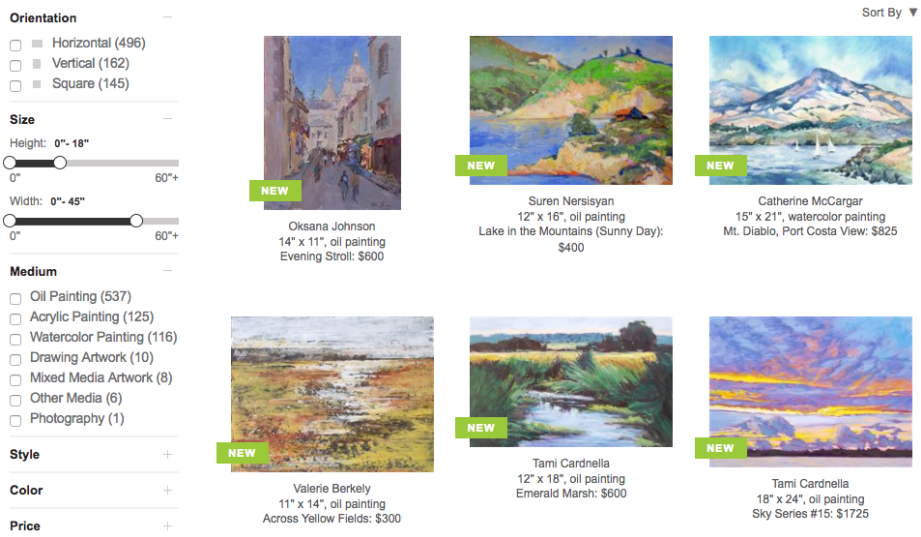

Fig. 1: Screenshot of the search interface of *UGallery*. Users can filter by different facets on the left side.

Rendle [39]. The latest work by He et al. [18] deals with visually-aware artistic recommendation, building a model which combines ratings, social signals and visual features. Another relevant work was the research by Lei et al. [30] who introduced comparative deep learning for hybrid image recommendation. In this work, they use a neural network architecture for making recommendations of images using users' information (such as demographics and social tags) as well as images in pairs (one liked, one disliked) in order to build a ranking model. The approach is interesting, but they work with regular images, not artwork images.

3.3 Differences to Previous Research

Almost all the surveyed articles on artwork recommendation have in common that they used standard techniques such as collaborative filtering and content-based filtering, but without exploiting visual features extracted from images. Unlike these works, we rely exclusively on content-based methods. We are unable to use traditional collaborative filtering, since there are no ratings or implicit feedback on the same item: once an item is purchased, it is out of stock due to its one-of-a-kind condition. In terms of content-based filtering, unlike the previous works we extract, compare and combine metadata, neural visual features and manually-engineered visual features.

Regarding the surveyed works on visually-aware recommendation methods, almost all of them have focused on tasks different from artwork recommendation such as for instance recommending visually similar clothing to people in Amazon.

Only one work, the research by He et al. [18] resembles ours in terms of the topic (artwork recommendation) and the use of visual features. However, there are several important differences: (i) First, although they do use visual features from

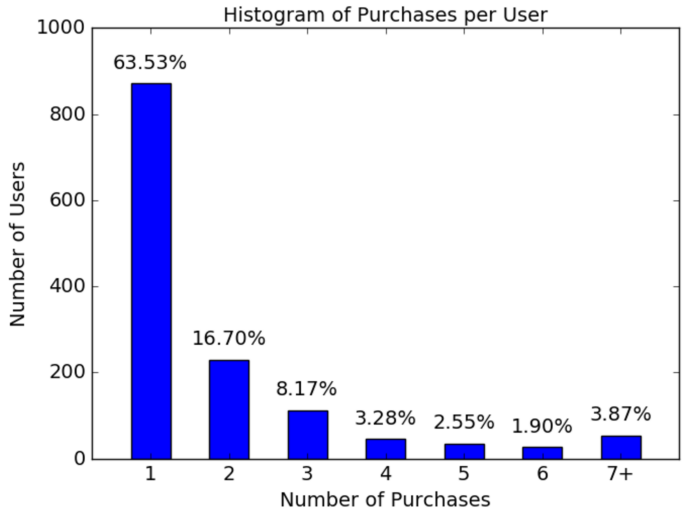

Fig. 2: Distribution of purchases per user. It resembles the typical skewed user consumption behavior in online websites.

DNN embeddings, they do not use manually-engineered visual features, such as brightness or sharpness. (ii) Second, in addition to visual features, we also consider artwork metadata (artwork artists and curated attributes). (iii) Third, our research deals with physical (real-world) artworks, not digital art. Hence, when an artwork is sold, it goes out of stock, whereas in the work of He et al. the digital artworks can be "copied" to an unlimited amount. For us this is a big impediment to using collaborative filtering, which is why our research focuses on content-based recommendation instead. (iv) An fourth, in our work we also perform an on-line evaluation with expert curators to verify consistency with off-line evaluation results.

4 Materials

The online web store *UGallery* has been selling artworks for more than 10 years [49]. They support emergent artists by helping them sell their artworks online. The *UGallery* website allows users (customers) to search for items and to browse the catalog based on different attributes with a predefined order: orientation, size, medium, style and others, as seen on the left side of Figure 1. However, what their current system does not support at the moment is the exploration of items via personalized recommendations, which is exactly what we aim for in this paper.

UGallery provided us with an anonymized dataset of $1,371$ users, $3,490$ items and $2,846$ purchases (transactions) of artistic artifacts, where all users have made at least one transaction. In average, each user has bought 2-3 items in the latest years[5]. Figure 2 shows the distribution of purchases per user. The distribution

[5] Our collaborators at UGallery requested us not to disclose the exact dates when the data was collected.

Table 1: Metadata attributes and attribute values for artworks in the *UGallery* dataset.

Attribute	**Type**	**Values**
Color	Nominal	B&W, Beige, Black, Blue, Brown, Dark Blue, Dark Green, Dark Red, Green, Grey, Orange, Pink, Purple, Red, Turquoise, Violet, White, Yellow
Subject	Nominal	Animals, Architecture, Cuisine, Fantasy, Fashion, Flora, Landscape, Nature, Nudes, People, Religion, Seascape, Sports, Still Life, Travel, Western
Style	Nominal	Abstract, Classical, Expressionism, Impressionism, Minimalism, Modern , Non-representational, Pop, Primitive, Realism, Representational, Street Art, Street Photography, Surrealism, Vintage
Medium	Nominal	Acrylic Painting, Ceramic Artwork, Chalk Drawing, Charcoal Drawing, Colored Pencil, Digital Printmaking, Drawing Artwork, Encaustic Artwork, Gouache Painting, Ink Artwork, Marker Artwork, Mixed Media Artwork, Oil Painting, Other Media, Pastel Artwork, Pencil Drawing, Photography, Printmaking, Sculpture, Watercolor
Energy	Ordinal	Calm, Neutral, Energetic
Seriousness	Ordinal	Playful, Neutral, Serious
Warmness	Ordinal	Warm, Neutral, Cool
Purpose	Ordinal	Decorative, Neutral, Thought-Provoking
Complexity	Ordinal	Simple, Neutral, Complex
Formality	Ordinal	Formal, Neutral, Informal
Age Perception	Ordinal	Young, Neutral, Old

is skewed since most users (871 in total) bought only one item, and only a few users (53 in total) have bought 7 or more items. Our data is not atypical, since it resembles the rating distribution on the Netflix prize or the Movielens dataset, where a few users account for most of the activity and most users have little or none [17,6].

The artworks in the *UGallery* dataset were manually curated by experts. Hence, every artwork has been described with metadata *attributes* such as color, style and medium, to enable the user to filter and browse in the *UGallery* interface. In total, there are eleven attributes, which are described with their respective *attribute values* in Table 1. The attributes in rows 1-4 (*Color* to *Medium*) are self-explainable by reading the examples. Attributes in rows 5-11 (*Energy* to *Age Perception*) are grouped into a meta-category called *Mood*. It is important to note that only from the very latest years onwards the artworks started being filled with all their attributes more systematically. As such, there is a distribution of at-

Table 2: Statistics of attributes' presence among artworks in the *UGallery* dataset.

	Color	**Style**	**Subject**	**Mood**	**Medium**
Present	3,391 (97.16%)	646 (18.51%)	578 (16.56%)	1,550 (44.41%)	3,490 (100%)

Table 3: Symbols used in our artwork recommender approaches.

Symbol	Description
U, I	user set, item set
u, i	a specific user or item (resp.)
P	set of all items purchased in the system up to an arbitrary point in time
P_u	set of all items purchased by user u up to an arbitrary point in time, we refer to these items as the *user profile* or the *user model*, indistinctly
CAV_i^X	set of all curated attribute values of type X present in item i, where X can be either *Color*, *Subject*, *Style*, *Medium*, *Mood* or *All* (all curated attributes at the same time)
a_i	the artist (creator) of item i
V_i	vector of visual features of item i, either manually engineered or obtained with a pre-trained DNN
V_i^X	vector of visual features (of item i) of the specific type X (where X can be e.g. *AlexNet*, *VGG*, *LBP* or *Attractiveness*)

tributes present and absent in the artworks, which is shown in Table 2. While *Color* (97.16%) is present in almost all the artworks, *Subject* is only present in 16,56%. In addition to these curated attributes, the artwork metadata also includes another important source of information: the artwork's artist. In the *UGallery* dataset, each artwork is associated to a unique artist. In total, there are 423 artists, who have 8.25 artworks in average each for sale.

5 Artwork Recommender Approaches

In this section we describe six different content-based artwork recommender approaches, which we have implemented to tackle the one-of-a-kind recommendation problem. Table 3 contains an overview of symbols used in the following subsections.

