International Journal of Multimedia Information Retrieval (2020) 9:17–30
https://doi.org/10.1007/s13735-019-00189-4

REGULAR PAPER

ContextNet: representation and exploration for painting classification and retrieval in context

Noa Garcia[1] · Benjamin Renoust[1] · Yuta Nakashima[1]

Received: 11 September 2019 / Revised: 13 October 2019 / Accepted: 7 December 2019 / Published online: 21 December 2019

Abstract
In automatic art analysis, models that besides the visual elements of an artwork represent the relationships between the different artistic attributes could be very informative. Those kinds of relationships, however, usually appear in a very subtle way, being extremely difficult to detect with standard convolutional neural networks. In this work, we propose to capture contextual artistic information from fine-art paintings with a specific ContextNet network. As context can be obtained from multiple sources, we explore two modalities of ContextNets: one based on multitask learning and another one based on knowledge graphs. Once the contextual information is obtained, we use it to enhance visual representations computed with a neural network. In this way, we are able to (1) capture information about the content and the style with the visual representations and (2) encode relationships between different artistic attributes with the ContextNet. We evaluate our models on both painting classification and retrieval, and by visualising the resulting embeddings on a knowledge graph, we can confirm that our models represent specific stylistic aspects present in the data.

Keywords Art classification · Multi-modal retrieval · Knowledge graphs · Visualisation · Multitask learning

1 Introduction

This work aims to represent and explore artistic attributes and their relationships in order to improve classification and retrieval of artworks in automatic art analysis. With the large-scale digitisation of art from collections all over the world, computer vision and machine learning have become important tools in the conservation and dissemination of cultural heritage. Some of the most promising work on this direction involves the automatic analysis of paintings, in which computer vision techniques are applied to study the content [12,47] and the style [9,45], or to classify the attributes [35,37] of a specific piece of art.

Automatic analysis of art usually involves the extraction of visual features from digitised artworks by using either hand-crafted [5,28,49] or deep learning techniques [27,34,35,54]. Visual features, specially the ones extracted from convolutional neural networks (CNNs) [24,30,50], have been shown to be very powerful at capturing content [12] and style [9] from paintings, producing outstanding results, for example, on the field of style transfer [45]. However, art specialists rarely analyse artworks as independent and isolated creations, but commonly study paintings within its artistic, historical and social contexts, such as the author influences or the connections between different schools, as illustrated in Fig. 1.

To analyse art from a global perspective, we propose to extract context-aware embeddings from paintings by considering both visual and contextual information. For the visual information, we use a standard convolutional neural network, which successfully encodes the content and the style of each sample. On the other hand, for the contextual information, we propose the use of ContextNets, which capture the relationships between the different artistic attributes that are present in the dataset. As context can be acquired from multiple sources, in this work we explore two modalities of ContextNets.

The first modality is based on multitask learning (MTL). We jointly compute several artistic-related tasks together

✉ Noa Garcia
noagarcia@ids.osaka-u.ac.jp

Benjamin Renoust
renoust@ids.osaka-u.ac.jp

Yuta Nakashima
n-yuta@ids.osaka-u.ac.jp

[1] Institute for Datability Science, Osaka University, Osaka, Japan

Fig. 1 Art as an element in a global context. In Guernica, Pablo Picasso, by means of his own style built upon many artistic influences, such as Cubism or African art, expressed his emotions against war inspired by its historical and political contexts. Image source: www.PabloPicasso.org

(e.g. author classification, type classification, etc.) and obtain an aggregated loss with the losses of each independent task. By optimising a single aggregated loss, the model is enforced to find common elements and capture relationships between the different artistic attributes. In this type of ContextNet, the context is captured from the visual information, as the only input provided to the system is the painting itself.

In the second modality, in contrast, we use a knowledge graph (KG) to learn the different relationships between artistic attributes. We create an art-specific KG by connecting a set of paintings with their artistic-related attributes. Then, node neighbourhoods and positions within the graph are encoded into a vector to represent context. Whereas the MTL model is able to capture relationships occurring at the visual level, the use of KGs offers a more flexible representation of arbitrary relationships, which might not be well-structured and more difficult to detect when considering visual content only. In any case, we incorporate the information obtained with the aforementioned models into the art analysis system.

The two proposed ContextNets are evaluated on the SemArt dataset [20] in four different art classification tasks and in two cross-modal retrieval tasks. We show that, although none of the proposed modalities show a superior performance with respect to the other one in all of the evaluated tasks, ContextNets consistently outperform methods based on visual embeddings only. Furthermore, our previous work on context-aware embeddings [18] is extended by exploring the representations obtained with our ContextNets and confirming the presence of specific stylistic aspects in the clusters of the high-dimensional embedding space.

1.1 Contributions

The contributions of this work can be summarised as follows:

- We propose to use specific networks, different from standard visual representation networks, to capture artistic context in paintings.
- We explore two different modalities of our proposed networks, one based on multitask learning and another one based on knowledge graphs.
- We investigate the resulting context-aware embeddings with a visualisation tool, finding insights on how the different artistic attributes are clustered in different embedding spaces.

2 Related work

2.1 Automatic art analysis

In order to identify specific attributes in paintings, early work in automatic art analysis was focused on representing the visual content of paintings by designing handcrafted feature extraction methods [5,25,28,37,49]. For example, [25] proposed to detect authors by analysing their brushwork using wavelet decompositions [28,49], combined colour, edge, or texture features for author, style, and school classification, and [5,37] used SIFT features [33] to classify paintings into different attributes.

In the last years, deep visual features extracted from CNNs have been repeatedly shown to be very effective in many computer vision tasks, including automatic art analysis [2,20,27,34,35,44,53,54]. At first, deep features were extracted from pre-trained networks and used off-the-shelf for automatic art classification [2,27,44]. Later, deep visual features were shown to obtain better results when fine-tuned using painting images [8,35,47,53,54]. Alternatively, [10–12] explored domain transfer for object and face recognition in paintings, whereas [20] introduced the use of joint visual and textual models to study paintings from a semantic perspective.

So far, most of the proposed methods in automatic art analysis have focused on representing the visual essence of an artwork by capturing style and/or content. However, the study of art is not only about the visual appearance of paintings, but also about their historical, social, and artistic contexts. In this work, we propose to consider both visual and contextual information in art by introducing ContextNet networks. Although the main focus of this work is on painting classification and retrieval, our findings can be easily applied to other artistic areas [39,40].

2.2 Multitask learning

Multitask learning models [6] aim to solve multiple tasks jointly with the hope that the generated generic features are more powerful than task-specific representations. In deep learning approaches, MTL is commonly performed via hard or soft parameter sharing [42]. Whereas in hard parameter sharing [6,48], except by the output layers, parameters are shared between all the tasks, in soft parameter sharing [32,58], each task is defined by its own parameters, which are encouraged to remain similar via regularisation methods.