5.1 Most Popular Curated Attribute Value (MPCAV)

The Most Popular Curated Attribute Value method is the first and most simple approach we tested. Together with Random, it is also used and referred to as a baseline throughout our paper. Since the concept of "popular item" is meaningless in a *one-of-a-kind* setting, instead we recommend based on the most popular *curated attribute values*. Given an artwork i and its corresponding set of curated attribute values CAV_i^X (where X can be either *Color*, *Subject*, *Style*, *Medium*, *Mood* or *All*), we compute its MPCAV score as the sum of the frequencies (popularities) of each of its curated attribute values. More formally, the MPCAV score is calculated as follows:

$$score(i)_{MPCAV} = \sum_{v \in CAV_i^X} \sum_{j \in P} \mathbb{1}(j, v) \tag{3}$$

, where P is the set of products purchased so far, and $\mathbb{1}(j, v)$ is an indicator function which returns 1 if item j has curated attribute value v or 0 otherwise. Intuitively, an item will have a higher score if its curated attribute values are more frequent (popular) among items already purchased in the system. Finally, we rank the items based on this score and recommend the top-n.

Because of the low granularity of the curated attribute values (which at least was the case with the UGallery dataset), one problem of this scoring function is that it may be prone to ties, i.e. many items with the same score. Therefore if there are too many items with the same score that do not fit into the top-n limit, as a workaround we uniformly sample a subset of these items just to fit the top-n recommendation.

5.2 Personalized Most Popular Curated Attribute Value (PMPCAV)

This method is equivalent to MPCAV, with the only difference that we just look at the past purchases of user u instead of the past purchases of the whole system. More formally, the formula for the PMPCAV scoring function is:

$$score(u, i)_{PMPCAV} = \sum_{v \in CAV_i^X} \sum_{j \in P_u} \mathbb{1}(j, v) \tag{4}$$

, which is almost exactly as equation 3, but here we consider only the set of items purchased by the user u, i.e., the set P_u. Then we can rank items and recommend the top-n based on this score. In case of ties, the same workaround as in MPCAV can be used (uniform sampling). On the other hand, if we are not able to build a user model because the user's purchased items lack proper tagging, a possible fallback option is to switch to MPCAV.

A weakness of this method compared to MPCAV is that it requires at least one previous purchase from the user to make recommendations. On the positive side, by considering the user's preferences, one should expect more accurate recommendations.

5.3 Personalized Favorite Artist (FA)

Besides curated attributes, the artwork metadata also includes another important source of information: the artist who created the painting. The FA method leverages this information by recommending artworks created by artists that the user has shown favoritism for. More formally, given a user u and an item i, the FA scoring function is defined as follows:

$$score(u, i)_{FA} = \sum_{j \in P_u} \mathbb{1}(j, a_i) \tag{5}$$

, where $\mathbb{1}(j, a_i)$ is an indicator function that returns 1 if the artist a_i of artwork i is also the creator of artwork j (in our dataset, each artwork is associated to a single creator). Intuitively, an artwork has a higher score if the user has purchased more artworks from the same artist in the past. Then we rank and recommend the top-n artworks based on this score. If there are too many items with the same score, a

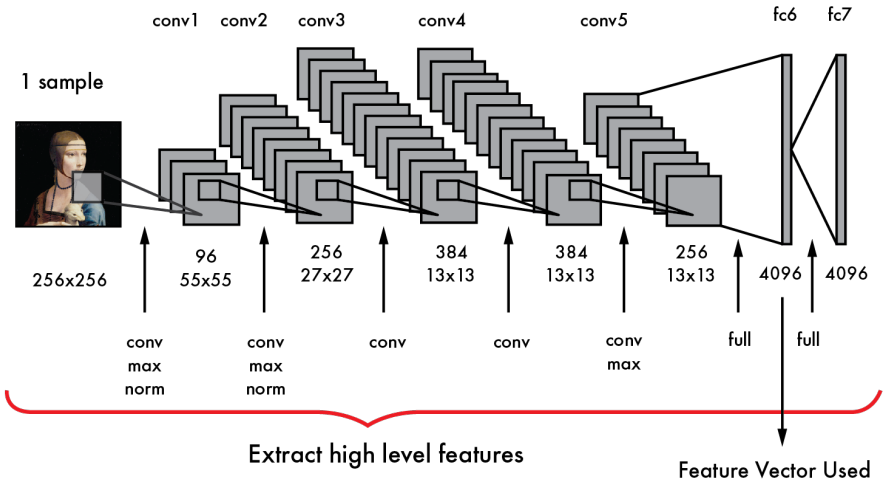

Fig. 3: Alexnet architecture. This shows the process to obtain the latent feature vector we use in our experiments, which corresponds to fc6. A convolutional window passes over the image, from each layer to the next layer, with different shapes and strides in every layer. This figure is inspired by [21].

subset of these items can be uniformly sampled to fit the top-n recommendation. On the other hand, if the are too few items with a positive score to recommend (e.g. because the user's favorite artists have sold almost all their artworks), we resort to the globally most favorite artists to rank the remaining artworks and fill the top-n recommendation.

5.4 Latent Visual Features: Deep Neural Network Embedding (DNN)

Since the dataset contains one image for every item, we tested visual features for artwork recommendation. One of the two visual embeddings used was a vector of features obtained from an AlexNet, a convolutional deep neural network developed to classify images [26]. In particular, we use an AlexNet model pre-trained with the ImageNet dataset [11]. Using the pre-trained weights for every image a vector of 4,096 dimensions was generated with the Caffe[6] framework. As seen in Figure 3, this vector corresponds to the ouput of the first fully connected layer of AlexNet, also known as fc6.

Although there are two fully connected layers (fc6 and fc7) we used fc6 rather than fc7 because previous works show better performance of this layer in a transfer learning setting, e.g., classifying regions using an embedding trained for a different task, object classification [14]. Our task is also transfer learning, since we are using an embedding originally trained for object classification, when our goal is recommendation. Figure 3 shows the architecture and the procedure to obtain the features from fc6 [**?**].

[6] http://caffe.berkeleyvision.org/

We also tested the Visual Geometry Group (VGG) network [45], a newer deep neural network architecture used to classify images. This network outperformed the results obtained by the AlexNet [45] This network was able to reach human level of performance in the task of image classification, in [41] they reported a human error from 5.1% to 12.0% and the VGG reported 6.8% using the same dataset in the same classification task, so it seemed reasonable to put this network to the test in the task of artwork recommendation as well. We used the first fully connected layer of this network also known as fc14, this layer also return a vector with 4096 components. For every image, we take 5 crops of it (upper left, upper right, down left, down right, center) and get a feature vector for each crop. Then, we concatenated them into a single vector of 20.480 components, and used it as the image feature vector.

DNN utility score. We make recommendations by maximizing the utility score that an item provides to a user. Given a user u who has consumed a set of artworks P_u, and an arbitrary artwork i from the inventory, the score of this item i to be recommended to u is defined as:

$$score(u,i)_X = \begin{cases} \max\limits_{j \epsilon P_u} \{sim(V_i^X, V_j^X)\} & (maximum) \\ \frac{\sum\limits_{j \epsilon P_u} sim(V_i^X, V_j^X)}{|P_u|} & (average) \\ \frac{\sum\limits_{r=1}^{\min\{K,|P_u|\}} \max\limits_{j \epsilon P_u}{}^{(r)} \{sim(V_i^X, V_j^X)\}}{\min\{K,|P_u|\}} & (average\ top\ K) \end{cases} \tag{6}$$

, where V_z^X is a feature vector of type X associated to item z. In this particular case V_z^X stands for the $\mathbb{R}^{4096}$ vector embedding of item z obtained with a pre-trained DNN of type X, where X can be either VGG or AlexNet. $\max^{(r)}$ denotes the r-th maximum value, e.g. if $r = 1$ it is the overall maximum, if $r = 2$ it is the second maximum, and so on. $sim(V_i, V_j)$ denotes a similarity function between vectors V_i and V_j. In this particular case, the similarity function used was cosine similarity, expressed as:

$$sim(V_i, V_j) = cos(V_i, V_j) = \frac{V_i \cdot V_j}{\|V_i\| \|V_j\|} \tag{7}$$

Essentially, the score in equation 6 looks at the similarity between item i and each item j in the user profile P_u, and then aggregates these similarities in 3 possible ways: taking either (a) the maximum, (b) the average or (c) the average of the top K most similar items, where K can be tuned empirically.

In addition to studying AlexNet and VGG separately, we also studied the performance of using both DNNs at the same time. For this purpose, we implemented the following hybrid score:

$$\begin{aligned} score(u,i)_{DNN} = \alpha_1 \cdot score(u,i)_{VGG} \\ + \alpha_2 \cdot score(u,i)_{AlexNet} \end{aligned} \tag{8}$$

, where $score(u,i)_{VGG}$ and $score(u,i)_{AlexNet}$ are calculated following equations (6) and (7), using VGG and AlexNet feature vectors, respectively, and α_1 and α_2

are weights to perform the linear combination between the two scores. After an optimization of the weights by grid search, this hybrid approach produced the best results, where the optimal values were $\alpha_1 = 0.8$ and $\alpha_2 = 0.2$.

5.5 Manually Engineered Visual Features (MEVF)

The visual features obtained with DNN techniques are of latent nature, i.e., they are not easily interpretable in terms of more intuitive features such as image colorfulness or brightness . To mitigate this problem, one might want to take advantage of manually engineered visual features which usually are much more intuitive and explainable than neural ones, and also suitable to be used in a search interface to support navigation. For example, imagine a use case recommending based on the brightness of an image. This information could be used to make an explanation –*you might like this image because of its brightness*– or to allow the user to filter search results based on the paintings' level of brightness. In order to choose which visual features to extract, we surveyed related work and found features related to *attractiveness* as potentially useful.

Attractiveness. San Pedro and Siersdorfer in [42] proposed several explainable visual features that can capture to a great extent the attractiveness of an image posted on Flickr. Following their procedure, for every image in our *UGallery* dataset we calculated: (a) average brightness, (b) saturation, (c) sharpness, (d) RMS-contrast, (e) colorfulness and (f) naturalness. In addition, we added (g) entropy, which is a good way to characterize and measure the texture of an image [16]. These metrics have also been used in another study [47], where we show how people nudge with images to take up more healthy recipe recommendations. Since each feature varies within different value ranges (e.g. 0-1, 10-100), we applied a feature-wise min-max normalization, to prevent biases in similarity calculations. Following, we provide a more detailed description of these attractiveness-based features:

- *Brightness*: It measures the level of luminance of an image. For images in the *YUV* color space, we obtain the average of the luminance component Y as follows:
$$B = \frac{1}{N} \sum_{x,y} Y_{x,y} \tag{9}$$
, where N is the amount of pixels and $Y_{x,y}$ is the value of the luminance in the pixel (x, y)
- *Saturation*: It measures the vividness of a picture. For images in the *HSV* or *HSL* color space, we obtain the average of the saturation component S as follows:
$$S = \frac{1}{N} \sum_{x,y} S_{x,y} \tag{10}$$
, where N is the amount of pixels and $S_{x,y}$ is the value of the saturation in the pixel (x, y)
- *Sharpness*: It measures how detailed is the image. For an image in gray-scale, it can be obtained using a Laplacian filter and luminance around every pixel:
$$L(x, y) = \frac{\delta^2 I}{\delta x^2} + \frac{\delta^2 I}{\delta y^2} \tag{11}$$

$$Sh = \frac{\sum_{x,y} \frac{L(x,y)}{\mu_{x,y}}}{n} \tag{12}$$

, where n is the number of pixels and $\mu_{x,y}$ is the average luminance of the pixels around the pixel (x, y).

- *Colorfulness*: It measures how distant are the colors from the gray color. For images in the RGB space, it can be obtained with the following formulas:

$$C = \sigma_{rgyb} + 0.3 \cdot \mu_{rgyb} \tag{13}$$

$$\sigma_{rgyb} = \sqrt{\sigma_{rg}^2 + \sigma_{yb}^2} \tag{14}$$

$$\mu_{rgyb} = \sqrt{\mu_{rg}^2 + \mu_{yb}^2} \tag{15}$$

where μ_{rg}^2, μ_{yb}^2 are the means of the components of the opponent color space. σ_{rg}^2, σ_{yb}^2 are the standard deviations of the component of opponent color space. This color space is defined as:

$$rg = R - G \tag{16}$$

$$yb = \frac{1}{2}(R + G) - B \tag{17}$$

- *Naturalness*: It measures how natural is a picture, grouping the pixels in Sky, Grass and Skins pixels and applying the formula in [42]. First, using the HSL color space, the pixels are filtered considering only the ones with $20 \leq L \leq 80$ and $S > 0.1$. Then, they are grouped by their hue value in the group "A - Skin", "B - Grass" and "C - Sky".
  - pixels with $25 \leq hue \leq 70$ belong to "A - Skin" set.
  - pixels with $95 \leq hue \leq 135$ belong to "B - Grass" set.
  - pixels with $185 \leq hue \leq 260$ belong to "C - Sky" set.

  For each set, average saturation is calculated and denoted as μ_S. Then, local naturalness for each set is calculated using the following formulas:

$$N_{skin} = e^{-0.5\left(\frac{\mu_S^A - 0.76}{0.52}\right)^2} \tag{18}$$

$$N_{Grass} = e^{-0.5\left(\frac{\mu_S^B - 0.81}{0.53}\right)^2} \tag{19}$$

$$N_{Sky} = e^{-0.5\left(\frac{\mu_S^C - 0.43}{0.22}\right)^2} \tag{20}$$

After this, Naturalness value is obtained by:

$$Na = \sum_i \omega_i N_i, \ i \in \{"Skin", "Grass", "Sky"\} \tag{21}$$

, where ω_i is the amount of pixels of set i divided by the total pixels in the image.

- *RMS-contrast*: Measures the variance of luminance in an image using the intensity of each pixel.

$$C^{rms} = \frac{1}{n}\sum_{x,y}^{n}(I_{x,y} - \bar{I})$$

where $I_{x,y}$ is the intensity of the pixel (x, y) and $\bar{I}$ is the average intensity.
- *Entropy*: The entropy of a gray-scale image is a way to measure and characterize the texture of the image [16]. Shannon's entropy is applied to the histogram of values of every pixel in a gray-scale image. The formula is defined as follows:

$$E = -\sum_{x\in[0..255]} p(x)\log p(x) \tag{22}$$

, where $p(x)$ is the probability of finding the gray-scale value x among all the pixels in the image.

Attractiveness utility scores. For the attractiveness features we studied the performance of (i) using each feature individually and (ii) using all features together. For the first case, we used $1D$ vectors of one single feature at a time. To calculate the similarity between two $1D$ vectors, we used euclidean distance, formally expressed as:

$$sim(V_i^X, V_j^X) = \left\| V_i^X - V_j^X \right\| \tag{23}$$

, where V_i^X and V_j^X are $1D$ vectors of items i and j, respectively, containing a single feature of type X (where X can be either *average brightness*, *saturation*, *sharpness*, *RMS-contrast*, *colorfulness*, *naturalness* or *entropy*).

For the second case (all features together), we put the 7 attractiveness-based features into a single $7D$ vector, which we denote as $V_i^{Attract}$. Then, to calculate the similarity between two vectors $V_i^{Attract}$ and $V_j^{Attract}$ we used cosine similarity, as per equation 7:

$$sim(V_i^{Attract}, V_j^{Attract}) = cos(V_i^{Attract}, V_j^{Attract}) \tag{24}$$

As for the utility score ($score(u, i)_X$) itself, we used the same similarity aggregation techniques outlined in equation 6 (*maximum*, *average* and *average-top-k*). This applies for both (i) $1D$ vectors of single features and (ii) $7D$ vectors with all attractiveness features, using the corresponding similarity function in each case.

LBP. Another set of features we explored apart from those of attractiveness were the *Local Binary Patterns* (LBP) [37]. Although this is not an actual "explicit" visual feature, it is a traditional baseline in several computer vision tasks such as image classification, so we tested it for the task of recommendation too. LBP is not represented as an scalar value, but rather as a feature vector of 59 dimensions. The values in the LBP feature vector represent counts in a histogram of the patterns found on an image. Figure 4 shows four of such patterns as example.

LBP utility score. Since the output of LBP is a feature vector, we calculated the similarity between two vectors V_i^{LBP} and V_j^{LBP} as we did with most of the feature vectors, using cosine similarity (7). Namely:

$$sim(V_i^{LBP}, V_j^{LBP}) = cos(V_i^{LBP}, V_j^{LBP}) \tag{25}$$

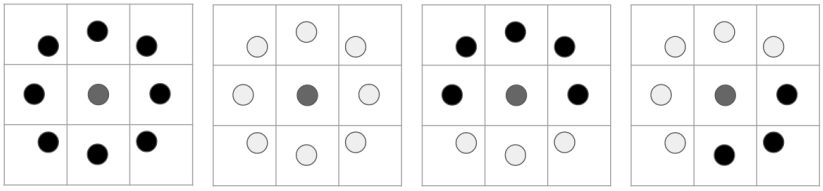

Fig. 4: Examples of pixelwise patterns extracted with local binary patterns (LBP). Each small square is a pixel, and these boxes with 9 pixels each represent patterns. The black circles represent pixels with value over a threshold (=1), while gray circles represent pixels with value below a threshold (=0). The threshold is set by the value of the pixel in the center of the pattern.

Finally, the utility score ($score(u,i)_{LBP}$) is calculated using the same similarity aggregation techniques outlined in equation (6): *maximum*, *average* and *average-top-k*.

MEVF hybrid utility score. In addition to studying Attractiveness and LBP separately, we also studied the performance of using both feature sets at the same time. We tried two approaches: (i) we put all features into a single $66D$ vector (7 + 59 = 66) and then performed recommendations using the same similarity aggregation techniques as we do with feature vectors – based on equations (6) and (7), and (ii) we treated Attractiveness ($7D$) and LBP ($59D$) as two separate vectors, computed one score from each one and merged the two scores with a convex linear combination analogous to the hybrid approach used with DNN – based on equation (8). As we will show in section 7, this hybrid approach achieved the best results.

5.6 Hybrid Recommendations (Hybrid)

Since different methods can measure different sources of similarity between items and the user profile, we developed a hybrid recommender model which integrates the previous approaches. The basic idea is to compute a hybrid score as a convex linear combination of the scores of individual methods. We took the best performing version of each individual method and tested multiple hybrid combinations of them.