Following the success of MTL in many computer vision problems, such as object detection and recognition [3,43], object tracking [60], facial landmark detection [61], or facial attribute classification [41], we propose a hard parameter sharing MTL approach for obtaining context-aware embeddings in the domain of art analysis. In our approach, by jointly learning related artistic tasks, the resulting visual representations are enforced to capture relationships and common elements between the different artistic attributes, such as author, school, type, or period, and thus, providing contextual information about each painting. In parallel with our work, Strezoski et al. [52] also show outstanding improvements in an art classification dataset by using MTL strategies, which encourage our claim that context is strongly beneficial in automatic art analysis.

2.3 Knowledge graphs

Knowledge graphs are complex graph structures able to capture non-structured relationships between the data represented in the graph. When KGs are used to add contextual information to a multimedia database, prior work has shown consistent improvements in annotation, classification, and retrieval benchmarks [7,13,15,17,26,36,43,55,59].

To extract contextual information from a KG, one strategy is to encode relationships from visual concepts detected in pictures, forming concept hierarchies [15,43]. Johnson et al. [26] introduced human-generated scene graphs based on descriptions of pictures to improve retrieval tasks, whereas [13] exploited semantic relationships between labels using ConceptNet [51]. Another strategy is to gather labelling information from social media to compute a word-image graph, in which random walks are proposed to extract topological information [59]. Other approaches incorporate the use of external knowledge bases. For example, [17] proposed to improve classifiers with the use of WordNet, [36,38] designed an end-to-end learning pipeline to incorporate large knowledge graphs, such as Visual Genome [29], into classification, and [55] trained image and graph embeddings using WordNet, NELL [4], or NEIL [7].

While related work mostly relies on the use of external knowledge, in our knowledge graph model, we propose to capture contextual information only by processing the data provided with art datasets. As the semantic of art pieces is extremely domain specific, the symbolism that is implied in mythological or religious representations may not benefit from general knowledge. Instead, we leverage on metadata information from art datasets to create a domain-specific knowledge graph, from which we train context embeddings without any task-specific supervision.

3 Multitask learning ContextNet

In the MTL ContextNet, artistic context is obtained by finding visual relationships between common elements in different artistic attributes. To compute context-aware embeddings, the model is trained to learn multiple artistic tasks jointly, so the generated embeddings are enforced to find visual similarities between the different tasks.

Formally, in a multitask learning problem, given T learning tasks, with the training setting for the tth task consisting of N_t training samples and denoted as $\{\mathbf{x}_j^t, y_j^t\}_{j=1}^{N_t}$, where $\mathbf{x}_j^t \in \mathbb{R}^d$ and y_j^t are the jth training sample and its label, respectively, the goal is to optimise:

$$\operatorname*{arg\,min}_{\{\mathbf{w}^t\}_{t=1}^T} \sum_{t=1}^{T} \sum_{j=1}^{N_t} \lambda^t \ell_t(f(\mathbf{x}_j^t; \mathbf{w}^t), y_j^t) \tag{1}$$

where f is a function parameterised by the vector $\mathbf{w}^t$, ℓ_t is the loss function for the tth task, and λ^t, with $\sum_{t=1}^T \lambda_t = 1$, weights the contribution of each task.

In our model, we aim to distinguish between the context-aware information and the task-specific data. We define the function parameters for the tth task as the contribution of two vectors, $\mathbf{w}^t = [\mathbf{w}_g^t; \mathbf{w}_s^t]$, so that f is defined as:

$$f(\mathbf{x}_j^t; \mathbf{w}^t) = f_s(\mathbf{v}_j^t; \mathbf{w}_s^t) \tag{2}$$

where

$$\mathbf{v}_j^t = f_g(\mathbf{x}_j^t; \mathbf{w}_g^t) \tag{3}$$

here f_g is a context-aware function parametrised by $\mathbf{w}_g^t$, f_s is a task-specific function parametrised by $\mathbf{w}_s^t$, and $\mathbf{v}_j^t$ is the jth context-aware embedding generated by task t.

By sharing both the training data and the context-aware parameters across all the tasks as $\mathbf{x}_j^t = \mathbf{x}_j^k$ and $\mathbf{w}_g^t = \mathbf{w}_g^k$ for $j \neq k$, the context-aware embedding $\mathbf{v}_j^t$ is defined as:

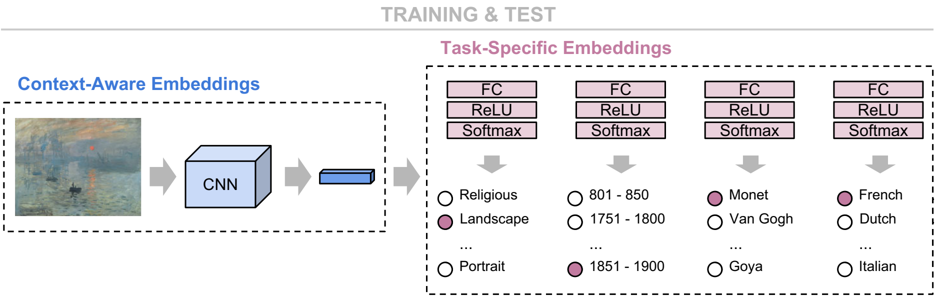
$$\mathbf{v}_j = f_g(\mathbf{x}_j; \mathbf{w}_g) \tag{4}$$

which enforces $\mathbf{v}_j$ to encode $\mathbf{x}_j$ in a generic and non-task-specific representation by identifying patterns and relationships within different tasks. The problem, finally, is formulated as:

$$\operatorname*{arg\,min}_{\mathbf{w}_g, \{\mathbf{w}_s^t\}_{t=1}^T} \sum_{t=1}^{T} \sum_{j=1}^{N} \lambda^t \ell_t(f_s(f_g(\mathbf{x}_j; \mathbf{w}_g); \mathbf{w}_s^t), y_j^t) \tag{5}$$

For solving this optimisation problem, we propose the model in Fig. 2, in which the T learning tasks correspond to multiple artistic classification challenges, such as

Fig. 2 Overview of the multitask learning ContextNet



type, school, timeframe, or author classification. To obtain context-aware embeddings, the context-aware function, f_g, is characterised by ResNet50 [24] after removing the last fully connected layer, whereas the task-specific functions, f_s, are described by a fully connected layer followed by a ReLU nonlinearity. The output of f_g is a 2048-dimensional embedding, which is the input of the task-specific classifiers. Each classifier produces a C_t-dimensional task-specific embedding as output, $\mathbf{z}^t_j$, where C_t is the number of classes in each task. Each tasks is formulated with the cross-entropy loss function as:

$$\ell_t(\mathbf{z}^t_j, y^t_j) = -\log\left(\frac{\exp(\mathbf{z}^t_j[y^t_j])}{\sum_c \exp(\mathbf{z}^t_j[c])}\right) \tag{6}$$

where $\mathbf{z}^t_j = f_s(f_g(\mathbf{x_j}; \mathbf{w}_g); \mathbf{w}^t_s)$.

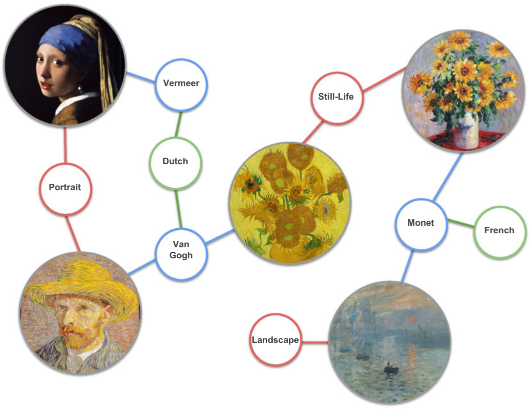


Fig. 3 An example of our artistic KG. Each node corresponds to either a painting or an artistic attribute, whereas edges correspond to existing interconnections

4 Knowledge graph ContextNet

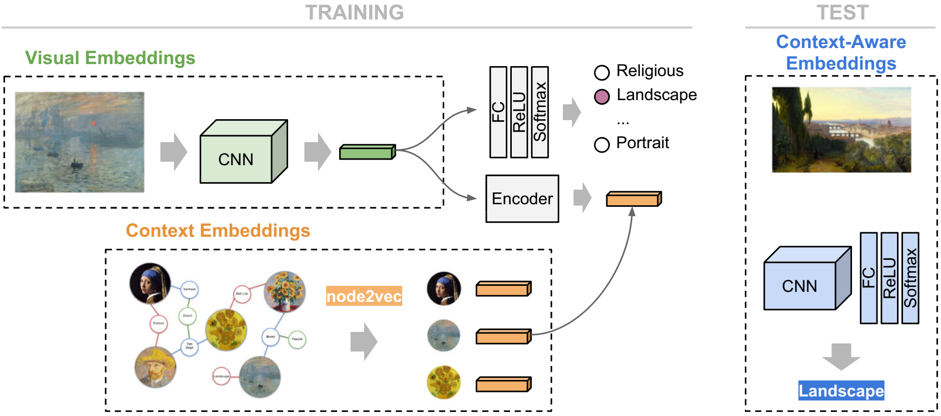
In the MTL ContextNet, contextual information is provided by the painting images themselves by considering the relationships between common elements in the visual appearance of the images when multiple artistic tasks are trained together. In the knowledge graph ContextNet (KGM), in contrast, contextual information is obtained from capturing relationships in an artistic knowledge graph built with non-visual artistic metadata.

4.1 Artistic knowledge graph

A KG is a graph structure, $G = (V, E)$, in which the entities and their relations are represented by a collection of nodes, V, and edges, E, respectively. We use a KG to capture contextual knowledge and similarities in the semantic space formed by the graph, often referred to as homophily [21].

To construct an artistic KG, one strategy is to connect paintings with edges when sharing a common attribute $a \in A$, where A is a collection of artistic attributes. However, the complexity of this approach is expensive, reaching the order of $|V|^2 \times |A|$. Instead, we propose to connect paintings with their attributes in a much sparser manner. We consider multiple types of node: paintings, $P \subseteq V$, which represent the paintings themselves (e.g. Girl with a Pearl Earring), and each family, ψ, of attributes $A_\psi \subseteq V$, which represent artistic concepts (e.g. a type such as Portrait or an author such as Van Gogh). We use the training data from the SemArt dataset [20], which contains 19,244 paintings labelled with the attributes *Author*, *Title*, *Date*, *Technique*, *Type*, *School*, and *Timeframe* to connect edges, $e = (V_p, V_q) \in E$, between painting nodes, V_p, and attribute nodes, $V_q \in A_\psi$, with $\psi \in \{\mathit{Type}, \mathit{Timeframe}, \mathit{Author}\}$, when an attribute exists in a painting. As *School* corresponds to an author's school, we connect an edge, $e = (V_a, V_s) \in E$, between an author, V_a, and a school, V_s. We additionally enrich our graph with three other families of attributes, which are connected to painting nodes. From *Technique*, we extract *Material*, such as oil, and *Support*, such as 210 × 80 cm. Also, by computing the most common n-grams in the titles, with n up to three, we extract keywords from the title of each painting, such as Three Graces. In total, the resulting KG presents 33,148 nodes and 125,506 edges, with 3166 authors, 618 materials, 26 schools, 8899 supports, 22 timeframes, 10 types, and 1163 keyword nodes. An example representation of our artistic graph is shown in Fig. 3.

Fig. 4 Overview of the knowledge graph ContextNet



4.2 Training

At training time, visual and context embeddings are computed from the painting image and from the KG, respectively, and used to optimise the weights of the model. Our training model is depicted in Fig. 4, and each of its parts are detailed below.

Visual embeddings Visual embeddings represent the visual appearance of paintings, containing information about the content and the style of the artwork. To obtain the visual embeddings, we use a ResNet50 [24] without the last fully connected layer.

Context embeddings Context embeddings encode the artistic context of an artwork by extracting data from the KG. For encoding the KG information into a vector representation, we adopt the node2vec model [22] because of its capacity to preserve a trade-off between homophily and structural equivalences, resulting in high performances in node classification tasks [21]. To capture node embeddings, node2vec operates skip-grams over random walks in the KG and associates a vector representing the neighbourhood and the overall position of each node in the graph.

Classifier The classifier takes as input the visual embedding and predicts the artistic attributes contained in the sample painting. We use different kinds of attribute classifiers, such as type, school, timeframe, or author. The classifier is composed of a fully connected layer followed by a ReLU nonlinearity, and its output is used to compute a classification loss using a cross-entropy loss function:

$$\ell_c(\mathbf{z}_j, class_j) = -\log\left(\frac{\exp(\mathbf{z}_j[class_j])}{\sum_i \exp(\mathbf{z}_j[i])}\right) \tag{7}$$

where $\mathbf{z}_j$ and $class_j$ are the output of the classifier and the assigned label of the attribute for the jth training painting, respectively.

Encoder The encoder module, which is composed of a single fully connected layer, is used to project the visual embeddings into the context embedding space. We compute the loss between the projected visual embedding, $\mathbf{p}_j$, and the context embedding, $\mathbf{u}_j$, of the j-training sample with a smooth L1 loss function:

$$\ell_e(\mathbf{p}_j, \mathbf{u}_j) = \frac{1}{n}\sum_i \delta_{ji} \tag{8}$$

where

$$\delta_{ji} = \begin{cases} \frac{1}{2}(p_{ji} - u_{ji})^2, & \text{if } |p_{ji} - u_{ji}| \le 1 \\ |p_{ji} - u_{ji}| - \frac{1}{2}, & \text{otherwise} \end{cases}$$

where p_{ji} and u_{ji} the ith elements in $\mathbf{p}_j$ and $\mathbf{u}_j$, respectively. To train the KGM, we compute the total loss function of the model as a combination of the losses obtained from the classifier and encoder modules:

$$\mathcal{L} = \lambda_c \sum_{j=1}^{N} \ell_c(\mathbf{z}_j, class_j) + \lambda_e \sum_{j=1}^{N} \ell_e(\mathbf{p}_j, \mathbf{u}_j) \tag{9}$$

where λ_c and λ_e are parameters that weight the contribution of the classification and the encoder modules, respectively, and N is the number of training samples.

Whereas the parameters of the context embeddings are learnt without supervision and frozen during the KGM training process, the loss score, $\mathcal{L}$, obtained from Equation (9) is backpropagated through the weights of the visual embedding module. This enforces ResNet50 to compute embeddings that are meaningful for artistic classification by decreasing ℓ_c, while incorporating contextual information from the knowledge graph by minimising ℓ_e.