Formally, given a user u who has purchased a set of artworks P_u, and an arbitrary artwork i from the inventory, we compute the hybrid score of item i for user u as a convex linear combination of multiple scores, which for the case of combining all features is given by:

$$\begin{aligned} score(u,i)_{Hybrid} = {} & \beta_1 \cdot score(u,i)_{\text{FA}} \\ & + \beta_2 \cdot score(u,i)_{\text{VGG}} \\ & + \beta_3 \cdot score(u,i)_{\text{AlexNet}} \\ & + \beta_4 \cdot score(u,i)_{\text{LBP}} \\ & + \beta_5 \cdot score(u,i)_{\text{Attract}} \\ & + \beta_6 \cdot score(u,i)_{\text{PMPCAV}} \end{aligned} \tag{26}$$

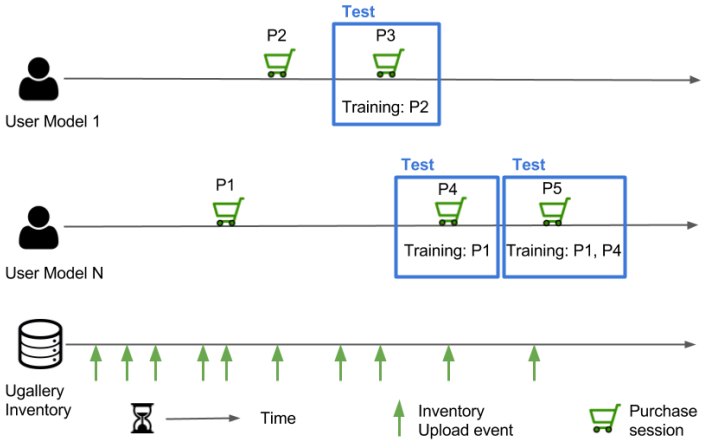

Fig. 5: off-line evaluation procedure. Each surrounding box represents a test, where we predict the items of the purchase session. In the figure, we predict which artworks User 1 bought in purchase P3. 'Training:P2' means we used items from purchase session P2 to train the model.

where β are global (non-personalized) coefficients such that $0 \leqslant \beta_i \leqslant 1$ and $\sum_i \beta_i = 1$. The β coefficients were tuned by exhaustive grid search, and in the case of the hybrid with all features the best coefficients found were $\beta_1 = 0.207$, $\beta_2 = 0.269$, $\beta_3 = 0.165$, $\beta_4 = 0.145$, $\beta_5 = 0.062$ and $\beta_6 = 0.153$. In the equation, $score(u,i)_{\text{VGG}}$, $score(u,i)_{\text{AlexNet}}$, $score(u,i)_{\text{LBP}}$ and $score(u,i)_{\text{Attract}}$ are calculated as in equation (6). Meanwhile, $score(u,i)_{\text{PMPCAV}}$ and $score(u,i)_{\text{FA}}$ had to be slightly modified to ensure normalized values in the range $[0,1]$:

$$score(u,i)_{\text{PMPCAV}} = \frac{\sum_{v \in CAV_i^{All}} \sum_{j \in P_u} \mathbb{1}(j,v)}{\sum_{j \in P_u} |CAV_j^{All}|} \tag{27}$$

$$score(u,i)_{\text{FA}} = \frac{\sum_{j \in P_u} \mathbb{1}(j,a_i)}{|P_u|} \tag{28}$$

, which are almost the same as equations (4) and (5) but with the addition of a normalizing denominator that represents the theoretical maximum of the score in each case.

6 Evaluation Methodology

The evaluation had two stages. The first was an off-line evaluation, conducted using a dataset of transactions (purchases) as described in section 4. With this off-line evaluation we can answer research questions RQ1, RQ2 and RQ3. The second stage was performed with expert curators from the UGallery store. We developed a web interface where the experts could rate recommendations based

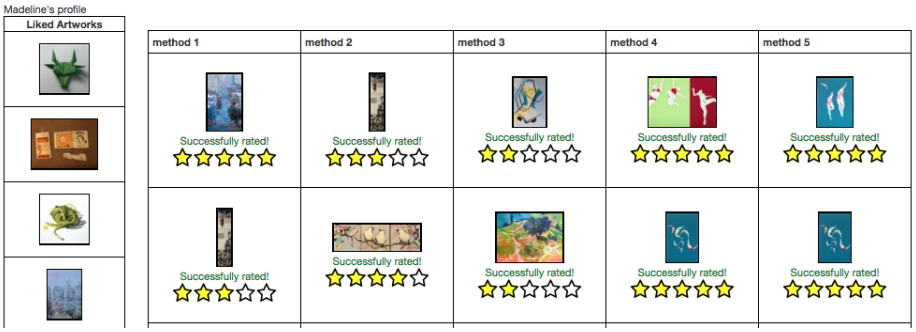

Fig. 6: Screenshot of the upper part of the interface used in the expert evaluation. On the left the items liked by the user. The large table to the right shows one column per each method used to make recommendations.

on algorithms selected from the off-line evaluation, and we analyzed consistency between results of both stages (RQ4).

6.1 Off-line Evaluation

The evaluation protocol we follow in this paper is the one usually used in order to evaluate predictive models and recommender systems off-line in a time-based manner [32]. Hence, the *UGallery* dataset was split into training and test samples according to the time line of every user, as seen in Figure 5. With this setting, we attempt to predict the items purchased by the user in every transaction, where the training set contains all the artworks bought by a user previous to the transaction to be predicted. Figure 5 shows that for every user we test the predictions made for every purchase session excepting the first one of each user. For instance, for User 1 we tested the predicted items of purchase *P3* using items in *P2* as training. In the same Figure, for User N we performed two predictions tasks: the first one predicting items bought in purchase *P4* using *P1* as training, and then testing a prediction on purchase *P5* using *P1* and *P4* as training. In our evaluation, most of the experiments considered only users who had at least 2 purchase sessions. Users who only had a single purchase session in their whole history were considered *cold start* users (only MPCAV and Random were able to make predictions in those cases, since they are non-personalized methods).

6.2 Online Evaluation

The online evaluation involved 8 expert curators from UGallery. We asked each expert to send us a list of 10 of their preferred paintings from the current UGallery dataset, which they sent us via email. For each expert we created five lists of recommendations based on different methods: FA, MEVF, DNN, and the hybrids DNN+MEVF, and FA+DNN+MEVF. Each recommendation list had 10 items, and the experts had to rate each painting recommended with stars in a scale from 1 to 5. In total, each expert rated 50 items. A screenshot of the rating interface for a ficticious user called “Madeline” is shown in Figure 6. We stored the user id, item

Table 4: Evaluation metrics symbol table.

Symbol	Description
t	a test case during the execution of an off-line evaluation of a certain recommendation algorithm
u_t	user whose purchase basket is predicted during off-line test case t
r_t^k	list of top k items recommended to user u_t at off-line test case t
R_t	the set of relevant items (i.e. items in the purchase basket) of user u_t during off-line test case t
T_u	the set of all test cases performed with purchase sessions of user u
U_r	set of all users who received at least 1 recommendation during a certain off-line evaluation (i.e., all $u \in U$ such that $\|T_u\| \geq 1$)
$i_{t,z}$	item appearing at position z in the recommended list at off-line test t
PS	total number of purchase sessions in the system

id and the ratings over every painting for each method, to eventually calculate the evaluation metrics and compare the results.

6.3 Evaluation Metrics

Table 4 shows a summary of symbols used in this section. As suggested by Cremonesi et al. [9] for Top-N recommendation, for our off-line evaluations we used Recall@k ($R@k$), Precision@k ($P@k$) and F1-score@k ($F1@k$), as shown in the equations below:

$$p@k(t) = \frac{|r_t^k \cap R_t|}{k} \tag{29}$$

$$r@k(t) = \frac{|r_t^k \cap R_t|}{|R_t|} \tag{30}$$

$$f1@k(t) = 2 \cdot \frac{p@k(t) \cdot r@k(t)}{p@k(t) + r@k(t)} \tag{31}$$

$$P@k = \frac{1}{|U_r|} \sum_{u \in U_r} \left(\frac{1}{|T_u|} \sum_{t \in T_u} p@k(t) \right) \tag{32}$$

$$R@k = \frac{1}{|U_r|} \sum_{u \in U_r} \left(\frac{1}{|T_u|} \sum_{t \in T_u} r@k(t) \right) \tag{33}$$

$$F1@k = \frac{1}{|U_r|} \sum_{u \in U_r} \left(\frac{1}{|T_u|} \sum_{t \in T_u} f1@k(t) \right) \tag{34}$$

, where $p@k(t)$, $r@k(t)$ and $f1@k(t)$ are precision, recall and f1-score at k, respectively, measured during the test case t, whereas $P@k$, $R@k$ and $F1@k$ are the overall aggregations of precision, recall and f1-score at k, respectively, by first calculating user averages and then average of averages. These are the evaluation metrics that we report in section 7.

In addition we report *Normalized Discounted Cumulative Gain (nDCG)* [34] which is a ranking-dependent metric that not only measures how relevant the items are but also takes the position of the items in the recommended list into account. The $nDCG$ metric with a cut-off of k items in the recommended list is based on the *Discounted Cumulative Gain (DCG@k)* which is defined as follows:

$$DCG@k(t) = \sum_{z=1}^{k} \frac{2^{B_t(i_{t,z})} - 1}{log_2(1+z)} \tag{35}$$

, where $B_t(i_{t,z})$ is a function that returns the graded relevance of item $i_{t,z}$ appearing at position z in the recommended list during the test case t. In our case, $B_t(i_{t,z})$ basically return 1 if item $i_{t,z}$ was present in the shopping basket of test case t, and 0 otherwise. $nD@k$ is calculated as $DCG@k$ divided by the ideal $DCG@k$ value $iDCG@k$ which is the highest possible $DCG@k$ value that can be achieved if all the relevant items were recommended in the correct order (i.e., all purchase basket items appearing first in the recommended list). Taken together, the overall $nD@k$ is defined as follows:

$$nD@k = \frac{1}{|U_r|} \sum_{u \in U_r} \left(\frac{1}{|T_u|} \sum_{t \in T_u} \frac{DCG@k(t)}{iDCG@k(t)} \right) \tag{36}$$

In addition, we calculated *user coverage* (UC), expressed as:

$$UC = \frac{|U_r|}{|U|} \tag{37}$$

User Coverage is defined as the number of users for whom at least one recommendation could be generated ($|Ur|$) divided by total number of users $|U|$ [28].

We also report *session coverage* (SC), expressed as:

$$SC = \frac{\sum_{u \in U_r} |T_u|}{PS} \tag{38}$$

Session Coverage is defined as the number of purchase sessions in which the recommender was able to generate a recommendation (i.e., total number of valid test cases) divided by the total number of purchase sessions of the system (PS).

Another metric we measured as well is the Number of Artists (NA), expressed as:

$$NA@k = \frac{\sum_{u \in U_r} \sum_{t \in T_u} \left| \bigcup_{i \in r_t^k} \{ a_i \} \right|}{\sum_{u \in U_r} |T_u|} \tag{39}$$

The Number of Artists measures the average number of distinct artists per recommendation. This metric is useful for getting a notion of how diverse is a recommendation in terms of the different artists recommended. The larger the metric, the more the chances of recommending items from novel artists to users.