4.3 ContextNet at test time

At test time, to obtain context-aware embeddings from unseen test samples, painting images are fed into the fine-tuned ResNet50 model. As context embeddings computed directly from the KG cannot be obtained for samples that are not contained as a node, the context embedding and the encoder modules are removed from the test model (Fig. 4).

However, the ResNet50 network has been enforced during the training process (1) to capture relevant visual information to predict artistic attributes and (2) to incorporate contextual data from the KG in the visual embeddings. Therefore, the output embeddings from the fine-tuned ResNet50 are, indeed, context-aware embeddings.

5 Art classification evaluation

We evaluated the two proposed ContextNets in multiple art classification tasks, including author identification and type classification.

5.1 Implementation details

In both of our proposed models, painting images are encoded into a vector representation by using ResNet50 [24] without the last fully connected layer. ReNet50 is initialised with its standard pre-trained weights for image classification, whereas the weights from the rest of the layers are initialised randomly. Images are scaled down to 256 pixels per side and randomly cropped into 224 × 224 patches. At training time, visual data are augmented by randomly flipping images horizontally. The size of the embeddings produced by ResNet50 is 2048, whereas the dimensionality produced by node2vec is 128. We use stochastic gradient descent with a momentum of 0.9 and a learning rate of 0.001 as optimiser. The training is conducted in mini-batches of 28 samples, with a patience of 30 epochs. In the MTL ContextNet, λ_t is set to 0.25 for all the tasks, whereas in the KGM ContextNet, λ_c is set to 0.9 and λ_c to 0.1.

5.2 Evaluation dataset

We use the SemArt dataset [20] in our art classification evaluation. The SemArt dataset is a collection of 21,384 painting images, from which 19,244 are used for training, 1069 for validation, and 1069 for test. Each painting is associated with an artistic comment, and the following attributes are: *Author*, *Title*, *Date*, *Technique*, *Type*, *School* and *Timeframe*. We implement the following four tasks for art classification evaluation.

- **Type classification** Using the attribute *Type*, each painting is classified according to 10 different common types of paintings: *portrait*, *landscape*, *religious*, *study*, *genre*, *still life*, *mythological*, *interior*, *historical* and *other*.
- **School classification** The *School* attribute is used to assign each painting to one of the schools of art that appear at least in ten samples in the training set: *Italian*, *Dutch*, *French*, *Flemish*, *German*, *Spanish*, *English*, *Netherlandish*, *Austrian*, *Hungarian*, *American*, *Danish*, *Swiss*, *Russian*, *Scottish*, *Greek*, *Catalan*, *Bohemian*, *Swedish*, *Irish*, *Norwegian*, *Polish* and *Other*. Paintings with a school different to those are assigned to the class *Unknown*. In total, there are 25 school classes.
- **Timeframe classification** The attribute *Timeframe*, which corresponds to periods of 50 years evenly distributed between 801 and 1900, is used to classify each painting according to its creation date. We only consider timeframes with at least ten paintings in the training set, obtaining a total of 18 classes, which includes an *Unknown* class for timeframes out of the selection.
- **Author identification** The *Author* attribute is used to classify paintings according to 350 different painters. Although the SemArt dataset provides 3281 unique authors, we only consider the ones with at least ten paintings in the training set, including an *Unknown* class for painters not contained in the final selection.

5.3 Baselines

Our models are compared against the following baselines:

- **Pre-trained Networks** VGG16 [50], ResNet50 [24] and Res-Net152 [24] with their pre-trained weights learnt in natural image classification. To adapt the models for art classification, we modified the last fully connected layer to match the number of classes of each task. The weights of the last layer were initialised randomly and fine-tuned during training, whereas the weights of the rest of the network were frozen.
- **Fine-tuned Networks** VGG16 [50], ResNet50 [24] and Res-Net152 [24] networks were fine-tuned for each art classification task. As in the pre-trained models, the last layer was modified to match the number of classes in each task.
- **ResNet50+Attributes** The output of each fine-tuned classification model from above was concatenated to the output of a pre-trained ResNet50 network without the last fully connected layer. The result was a high-dimensional embedding representing the visual content of the image and its attribute predictions. The high-dimensional embedding was input into a last fully connected layer with ReLU to predict the attribute of interest. Only the weights from the pre-trained ResNet50 and the last layer were fine-tuned, whereas the weights of the attribute classifiers were frozen.
- **ResNet50+Captions** For each painting, we generated a caption using the captioning model from [57]. Captions

Table 1 Art classification results on SemArt dataset

Method	Type	School	TF	Author
VGG16 pre-trained	0.706	0.502	0.418	0.482
ResNet50 pre-trained	0.726	0.557	0.456	0.500
ResNet152 pre-trained	0.740	0.540	0.454	0.489
VGG16 fine-tuned	0.768	0.616	0.559	0.520
ResNet50 fine-tuned	0.765	0.655	0.604	0.515
ResNet152 fine-tuned	0.790	0.653	0.598	0.573
ResNet50+Attributes	0.785	0.667	0.599	0.561
ResNet50+Captions	0.799	0.649	0.598	0.607
MTL context-aware	0.791	**0.691**	**0.632**	0.603
KGM context-aware	**0.815**	0.671	0.613	**0.615**

Bold values indicate the best result

were represented by a multi-hot vector with a vocabulary size of 5000 and encoded into a 512-dimensional embedding with a fully connected layer followed by an hyperbolic tangent or tanh activation. The caption embeddings were then concatenated to the output of a ResNet50 network without the last fully connected layer. The concatenated vector was fed into a fully connected layer with ReLU to obtain the prediction.

5.4 Results analysis

We measured classification performance in terms of accuracy, i.e. the ratio of correctly classified samples over the total number of samples. Results are provided in Table 1. In every task, the best accuracy was obtained when a ContextNet, MTL or KGM, was used. The MTL ContextNet performed slightly better than the KGM in *School* and *Timeframe* tasks, whereas the KGM was the best in classifying *Type* and *Author* attributes. Unsurprisingly, the pre-trained models obtained the worst results among all the baselines, as they do not present enough discriminative power in the domain of art. Also, there was a clear improvement with respect to pre-trained baselines when the networks were fine-tuned, as already noted in previous work [35,47,53,54]. On the other hand, adding attributes or captions to the visual representations seemed to improve the accuracy, although not in all the scenarios, e.g. *Timeframe* was better classified with the fine-tuned ResNet50 model than with ResNet50+Attributes or ResNet50+Captions, whereas *School* was better classified with the fine-tuned ResNet50 than with ResNet50+Captions. This suggests that informing the model with extra information is beneficial. When the data used to inform the model were from a ContextNet, accuracy was boosted, with improvements ranging from 3.16 to 7.3% with respect to fine-tuned networks and from 1.32 to 5.5% with respect to ResNet50+Attributes and ResNet50+Captions.

View of Florence from Villa San Firenze, near San Miniato

This view of Florence is one of a number of views by Lear based upon on the spot sketches he produced in 1861.

Ships Moored Off a Rocky Coastline

This landscape depicts ships moored off a rocky coastline with fishermen unloading their catch.