Finally, another important dimension we measured was the visual diversity between the items recommended within the same list. We calculate diversity as:

$$D@k(t) = \frac{D@k(t)_{VGG} + D@k(t)_{AlexNet} + D@k(t)_{LBP} + D@k(t)_{Attract}}{4} \quad (40)$$

, where $D@k(t)_X$ is the pair-wise average distance in the recommended list associated to test case t, using visual feature vectors of type X, where X can be either *VGG*, *AlexNet*, *LBP* or *Attract*. Formally, $D@k(t)_X$ is expressed as:

$$D@k(t)_X = \frac{\sum_{y=1}^{k-1} \sum_{z=y+1}^{k} \left[1 - cos(V^X_{i_{t,y}}, V^X_{i_{t,z}})\right]}{\frac{k\cdot(k-1)}{2}} \quad (41)$$

, where $i_{t,y}$ and $i_{t,z}$ are the items at positions y and z of the recommended list of the test case t, respectively, and V_i^X is item i's visual feature vector of type X. Thus, the overall diversity $D@k$ is finally calculated as follows:

$$D@k = \frac{\sum_{u \in U_r} \sum_{t \in T_u} D@k(t)}{\sum_{u \in U_r} |T_u|} \quad (42)$$

In addition to these off-line evaluation metrics, we also report Precision@k and nD@k for the on-line evaluation with 8 UGallery expert curators. In this setting, the metrics were calculated as follows:

$$nD@k = \frac{1}{8} \sum_{x=1}^{8} \frac{DCG@k(x)}{iDCG@k(x)} \quad (43)$$

$$DCG@k(x) = \sum_{z=1}^{k} \frac{2^{B_x(i_{x,z})} - 1}{log_2(1+z)} \quad (44)$$

$$P@k = \frac{1}{8} \sum_{x=1}^{8} p@k(x) \quad (45)$$

$$p@k(x) = \frac{1}{k} \sum_{z=1}^{k} \mathbb{1}_x(i_{x,z}) \quad (46)$$

, where x stands for the x-th expert curator, $i_{x,z}$ is the item appearing at position z in the list recommended to expert x, $B_x(i_{x,z})$ returns the original rating $S_x(i_{x,z})$ given by expert x to item $i_{x,z}$ if $S_x(i_{x,z})$ >= 4, or 0 otherwise, and $\mathbb{1}_x(i_{x,z})$ is an indicator function that returns 1 if rating $S_x(i_{x,z}) \geq 4$, or 0 otherwise (i.e., we used 4 as the relevance threshold for the calculation of these metrics).

Table 5: nDCG (nD), Recall (R), Precision (P), Diversity (D), and Coverage (UC and SC) for MPCAV and PMPCAV by attribute. The best three absolute results of each metric are highlighted. The superindex indicates the ID of the method with the closest but still significantly smaller result. For instance, FA R@5 = .1600^{12} tells that FA is significantly larger than at least (12) PMPCAV(All) R@5 = .0785, as well as significantly larger than all the other methods with R@5 < .0785.

ID	Method	nD@5	nD@10	R@5	R@10	P@5	P@10	D@10	UC	SC
1	MPCAV(Subject)	.0082	.0115	.0099	.0172	.0025	.0023	**.2372**^{14}	.9985	.9991
2	MPCAV(Medium)	.0073	.0106	.0110	.0211	.0027	.0025	.2151^{6}	**.9993**	**.9995**
3	MPCAV(Style)	.0059	.0096	.0089	.0176	.0023	.0025	.2159^{10}	.9978	.9972
4	MPCAV(Color)	.0056	.0095	.0083	.0190	.0021	.0023	.1910	**.9993**	**.9995**
5	MPCAV(Mood)	**.0090**	**.0148**	**.0123**	**.0279**	**.0029**	**.0034**	.2205^{3}	.8483	.8229
6	MPCAV(All)	.0054	.0087	.0063	.0157	.0019	.0020	.2085^{13}	**.9993**	**.9995**
7	PMPCAV(Subject)	.0064	.0099	.0063	.0136	.0020	.0021	.1903	.0890	.1407
8	PMPCAV(Medium)	.0113	.0190	.0144	.0363	.0036	.0044	.2124^{13}	**.2640**	**.3593**
9	PMPCAV(Style)	.0154	.0237	.0238	.0485	.0065	.0060	.1848	.0766	.1168
10	PMPCAV(Color)	.0165	.0264	.0228	.0486^{3}	.0060	.0063	.2104^{4}	.2619	.3570
11	PMPCAV(Mood)	**.0373**	**.0507**	**.0483**	.0774	**.0113**^{1}	**.0098**	**.2162**^{2}	.1327	.1822
12	PMPCAV(All)	.0333^{1}	.0448^{14}	.0468^{14}	**.0785**^{5}	.0111^{1}	.0095^{5}	.2115^{6}	**.2640**	**.3593**
13	FA	**.1209**^{11}	**.1380**^{11}	**.1600**^{12}	**.2067**^{11}	**.0383**^{11}	**.0259**^{11}	**.1927**	**.2640**	**.3593**
14	Random	.0083	.0122	.0105	.0214	.0029	.0027	.2292^{5}	1.0000	1.0000

Stat. significance by multiple t-tests, Bonferroni corr. $\alpha_{bonf} = \alpha/n = 0.05/91 = .00055$.

7 Results

In this section, we report the results focusing on different aspects. With respect to research question RQ1 –analyzing the impact of each single feature–, we analyze: a) metatadata features (personalized and non-personalized), and b) visual features (DNN and MEVF). For RQ2, we compare between visual features and metadata. Regarding research question RQ3, we test several combinations of features to identify the best hybrid recommender. Finally, regarding research question RQ4, the on-line validation, we report and discuss the results of recommendations evaluated by expert curators from UGallery.

7.1 Metadata features (RQ1)

Table 5 summarizes all the results for this analysis of metadata features. Here we report MPCAV (Most Popular Curated Attribute Value), its personalized version PMPCAV, and Favorite artist (FA). First we discuss results separately to assess the impact of different attributes and then comparing the attribute sets against each other in order to assess the value of personalization. We also report the results of the method Favorite Artist (FA), based on the artwork's artist.

MPCAV. We tested the performance of MPCAV features separately as well as combined (*MPCAV(All)*). The results in Table 5 show that these results are not significantly different from random prediction in the performance metrics studied (nDCG, Recall, Precision). In terms of Diversity, MPCAV Subject presents significantly more diverse results than random.

PMPCAV. For the personalized metadata approach, in Table 5 we see a small but significant improvement in the ranking metrics over MPCAV, but it is necessary to combine them all (*PMPCAV(All)*) in order to obtain results significantly better than random, achieving an improvement by 3 to 4 times over it. The attribute Mood seems to report good results in several ranking metrics (*PMPCAV(Mood)*)), but at the expense of a very low user and session coverage (UC=.1327, SC=.1822). Notably, the accuracy improvement does not critically affect the diversity of results compared to the non-personalized alternatives.

Favorite Artist (FA). One result that stands out overall is the performance of the artist feature. In this method, we tested whether making personalized recommendations from the user's most frequently purchased artists could provide good recommendations. Our results, indicate that FA is actually the single most accurate feature ($nD@5 = 0.1209$, $nD@10 = 0.1380$, $R@5 = 0.16$, $R@10 = 0.2067$), between 3 to 4 times better than the second best method –the combination of all PMPCAV features. The only downside of this method is that it recommends on average from a small number of artists (1.5 in average), which prevents the promotion of a more diverse set of artists in the UGallery website, a core component of their business.

MPCAV vs PMPCAV. The most outstanding lesson about these features is the relatively poor performance of both curation-based methods with respect to a random baseline, although personalization (PMPCAV) produces a significant improvement. Our results support the importance of personalization to improve the performance, as seen in Table 5. As additional evidence, all the other more sophisticated personalized methods (EVF, DNN, FA and Hybrids) are significantly better than MPCAV, shown in Table 7. Regarding diversity, there are significant but rather small differences between both approaches.

7.2 DNN and MEVF Visual Features (RQ1)

To the best of our knowledge, our work presents the first analysis comparing the performance of manually engineered visual features (brightness, contrast, etc.) versus automatically extracted features (DNN) for the task of recommending artworks. Table 6 presents the results, where it is clear that DNN embeddings significantly outperform all MEVF features, either combined or isolated, almost doubling their performance in almost all the performance metrics. This result reflects the current state-of-the-art of deep neural networks in computer vision, which outperform other methods in several tasks [21, 44, 19]. Diversity though, is affected in the opposite direction: manually engineered features (MEVF) result in significantly more diversity than DNN embeddings.

Combining AlexNet and VGG shows a small improvement over using either DNN isolated, but actually the statistical tests show no significant differences between them.

Combining MEVF features improves their performance compared to using them isolated. The improvement is specially notorious in the case of Attractiveness, where using each feature isolated shows poor results; performance of isolated features is not significantly different from random. In terms of the manually-engineered visual features, combining Attractiveness and LBP yields the best results, showing an improvement of about 400% above random.

Table 6: nDCG (nD), Recall (R), Precision (P), Diversity (D). User and Session Coverage are all the same for every experiment, UC=.2640 and SC= .3593. The best absolute result of each metric is highlighted. The superindex indicates the ID of the method with the closest but still significantly smaller result. For instance, DNN-2 R@5 = $.1239^{4}$ tells that DNN-2 is significantly larger than at least (4)MEVF (LBP+Attract:all) R@5 = .0607, as well as significantly larger than all the other methods with R@5 < .0607.

ID	Method	nD@5	nD@10	R@5	R@10	P@5	P@10	D@10
1	DNN-2 (VGG+AlexNet)	$\mathbf{.1043}^{4}$	$\mathbf{.1187}^{4}$	$\mathbf{.1239}^{4}$	$\mathbf{.1671}^{4}$	$\mathbf{.0307}^{4}$	$\mathbf{.0210}^{4}$	.1503
2	DNN (VGG)	$.0965^{5}$	$.1123^{4}$	$.1178^{4}$	$.1614^{4}$	$.0295^{4}$	$.0203^{4}$	.1524
3	DNN (AlexNet)	$.0929^{5}$	$.1094^{4}$	$.1111^{4}$	$.1571^{4}$	$.0279^{4}$	$.0201^{4}$	$.1529^{1}$
4	MEVF (LBP + Attract:all)	$.0514^{8}$	$.0674^{7}$	$.0607^{8}$	$.0998^{7}$	$.0144^{7}$	$.0118^{10}$	$.1784^{3}$
5	MEVF (LBP)	$.0363^{11}$	$.0500^{7}$	$.0538^{12}$	$.0897^{7}$	$.0119^{11}$	$.0104^{7}$	$.1869^{4}$
6	MEVF (Att: all)	$.0328^{11}$	$.0424^{7}$	$.0446^{10}$	$.0637^{9}$	$.0109^{11}$	$.0085^{7}$	$.1987^{5}$
7	MEVF (Att: contrast)	.0093	.0120	.0159	.0230	.0037	.0027	$.2310^{8}$
8	MEVF (Att: naturalness)	.0084	.0106	.0151	.0204	.0035	.0025	$.2270^{6}$
9	MEVF (Att: saturation)	.0059	.0091	.0096	.0197	.0021	.0021	$.2387^{14}$
10	MEVF (Att: brightness)	.0053	.0085	.0080	.0186	.0035	.0031	$.2217^{5}$
11	MEVF (Att: sharpness)	.0068	.0095	.0117	.0178	.0024	.0021	$.2467^{9}$
12	MEVF (Att: entropy)	.0095	.0106	.0110	.0137	.0026	.0019	$\mathbf{.2893}^{13}$
13	MEVF (Att: colorfulness)	.0022	.0073	.0034	.0132	.0008	.0020	$.2587^{11}$
14	Random	.0083	.0122	.0105	.0214	.0029	.0027	.2292

Stat. significance by multiple t-tests, Bonferroni corr. $\alpha_{bonf} = \alpha/n = 0.05/91 = .00055$.

When comparing isolated MEVF features, we observe that *LBP* performs the best (about 300% better than random) because it encodes texture patterns and local contrast very well, however its explanation is more complex than e.g. image *brightness* or *contrast*.

In summary, these results show the positive side of providing more support to the use of pre-trained deep neural networks for transfer learning. Their only drawback is how difficult is knowing exactly what the neural networks encode. They are mostly black boxes, and this aspect increases the difficulty of producing explanations of recommendations based on these DNN visual features. This lack of transparency and explainability can hinder the user acceptance of these recommendations [24, 48].

7.3 Comparing Visual Features vs Metadata (RQ2)

Visual Features vs Curated Attributes. From Table 7, which shows results of the overall analysis, we observe that both DNN and MEVF methods significantly outperformed curation-based methods (PMPCAV and MPCAV). MEVF shows a performance significantly better than some combinations of manually curated metadata –vs. PMPCAV+MPCAV (Mood) and MPCAV (Mood)– but not better than PMPCAV(All). On the other side, DNN features always outperform both MEVF and PMPCAV(All), showing the potential of neural networks for automatic feature generation and selection. In general, these results indicate that for the task

of recommendation it is possible to leverage automatic visual feature extraction techniques from artwork images to achieve better performance, rather than having to go through the whole dataset and manually tag each item.

Visual Features vs Favorite Artist (FA). In total contrast to curated attributes, recommending based on the user's favorite artists surprisingly outperforms both MEVF and DNN in terms of ranking metrics in the off-line evaluation, as can be seen in Table 7. In fact, FA ($nD@ = 0.1267$ and $R@10 = 0.2067$) outperforms the best DNN ($nD@ = 0.1074$ and $R@10 = 0.1671$) by more than a 20%. These off-line results may be explained by the fact that users are more likely to explore and find items they like from artists they are more familiar with. However, when we look at the online results with expert curators (Table 8), the differences between FA and visual methods become much narrower, with even DNN and the hybrid DNN+MEVF showing better results than FA practically in all metrics. This shows that FA is a very good heuristic for filtering the items search space when predicting next purchases (as reflected off-line), but at the same time it is a heuristic with the weakness of being blind to the visual content of artworks, which is reflected in the less favorable results in the online evaluation compared to DNN and MEVF. Additionally, FA has the problem of recommending from very few artists on average ($NA@10 = 1.5142$), whereas the visual methods tend to recommend from 4-5 different artists which is much more helpful in terms of promoting novel artists to users.

7.4 Hybrid recommendation (RQ3)

The Hybrid recommenders, summarized in Table 7, show a clear tendency: they always beat the performance of the isolated features they are combining, with the exception of Favorite Artist (FA), which is only significantly smaller than the top hybrids in terms of P@10. Although there are no significant differences between the top 4 Hybrid methods, there is a regular trend towards showing Hybrid_2(FA+DNN-2+MEVF) as the best combination. The hybrid methods which combine only visual features (Hybrid_5 and Hybrid_6) usually outperform MEVF and metadata-based methods (PMPCAV), and are never significantly smaller than FA, though clearly more expensive to obtain. Nevertheless, we again higlight that the biggest problem of FA is their lack of diversity. With respect to artist diversity, it's worth noting that the top 4 hybrid methods on average achieved $NA@10 = 3$ approximately, twice as many as FA ($NA@10 = 1.5$ approx.), which means they not only are better rankers, but also are able to recommend from a larger number of artists than FA. Under the light of these offline results, it is then interesting answering whether the online validation with expert users is consistent or not.

7.5 Validation with Expert Users (RQ4)

Table 8 presents the results of the evaluation with experts, showing the mean over four metrics: nDCG@5, nDCG@10, Precision@5 and Precision@10. The most important aspect to highlight is that results are consistent with the offline analysis in terms of the performance of the hybrid method combining FA with visual features over the hybrid of solely visual features and over single feature sets (FA,

Table 7: Results of experiments for all methods. The best three absolute results of each metric are highlighted. The superindex indicates the ID of the method with the closest but still significantly smaller result. For instance, Hybrid_1 R@5 = $.1758^{8}$ tells that Hybrid_1 is significantly larger than at least (8)DNN-2 R@5 = .1239, as well as significantly larger than all the other methods with R@5 < .1239.

ID	Method	nD@5	nD@10	R@5	R@10	P@5	P@10	F1@5	F1@10	D@10	NA@10	UC	SC
1	Hybrid_1 (FA+DNN-2+MEVF+PMPCAV)	**.1429**^{9}	**.1660**^{6}	**.1758**^{8}	**.2414**^{6}	**.0424**^{9}	**.0296**^{5}	**.0662**^{8}	**.0515**^{6}	.1701^{3}	3.2534	**.2640**	**.3593**
2	Hybrid_2 (FA+DNN-2+MEVF)	**.1465**^{7}	**.1695**^{5}	**.1758**^{8}	**.2379**^{6}	**.0427**^{8}	**.0296**^{5}	**.0664**^{8}	**.0513**^{6}	.1597^{10}	3.0665	**.2640**	**.3593**
3	Hybrid_3 (FA+DNN-2)	**.1475**^{7}	**.1680**^{5}	**.1764**^{8}	**.2373**^{6}	**.0428**^{8}	**.0292**^{5}	**.0666**^{8}	**.0507**^{6}	.1627	3.0048	**.2640**	**.3593**
4	Hybrid_4 (FA+MEVF)	.1394^{10}	.1569^{10}	**.1780**^{4}	.2252^{9}	.0420^{10}	**.0272**^{11}	.0659^{9}	.0474^{11}	.1917^{11}	2.9921	**.2640**	**.3593**
5	FA	.1209^{11}	.1380^{11}	.1600^{11}	.2067^{11}	.0383^{11}	.0259^{11}	.0595^{11}	.0446^{11}	.1924^{11}	1.5142	**.2640**	**.3593**
6	Hybrid_5 (DNN-2+MEVF+PMPCAV)	.1060^{11}	.1207^{11}	.1314^{11}	.1767^{11}	.0320^{11}	.0215^{11}	.0500^{11}	.0376^{11}	.1579^{10}	4.4782	**.2640**	**.3593**
7	Hybrid_6 (DNN-2+MEVF)	.1068^{11}	.1204^{11}	.1272^{11}	.1713^{11}	.0322^{11}	.0215^{11}	.0496^{11}	.0374^{11}	.1467	4.5432	**.2640**	**.3593**
8	DNN-2	.1043^{11}	.1187^{11}	.1239^{11}	.1671^{11}	.0307^{11}	.0210^{11}	.0476^{11}	.0365^{11}	.1503^{7}	4.5265	**.2640**	**.3593**
9	DNN (VGG)	.0965^{12}	.1123^{11}	.1178^{11}	.1614^{11}	.0295^{11}	.0203^{11}	.0456^{11}	.0352^{11}	.1524^{8}	4.6334	**.2640**	**.3593**
10	DNN (AlexNet)	.0929^{12}	.1094^{11}	.1111^{11}	.1571^{11}	.0279^{11}	.0201^{11}	.0431^{11}	.0348^{11}	.1528^{8}	4.6282	**.2640**	**.3593**
11	MEVF	.0514^{15}	.0674^{13}	.0558^{15}	.0999^{13}	.0139^{14}	.0122^{12}	.0215^{13}	.0213^{13}	.1784^{1}	5.2033	**.2640**	**.3593**
12	PMPCAV (All)	.0333	.0448^{14}	.0468	.0785^{13}	.0111	.0095^{13}	.0175^{14}	.0165^{13}	.2126^{5}	5.0823	**.2640**	**.3593**
13	PMPCAV+MPCAV (Mood)	.0109	.0168	.0146	.0301	.0035	.0037	.0054	.0063	**.2212**^{12}	**6.8092**	**.8483**	**.8229**
14	MPCAV (Mood)	.0090	.0148	.0123	.0279	.0029	.0034	.0046	.0059	**.2208**^{5}	**6.7951**	**.8483**	**.8229**
15	Random	.0083	.0122	.0105	.0214	.0029	.0027	.0042	.0046	**.2292**^{13}	**7.8234**	**1.0000**	**1.0000**

Note: Statistical significance was obtained using multiple pairwise t-tests with Bonferroni correction, $\alpha_{bonf} = \alpha/n = 0.05/105 = .00048$.