Water Carriers

This painting was inspired by the painter's travels in Italy. The costume of the two girls and the landscape suggests the Amalfi coast, or Capri as the setting of the scene.

Still-Life

This painting depicts a still-life of grapes, cherries, peaches and other fruit in a basket, with a rose and a dragonfly on a stone ledge.

Fig. 5 Examples of the SemArt dataset

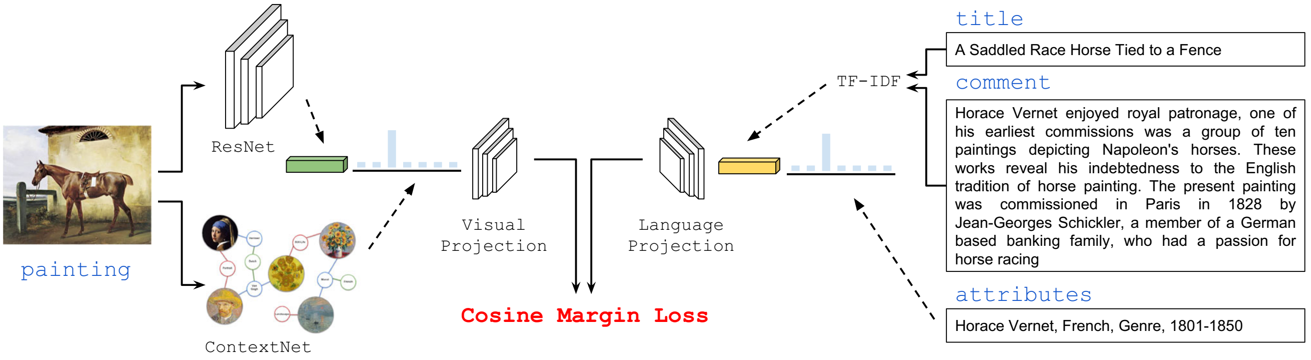
6 Art retrieval evaluation

We additionally evaluated the our ContextNets on art retrieval problems by incorporating context-aware embeddings into a cross-modal retrieval algorithm.

6.1 Implementation details

As evaluation protocol, we used the SemArt dataset and its proposed Text2Art challenge, which consists of two cross-modal retrieval tasks: text-to-image and image-to-text. In text-to-image retrieval, given an artistic comment and its attributes, the goal is to find the correct painting within all the test paintings in the dataset. Similarly, in image-to-text retrieval, given a sample painting, the goal is to find the correct comment. Examples of paintings and their comments in the dataset can be seen in Fig. 5. We incorporate our ContextNets in a cross-modal retrieval system as shown in Fig. 6 and described below [19].

Visual encoder Painting images are scaled down to 256 pixels per side and randomly cropped into 224×224 patches. Then, paintings are fed into ResNet50, initialised with its standard pre-trained weights, to obtain a 1000-dimensional vector, $\mathbf{h}_{\text{cnn}}$, from the last convolutional layer. At the same time, paintings are fed into a ContextNet classifier to obtain a c-dimensional vector, $\mathbf{h}_{\text{att}}$, containing the predicted attributes, with c being the number of output classes in the classifier. The final visual representation, $\mathbf{h}$, is then computed as $\mathbf{h} = \mathbf{h}_{\text{cnn}} \oplus \mathbf{h}_{\text{att}}$, where $\oplus$ is concatenation.



A Saddled Race Horse Tied to a Fence

Horace Vernet enjoyed royal patronage, one of his earliest commissions was a group of ten paintings depicting Napoleon's horses. These works reveal his indebtedness to the English tradition of horse painting. The present painting was commissioned in Paris in 1828 by Jean-Georges Schickler, a member of a German based banking family, who had a passion for horse racing

Horace Vernet, French, Genre, 1801-1850

Fig. 6 ContextNets for cross-modal retrieval in art

Comment and attribute encoder We encode each comment as a term frequency–inverse document frequency (tf–idf) vector, $\mathbf{q}_{\text{com}}$, using a vocabulary of size 9708, which is built with the alphabetic words that appear at least ten times in the training set. We encode titles as another tf–idf vector, $\mathbf{q}_{\text{tit}}$, with a vocabulary of size 9092, which is built with the alphabetic words that appear in the titles of the training set. Additionally, we encode *Type*, *School*, *Timeframe*, or *Author* attributes using a c-dimensional one-hot vector, $\mathbf{q}_{\text{att}}$, with c being the number of classes in each attribute. The final joint comment and attributes representation, $\mathbf{q}$, is computed as $\mathbf{q} = \mathbf{q}_{\text{com}} \oplus \mathbf{q}_{\text{tit}} \oplus \mathbf{q}_{\text{att}}$.

Cross-modal projections To compute similarities between cross-modal data, the visual representation, $\mathbf{h}$, and the joint comment and attributes representation, $\mathbf{q}$, are projected into a common 128-dimensional space using the nonlinear functions f_h and f_q, respectively. The nonlinear functions are implemented with a fully connected layer followed by tanh activation and a ℓ_2-normalisation. Once projected into the common space, elements are retrieved according to their cosine similarity.

The weights of the retrieval model, except from the ContextNet which is frozen, are trained using both positive (i.e. matching) and negative (i.e. non-matching) pairs of samples with the cosine margin loss function:

$$\mathcal{L}(\mathbf{h}_k, \mathbf{q}_j) = \begin{cases} 1 - \text{sim}(f_h(\mathbf{h}_k), f_q(\mathbf{q}_j)), & \text{if } k = j \\ \max(0, \text{sim}(f_h(\mathbf{h}_k), f_q(\mathbf{q}_j)) - \Delta), & \text{if } k \neq j \end{cases} \tag{10}$$

where sim is the cosine similarity between two vectors and $\Delta = 0.1$ is the margin. We use Adam optimiser with learning rate 0.0001.

6.2 Results analysis

Results are reported as median rank (MR) and recall rate at K (R@K), with K being 1, 5, and 10. MR is the value separating the higher half of the relevant ranking position amount all samples, i.e. the lower the better, whereas R@K is the rate of samples for which its relevant image is in the top K positions of the ranking, i.e. the higher the better.

We report results of the proposed cross-modal retrieval model using the following ContextNets: MTL-Type, MTL-Timeframe, MTL-School, MTL-Author, KGM-Type, KGM-School, KGM-Timeframe, and KGM-Author, in which only the specified attribute is used. As a baseline of the proposed model, results when using fine-tuned ResNet152 instead of a ContextNet are also reported. Our methods are compared against previous work: CML [20], which encodes comments and titles without attribute information, and AMD [20], in which attributes are used at training time to learn the visual and textual projections. CML* is a reimplementation of CML with slightly better results.

Results are summarised in Table 2. The KGM-Author model obtained the best results, improving previous state of the art, CML*, by a 37.24% in average. When comparing ContextNets, in agreement with classification results (Table 1), MTL performed better than KGM when using *School*, whereas KGM was the best in *Type* and *Author* attributes. We also noted that concatenating the output of an attribute classifier as proposed (ResNet152, MTL, and KGM models) improved results considerably with respect to AMD. However, we observed a big difference in performance when using the different attributes, being *Author* and *Type* the best and the worst ones, respectively. A possible explanation for this phenomenon may lay in the difference on the number of classes of each attribute.

Finally, our best model, KGM-Author, was further compared against human evaluators. In the easy set-up, evaluators were shown an artistic comment, a title, and the attributes *Author*, *Type*, *School*, and *Timeframe* and were asked to choose the most appropriate painting from a pool of ten random images. In the difficult set-up, however, instead of random paintings, the images shown shared the same attribute *Type*. Results are provided in Table 3. Our model reached values closer to human accuracy than previous work,

Table 2 Results on the Text2Art challenge

Model	Text-to-image				Image-to-text			
	R@1	R@5	R@10	MR	R@1	R@5	R@10	MR
CML	0.144	0.332	0.454	14	0.138	0.327	0.457	14
CML*	0.164	0.384	0.505	10	0.162	0.366	0.479	12
AMD								
Type	0.114	0.304	0.398	17	0.125	0.280	0.398	16
School	0.103	0.283	0.401	19	0.118	0.298	0.423	16
TF	0.117	0.297	0.389	20	0.123	0.298	0.413	17
Author	0.131	0.303	0.418	17	0.120	0.302	0.428	16
Res152								
Type	0.178	0.383	0.525	9	0.165	0.364	0.491	11
School	0.192	0.386	0.507	10	0.163	0.364	0.484	12
TF	0.127	0.322	0.432	18	0.130	0.336	0.444	16
Author	0.236	0.451	0.572	7	0.204	0.440	0.535	8
MTL								
Type	0.145	0.358	0.474	12	0.150	0.350	0.475	12
School	0.196	0.428	0.536	8	0.172	0.396	0.520	10
TF	0.171	0.394	0.525	9	0.138	0.353	0.466	12
Author	0.232	0.452	0.567	7	0.206	0.431	0.535	9
KGM								
Type	0.152	0.367	0.506	10	0.147	0.367	0.507	10
School	0.162	0.371	0.483	12	0.156	0.355	0.483	11
TF	0.175	0.399	0.506	10	0.148	0.360	0.472	12
Author	**0.247**	**0.477**	**0.581**	**6**	**0.212**	**0.446**	**0.563**	**7**

Bold values indicate the best result

Table 3 Comparison against human evaluation

Model	Land	Relig	Myth	Genre	Port	Total
Easy set						
CCA [20]	0.708	0.609	0.571	0.714	0.615	0.650
CML [20]	**0.917**	0.683	0.714	**1**	0.538	0.750
KGM Author	0.875	**0.805**	**0.857**	0.857	**0.846**	**0.830**
Human	0.918	0.795	0.864	1	1	0.889
Difficult set						
CCA [20]	**0.600**	0.525	0.400	0.300	0.400	0.470
CML [20]	0.500	**0.875**	0.600	0.200	0.500	0.620
KGM Author	**0.600**	0.825	**0.700**	**0.400**	**0.650**	**0.680**
Human	0.579	0.744	0.714	0.720	0.674	0.714

Bold values indicate the best result

outperforming CML by a 10.67% in the easy task and a 9.67% in the difficult task.

7 Discussion and visualisation

To further understand the quality of our results, we investigate the ability of ContextNets to discern between different contextual cues. We additionally explore the generated embedding space using the knowledge graph as a visualisation tool.

7.1 Separability of embeddings

We study how well context is captured in different types of embeddings by analysing the separability of artistic attributes in clusters. To estimate the separability between clusters, we applied the Davies–Bouldin index [14], Q, which measures a trade-off between dispersion, S_i, and separation, D_{ij}, of the clusters i and j:

$$Q = \frac{1}{k}\sum_{i=1}^{k}\left(\max_{i \neq j}\left(\frac{S_i + S_j}{D_{ij}}\right)\right) \tag{11}$$

where k is the number clusters, and S_i and D_{ij} are computed as:

$$S_i = \left(\frac{1}{|C_i|}\sum_{\mathbf{x}\in C_i}\|\mathbf{x} - A_i\|^p\right)^{1/p} \qquad D_{ij} = \|A_i - A_j\|_p$$

where A_i the centroid of cluster i of element $\mathbf{x} \in C_i$ computed using the ℓ_p distance, and $|C_i|$ the number of elements in C_i.

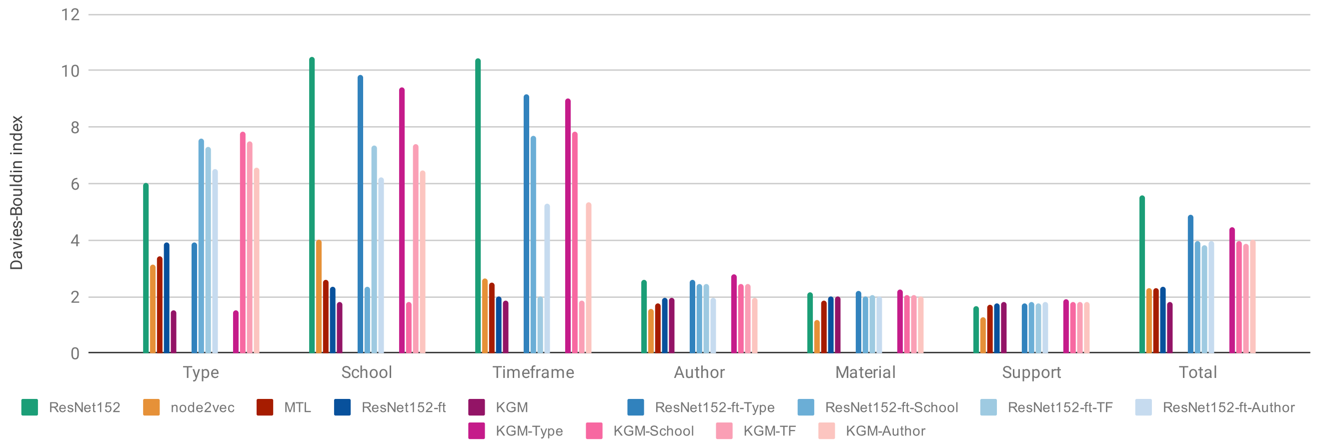
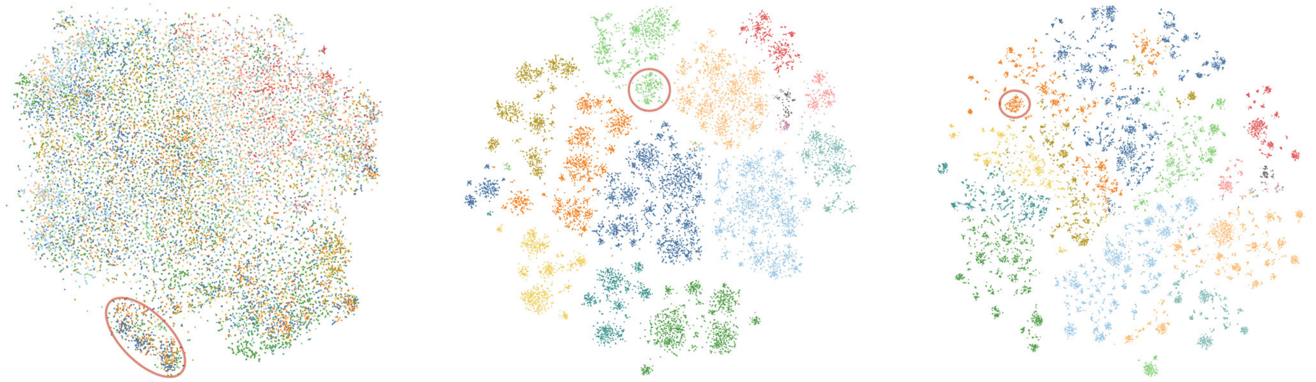