Table 8: nD@5, nD@10, P@5 and P@10 for algorithms tested with 8 UGallery experts. For nD@k, all ratings <= 3 were set to 0. For P@k, only ratings ≥ 4 were regarded as relevant.

Name	nD@5	nD@10	P@5	P@10
Hybrid (FA+DNN+MEVF)	**0.9042**	**0.8913**	**0.7500**	**0.6750**
Hybrid (DNN+MEVF)	0.6747	0.6638	0.5000	0.4250
DNN	0.7176	0.6947	0.5000	0.4000
FA	0.4276	0.5662	0.3000	0.4000
MEVF	0.5498	0.5314	0.3500	0.2625

DNN, MEVF). In particular, combining FA+DNN+MEVF outperforms all the other features, either hybrid or single, in all four metrics. Another interesting result is that DNN shows better performance than FA, which is the opposite to the offline evaluation. We think that this might be due to the lack of diversity that FA promotes, but also to the inherent "gambling" involved when sampling artworks from artists to fit a top$-n$ recommendation without any awareness of the visual content, which is much more penalized in an online setting. It is also notable that the isolated features show smaller differences between them in this user experiment than in the offline evaluation. In terms of nDCG@5, nDCG@10 and Precision@5, DNN seems to outperform both FA and MEVF, while it has the same performance as FA in terms of Precision@10. Given the small sample size, we can not report tests of statistical significance, but the trend of results points toward implementing a hybrid recommender with FA and visual features for the best performance without hindering diversity.

8 Summary & Discussion

The main findings with respect to our RQs can be summarized as follows:

- **RQ1. Metadata**: In general, using the most popular curated attribute values (MPCAV) performed not significantly different than random prediction. The Personalized version PMPCAV, specially the one using all attributes, performed significantly better than the non-personalized version MPCAV, but still the results were rather poor. Notably, just recommending based on user's favorite artist produced very high ranking metrics.
- **RQ1. Visual Features**: The features automatically obtained from pre-trained neural networks (DNN) significantly outperformed manually-engineered visual features (MEVF). This is an interesting result, considering that the AlexNet and VGG neural networks were trained for object classification, not for recommendation. This supports the use of transfer learning.
- **RQ2. Visual Features vs. metadata**: Visual features performed better than curated attributes. This is an important result since it points towards using features extracted directly from the images rather than spending resources for manually tagging the images. However, the single best predictive feature overall was Favorite Artist, so combining the strenghts of both visual features and FA seems like a promising approach.

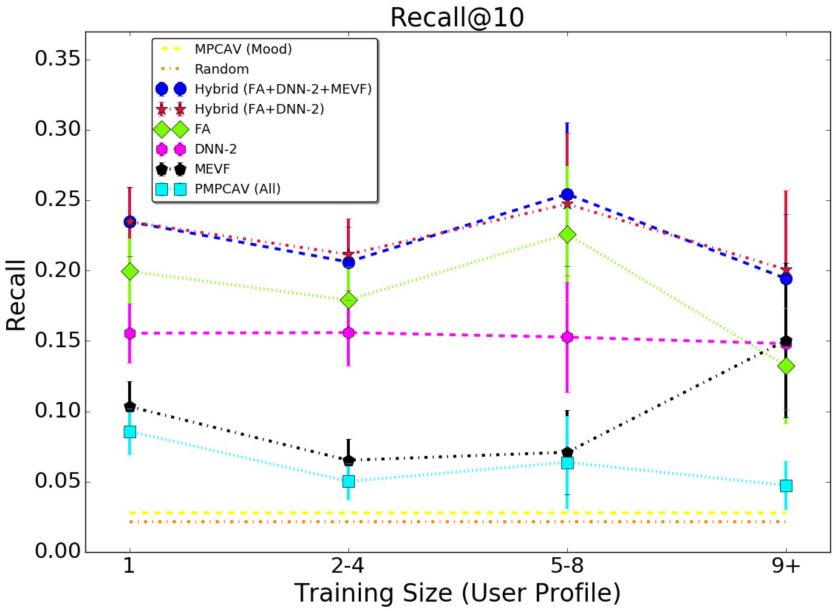

Fig. 7: Recall@10 of different methods at different user profile sizes.

- **RQ3. Hybrids**: Hybrid methods outperformed single visual features. The hybrid method which combined FA, DNN and MEVF produced the best results (a variant including PMPCAV performed equally well), in both offline and online evaluation.
- **RQ4. Expert online evaluation**: The expert evaluation allows us to show the consistency of the offline results when assessed by real people. There was consistency in terms of the best hybrid (FA+DNN+MEVF), which outperformed the other 4 alternatives. Also notable was the small difference among isolated features (DNN, MEVF, FA) compared to their offline results.

Taken together, our results show that a recommender system which utilizes several types of content could indeed support people who buy artworks online based on their personal taste. Moreover, we have some additional thoughts with respect to the intriguing high predictive power of favorite artist and the risk of relying solely on features such as those from neural networks.

Our offline evaluation results indicate that the method FA (based simply on sampling artworks from the favorite artists of a user) performs really well, with a 20%-30% improvement over the next competitor DNN, whereas the best Hybrid improves FA by a smaller margin of 10%-23%. We investigated further whether the size of the user profile (items in training) could give us more evidence of this effect. Our intuition behind this analysis is that artists have in average 8 artworks for sale, and if a customer buys them all, then it will be more difficult to predict the next potential favorite artist. Figure 7 shows the recall@10 of different methods considering different user profile sizes. The plot shows that DNN, EVF, and PMPCAV return always very consistent results independent of profile size and that, among them, DNN performs the best. FA and hybrid methods perform better

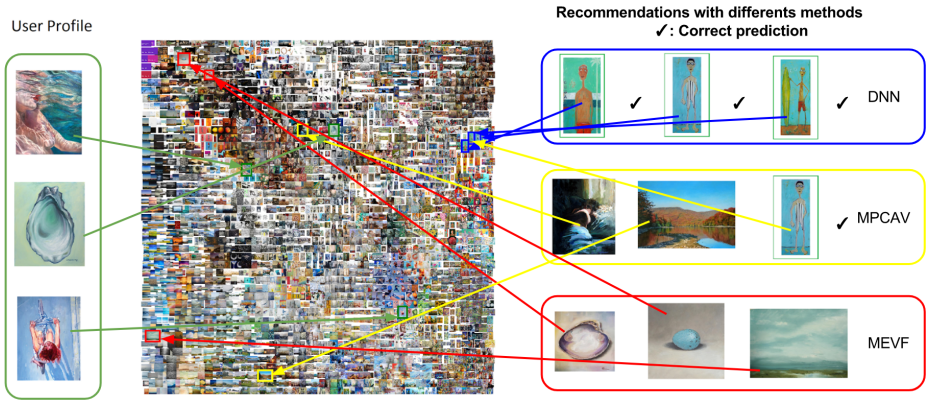

Fig. 8: t-SNE map of the DNN image embedding displaying paintings of an anonymized user profile (green), and recommendations contextualized with three methods: DNN (blue), MPCAV (yellow) and MEVF (red). Check marks indicate correct predictions.

than DNN and MEVF up to user profiles with 5-8 items. However, with larger user profiles (9+) DNN and MEVF seem to improve or maintain results, whereas other methods such as FA suffer an important decrease. This implies that DNN and MEVF might be capturing features related to long-term artistic taste. This analysis highlights the need for more sophisticated methods or additional data, maybe considering temporal decay to account for the effect of users' preference drift over time [25,29].

Furthermore, we show evidence that deep neural networks can be of great help in the field of personalized artwork recommendations, since they decrease the cost of domain expert knowledge to identify the visual features which can be most successful, with a small compromise on diversity. However, in order to make recommendations really useful and not only persuasive [46], researchers and developers need to make sure that people can inspect and have a sense of control [22,38], which is achieved by combining latent easy-to-engineer information (such as features from deep learning models) with actual explicit features, such as artist, color, style, or brightness. One way we have thought of to provide users with such control is by using techniques such as t-SNE with an interactive interface. t-SNE [31] is a dimensionality reduction technique commonly used to visualize what DNN embeddings might encode [18,19,36]. This technique has interesting potential to help users visualize high-dimensional data into a lower-dimensional space in order to understand recommendations, explore them and inspect them, features associated with improved user satisfaction [22,48]. For instance, Figure 8 uses t-SNE to reduce DNN embeddings and then display an anonymized user profile and the images predicted by three different methods: DNN, MPCAV and MEVF. We could perform a similar process over the MEVF embedding and show users differences between both representations, as well as allowing them rich exploration. We foresee building rich visual applications providing user control, transparency and explainability, important characteristics to build user trust and acceptance on recommendations [46,12].

Limitations. An important aspect to bear in mind when interpreting our results is that they relate to only one single artwork retailer website, although one of the most popular on the Web. This might hinder the generalizability of our results. In addition, other forms of user evaluation are needed in order to test whether user evaluation correlates with our offline results, such as a large controlled laboratory study as well as a field online study using A/B testing.

9 Conclusions and Future Work

In this article, we have presented several notable results in the area of content-based artwork recommendation under the one-of-a-kind item problem. We have investigated the potential of several different features for this task. As our results reveal (in the context of a physical artwork online store dataset named UGallery), curated metadata performs not better than random predictions, unless it is combined in a personalized manner, which can improve the results by a small margin. However, recommending solely based on the favorite artist (FA) of the user can yield, surprisingly, very good results, at the expense of a small diversity in recommendation lists. Moreover, we found that visual features are more useful in predicting future purchases than manually curated metadata. Among the visual features investigated, image embeddings from Deep Neural Networks work better than manually-engineered visual features, but overall, the hybrid combination of FA+DNN+MEVF produces the best results. Finally, a user study with expert curators from UGallery supports the use of a hybrid combining FA+DNN+MEVF for the optimal results.