Fig. 7 Davies–Bouldin index for each different attribute. The blue and red groups correspond to single task of ResNet152-ft and KGM, respectively. Their best results are reported in both ResNet152-ft and KGM columns of the first group (colour figure online)



(a) ResNet152 pre-trained **(b)** node2vec **(c)** MTL

Fig. 8 Embeddings of paintings projected in Tulip [1] using t-SNE [56]. Each node is a painting, and the colouring is mapped to the *Timeframe* attribute. There is a good separability of *Timeframe* values in the node2vec and MTL, as opposed to ResNet152. Each red circled area corresponds to its respective cluster selected for inspection in Fig. 10 (colour figure online)

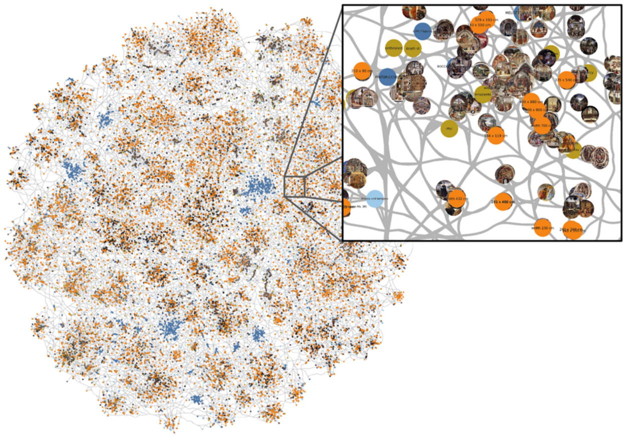
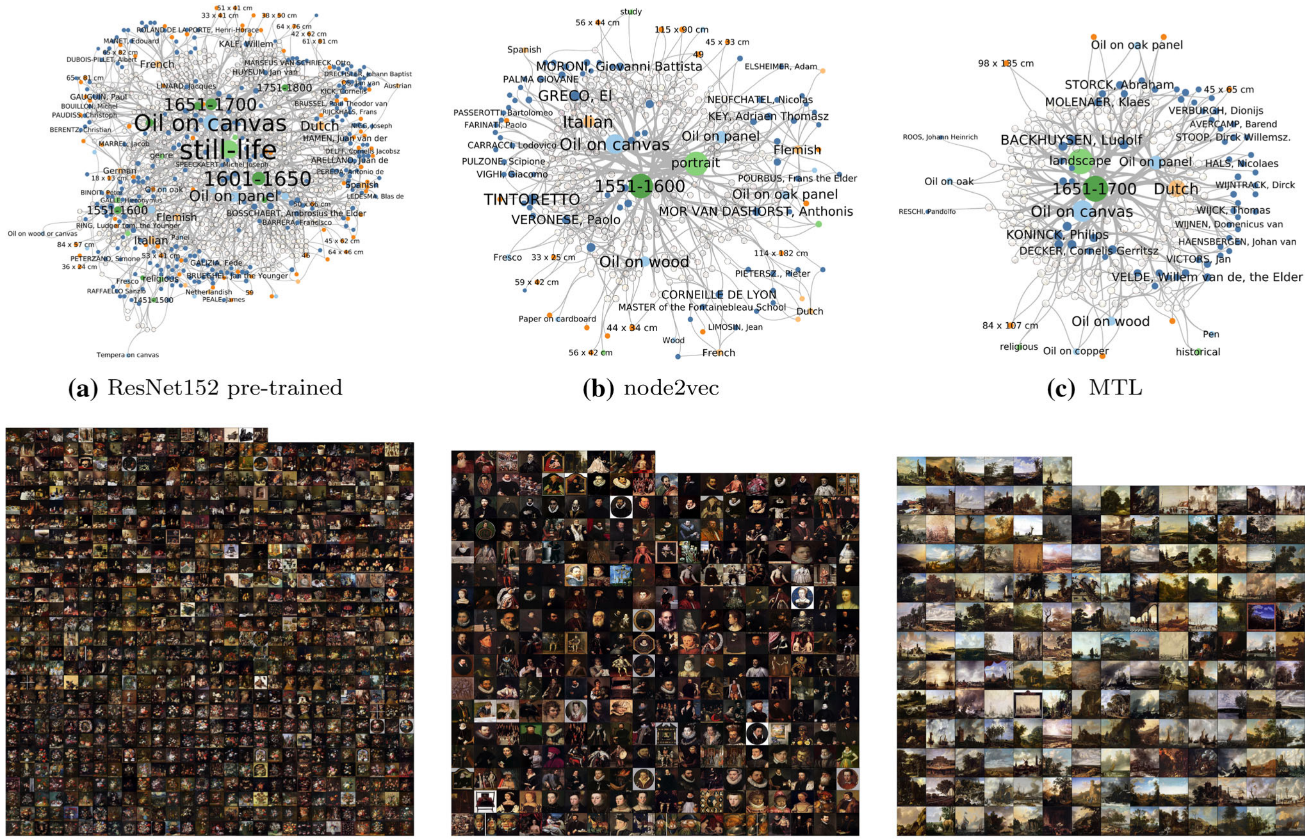


Fig. 9 The overview of the knowledge graph visualised

To compare the different settings, we used the samples from the training set and we applied Q with $p = 2$ to multiple types of embeddings on different attributes, as reported in Fig. 7. When compared on the same task, the smaller value of Q, the better the cluster separation tends to be. We used *Type*, *School*, *Timeframe*, and *Author* attributes to compare performances between models. We also included the derived *Material* and *Support* attributes, for which none of our models was fine-tuned. Along with *Author*, these new attributes have the highest dispersion due to their large number of classes, showing the lowest Q values.

The compared embeddings are detailed in Fig. 7. The pre-trained ResNet152 baseline (in green) shows consistently the worst results in most categories, whereas the node2vec baseline trained on our KG (in orange) shows a good trade-off between categories and the best performance on the most complex attributes *Author*, *Material* and *Support*. On average, KGM (in purple) performs the best due to its high quality on each of the *Type*, *School*, and *Timeframe* attributes for which it has been trained. On average, the MTL (in red) shows a comparable performance to the multiple single-task fine-tuned ResNet152 (in blue).