In a deeper analysis, our study of the user profile sizes revealed that time may play an important role in recommending artwork to people. Though further investigation is needed, our results that consider different user profile sizes for training the user models can produce important differences in terms of Recall@k. As such we are interested in investigating the time dimension in more detail, which was not the focus of this work so far. The previous work by Hidasi et al [20] which introduces a neural network model for feature-rich session-based recommendations could be starting point in this direction.

In this work we focused on comparing useful content features rather than on developing state-of-the-art recommendation models. As several new neural network architectures have been introduced to the recommender and visualization communities, we could apply some of these approaches to our problem. One example of such architectures is 'autoencoders' to learn in an unsupervised manner compact representations of images, as for instance in the work of David et al.[10], who use autoencoders to learn an unsupervised compact representation of images, and eventually used the learned embedding for a supervised painter classification task. This procedure could allow us to learn different image embeddings to eventually use them for learning a recommendation model. We can also test a siamese network architecture to learn a model which can help directly rank images given a specific user, using all the rich content-based information as input, as well as dealing with cold start, as the work by Koch [23] suggests.

10 Acknowledgements

This research has been supported by the Chilean research agency Conicyt, under Fondecyt Grant 11150783, and partially funded by the Millennium Institute for Foundational Research on Data (IMFD). We also acknowledge the help from Felipe del Río and Domingo Mery, who helped us frame some evaluations and provided us with some interesting ideas for future work.

References

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering **17**(6), 734–749 (2005)
2. Akay, S., Kundegorski, M.E., Devereux, M., Breckon, T.P.: Transfer learning using convolutional neural networks for object classification within x-ray baggage security imagery. In: Proceedings of the IEEE International Conference on Image Processing (ICIP), pp. 1057–1061 (2016)
3. Albanese, M., d'Acierno, A., Moscato, V., Persia, F., Picariello, A.: A multimedia semantic recommender system for cultural heritage applications. In: Proceedings of the Fifth IEEE International Conference on Semantic Computing (ICSC), pp. 403–410 (2011)
4. Amatriain, X.: Mining large streams of user data for personalized recommendations. ACM SIGKDD Explorations Newsletter **14**(2), 37–48 (2013)
5. Aroyo, L., Wang, Y., Brussee, R., Gorgels, P., Rutledge, L., Stash, N.: Personalized museum experience: The rijksmuseum use case. In: Proceedings of Museums and the Web (2007)
6. Bennett, J., Lanning, S., et al.: The netflix prize. In: Proceedings of KDD cup and workshop, vol. 2007, p. 35 (2007)
7. Benouaret, I., Lenne, D.: Personalizing the museum experience through context-aware recommendations. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 743–748 (2015)
8. Celma, O.: Music recommendation. In: Music Recommendation and Discovery, pp. 43–85. Springer (2010)
9. Cremonesi, P., Koren, Y., Turrin, R.: Performance of recommender algorithms on top-n recommendation tasks. In: Proceedings of the Fourth ACM Conference on Recommender Systems, RecSys '10, pp. 39–46 (2010)
10. David, O.E., Netanyahu, N.S.: DeepPainter: Painter Classification Using Deep Convolutional Autoencoders, pp. 20–28. Springer International Publishing (2016)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 248–255 (2009)
12. Ekstrand, M.D., Kluver, D., Harper, F.M., Konstan, J.A.: Letting users choose recommender algorithms: An experimental study. In: Proceedings of the 9th ACM Conference on Recommender Systems, RecSys '15, pp. 11–18 (2015)
13. Esman, A.R.: The World's Strongest Economy? The Global Art Market. https://www.forbes.com/sites/abigailesman/2012/02/29/the-worlds-strongest-economy-the-global-art-market/ (2012). [Online; accessed 21-March-2017]
14. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580–587 (2014)
15. Gomez-Uribe, C.A., Hunt, N.: The netflix recommender system: Algorithms, business value, and innovation. ACM Transactions on Management Information Systems (TMIS) **6**(4), 13 (2016)
16. Gonzalez, R.C., Eddins, S.L., Woods, R.E.: Digital Image Publishing Using MATLAB. Prentice Hall (2004)
17. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. ACM Trans. Interact. Intell. Syst. **5**(4), 19:1–19:19 (2015)

18. He, R., Fang, C., Wang, Z., McAuley, J.: Vista: A visually, socially, and temporally-aware model for artistic recommendation. In: Proceedings of the 10th ACM Conference on Recommender Systems, RecSys '16, pp. 309–316 (2016)
19. He, R., McAuley, J.: VBPR: Visual Bayesian Personalized Ranking from implicit feedback. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, pp. 144–150 (2016)
20. Hidasi, B., Quadrana, M., Karatzoglou, A., Tikk, D.: Parallel recurrent neural network architectures for feature-rich session-based recommendations. In: Proceedings of the 10th ACM Conference on Recommender Systems, RecSys '16, pp. 241–248 (2016)
21. Karnowski, J.: AlexNet + SVM. `https://jeremykarnowski.files.wordpress.com/2015/07/alexnet2.png` (2015). [Online; accessed 1-December-2017]
22. Knijnenburg, B.P., Bostandjiev, S., O'Donovan, J., Kobsa, A.: Inspectability and control in social recommenders. In: Proceedings of the sixth ACM conference on Recommender systems, pp. 43–50 (2012)
23. Koch, G., Zemel, R., Salakhutdinov, R.: Siamese neural networks for one-shot image recognition. In: Proceedings of ICML Deep Learning Workshop, vol. 2 (2015)
24. Konstan, J.A., Riedl, J.: Recommender systems: from algorithms to user experience. User Modeling and User-Adapted Interaction **22**(1-2), 101–123 (2012)
25. Koren, Y.: Collaborative filtering with temporal dynamics. Communications of the ACM **53**(4), 89–97 (2010)
26. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Proceedings of Advances in neural information processing systems 25 (NIPS), pp. 1097–1105 (2012)
27. La Cascia, M., Sethi, S., Sclaroff, S.: Combining textual and visual cues for content-based image retrieval on the world wide web. In: Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries, pp. 24–28 (1998)
28. Lacic, E., Kowald, D., Eberhard, L., Trattner, C., Parra, D., Marinho, L.B.: Utilizing online social network and location-based data to recommend products and categories in online marketplaces. In: Mining, Modeling, and Recommending'Things' in Social Media, pp. 96–115. Springer (2015)
29. Larrain, S., Trattner, C., Parra, D., Graells-Garrido, E., Nørvåg, K.: Good times bad times: A study on recency effects in collaborative filtering for social tagging. In: Proceedings of the 9th ACM Conference on Recommender Systems, RecSys '15, pp. 269–272 (2015)
30. Lei, C., Liu, D., Li, W., Zha, Z.J., Li, H.: Comparative deep learning of hybrid representations for image recommendations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2545–2553 (2016)
31. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of Machine Learning Research **9**(Nov), 2579–2605 (2008)
32. Macedo, A.Q., Marinho, L.B., Santos, R.L.: Context-aware event recommendation in event-based social networks. In: Proceedings of the 9th ACM Conference on Recommender Systems, pp. 123–130 (2015)
33. Maes, P., et al.: Agents that reduce work and information overload. Communications of the ACM **37**(7), 30–40 (1994)
34. Manning, C.D., Raghavan, P., Schütze, H., et al.: Introduction to information retrieval, vol. 1. Cambridge university press Cambridge (2008)
35. McAuley, J., Targett, C., Shi, Q., Van Den Hengel, A.: Image-based recommendations on styles and substitutes. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 43–52 (2015)
36. Nguyen, A., Yosinski, J., Clune, J.: Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. arXiv preprint arXiv:1602.03616 (2016)
37. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions. Pattern recognition **29**(1), 51–59 (1996)
38. Parra, D., Brusilovsky, P.: User-controllable personalization: A case study with setfusion. International Journal of Human-Computer Studies **78**, 43–67 (2015)
39. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence, pp. 452–461 (2009)
40. Rui, Y., Huang, T.S., Ortega, M., Mehrotra, S.: Relevance feedback: a power tool for interactive content-based image retrieval. IEEE Transactions on circuits and systems for video technology **8**(5), 644–655 (1998)

41. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision **115**(3), 211–252 (2015)
42. San Pedro, J., Siersdorfer, S.: Ranking and classifying attractiveness of photos in folksonomies. In: Proceedings of the 18th International Conference on World Wide Web, WWW '09, pp. 771–780 (2009)
43. Semeraro, G., Lops, P., De Gemmis, M., Musto, C., Narducci, F.: A folksonomy-based recommender system for personalized access to digital artworks. Journal on Computing and Cultural Heritage (JOCCH) **5**(3), 11 (2012)
44. Sharif Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 806–813 (2014)
45. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
46. Tintarev, N., Masthoff, J.: Explaining recommendations: Design and evaluation. In: Recommender Systems Handbook, pp. 353–382. Springer (2015)
47. Trattner, C., Elsweiler, D.: Investigating the healthiness of internet-sourced recipes: implications for meal planning and recommender systems. In: Proceedings of the 26th International Conference on World Wide Web, pp. 489–498 (2017)
48. Verbert, K., Parra, D., Brusilovsky, P., Duval, E.: Visualizing recommendations to support exploration, transparency and controllability. In: Proceedings of the 2013 international conference on Intelligent user interfaces, pp. 351–362 (2013)
49. Weinswig, D.: Art Market Cooling, But Online Sales Booming. https://www.forbes.com/sites/deborahweinswig/2016/05/13/art-market-cooling-but-online-sales-booming/ (2016). [Online; accessed 21-March-2017]