(a) ResNet152 pre-trained **(b)** node2vec **(c)** MTL

(d) ResNet152 pre-trained **(e)** node2vec **(f)** MTL

Fig. 10 Selected cluster for each different embeddings. Top. Each cluster has been enriched with a knowledge graph and redrawn accordingly. The colour encoding is the following: in dark green, time periods; in light green, type of paintings; in dark blue, author; in light blue: material; in dark orange, support; in light orange, school. Bottom. The list paintings thumbnails for each cluster (colour figure online)

These results rule in favour of the added value that contextual knowledge brought by the KG improves overall performances. We may further confirm this intuition from the 2D-projected embeddings in Fig. 8: while the space represented by pre-trained ResNet152 applied to art does not show any convincing separability, the subspace formed by paintings in the node2vec embeddings shows clear separability and sub-densities. MTL does display such a structure, while being much more fractioned.

7.2 Knowledge graph visualisation

We further investigate the content of these clusters and how they capture abstract concepts of art by using the knowledge graph as a visualisation tool. An overview of the knowledge graph is given in Fig. 9.

We inspect one cluster—i.e. a density in the projected space—per each of the embeddings in Fig. 8. To identify such densities, we first apply a DBScan [46] clustering from the 2D projections.[1] We obtain 10 clusters for ResNet 152 pre-trained, 106 clusters for node2vec, and 285 clusters for MTL. We further rank the top 10 clusters for each type of embeddings based on the averaged pairwise Euclidean distance of their content, with a minimum size of 100 paintings per cluster. Then, we arbitrarily picked one cluster per type of embedding based on its size and visual appeal.

To explore each cluster, we construct the knowledge subgraph induced by all the paintings contained in the selected cluster. To reduce the visual clutter, we remove all the knowledge graph nodes of degree 1.[2] In these mini knowledge graphs, the degree shows the influence of a node in the cluster. We thus mapped their degree on the node size of each node and computed a force-directed layout [23] and then removed overlap [16]. We further used edge bundling to remove the visual clutter induced by too many edges [31]. Results are shown in Fig. 10, using Tulip [1].

[1] We use the same parameters for all settings: Euclidean distance, with at least 10 sample points in a cluster, with a maximum distance of 2

[2] Degree being the number of edges connected to a node.

Table 4 Top degree nodes for each embeddings

ResNet		node2vec		MTL	
Node	Degree	Node	Degree	Node	Degree
Still life	707	1551–1600	297	1651–1700	174
Oil on canvas	463	portrait	287	landscape	167
1601–1650	321	Oil on canvas	174	Oil on canvas	112
1651–1700	210	Oil on panel	45	Dutch	63
Oil on panel	139	Italian	40	Oil on panel	34
Dutch	93	Oil on wood	28	BACKHUYSEN, Ludolf	11
1701–1750	63	TINTORETTO	27	POST, Frans	10
1851–1900	60	GRECO, El	26	KONINCK, Philips	10
1551–1600	53	ARCIMBOLDO, Giuseppe	17	Oil on wood	9
Italian	50	Oil on oak panel	16	VELDE, Adriaen van de	9
Oil on wood	49	MORONI, Giovanni Battista	14	CAPPELLE, Jan van de	9
Oil on oak panel	46	Flemish	14	MOUCHERON, Frederick de	9
Flemish	41	VERONESE, Paolo	14	PYNACKER, Adam	8
French	34	MOR VAN DASHORST, Anthonis	14	WYNANTS, Jan	8

Following the selected clusters, we obtained 774 paintings, 261 authors, 83 supports, 14 materials, 11 schools, 9 timeframes, and 7 types in ResNet (Fig. 10d); 297 paintings, 74 authors, 18 supports, 9 materials, 8 schools, 1 timeframe, and 4 types in node2vec (Fig. 10e); and 174 paintings, 65 authors, 7 supports, 7 materials, 1 school, 1 timeframe, and 3 types in MTL (Fig. 10f).

The top nodes ranking by degree are reported in Table 4. As we can see, the ResNet cluster concentrates still-life oil paintings mostly from the seventeenth century from many different authors, among which Dutch and Italian painters are well represented. The node2vec cluster focuses almost exclusively on portraits of the second half of the sixteenth century, mostly oil paintings, among which Italian and Flemish painters are well represented. The MTL cluster focuses almost exclusively on landscapes from the seventeenth century, mostly oil paintings, among which the Dutch masters are well represented. The characteristics of the painting type may be easily confirmed from the paintings in Fig. 10, which shows that both MTL- and node2vec-based embeddings well capture not only the timeframe but also more specific stylistic aspects of the dataset (i.e. in combination with type and school).

8 Conclusions

This work proposed to use ContextNets to capture the relationship between artistic attributes in art classification and retrieval. Two modalities of ContextNet were introduced. The first one, based on multitask learning, captures the relationships between visual artistic elements in paintings, whereas the second one, based on knowledge graphs, encodes the interconnections between non-visual artistic attributes. The reported results showed that context-aware embeddings are beneficial in many automatic art analysis problems, improving art classification accuracy by up to a 7.3% with respect to classification baselines. In cross-modal retrieval tasks, our best model outperformed previous work by a 37.24%. We further investigated the clusters obtained from the context-aware embeddings, revealing that similar stylistic attributes were placed close to each other.

Acknowledgements This work was supported by JSPS KAKENHI Grant No. 18H03264.

References

1. Auber D, Archambault D, Bourqui R, Delest M, Dubois J, Lambert A, Mary P, Mathiaut M, Mélançon G, Pinaud B, Renoust B, Vallet J (2018) Tulip 5. In: Alhajj R, Rokne J (eds) Encyclopedia of social network analysis and mining. Springer, New York, pp 1–28
2. Bar Y, Levy N, Wolf L (2014) Classification of artistic styles using binarized features derived from a deep neural network. In: Agapito L, Bronstein M, Rother C (eds) European conference on computer vision workshops. Springer, Cham, pp 71–84

3. Bilen H, Vedaldi A (2016) Integrated perception with recurrent multi-task neural networks. In: Advances in neural information processing systems, p 235–243
4. Carlson A, Betteridge J, Kisiel B, Settles B, Hruschka Jr, E.R, Mitchell T.M (2010) Toward an architecture for never-ending language learning. In: AAAI, vol 5. Atlanta, p 3
5. Carneiro G, da Silva NP, Del Bue A, Costeira JP (2012) Artistic image classification: an analysis on the printart database. In: European conference on computer vision, pp 143–157
6. Caruana R (1997) Multitask learning. Mach Learn 28(1):41–75
7. Chen X, Shrivastava A, Gupta A (2013) Neil: extracting visual knowledge from web data. In: Proceedings of the IEEE international conference on computer vision, pp 1409–1416
8. Chu WT, Wu YL (2018) Image style classification based on learnt deep correlation features. IEEE Trans Multimed 20(9):2491–2502
9. Collomosse J, Bui T, Wilber M.J, Fang C, Jin H (2017) Sketching with style: Visual search with sketches and aesthetic context. In: Proceedings of the IEEE international conference on computer vision, pp 2679–2687
10. Crowley E, Zisserman A (2014) The state of the art: object retrieval in paintings using discriminative regions. In: Proceedings of the British machine vision conference. BMVA Press
11. Crowley EJ, Parkhi OM, Zisserman A (2015) Face painting: querying art with photos. In: BMVC, pp 65–1
12. Crowley E.J, Zisserman A (2016) The art of detection. In: European conference on computer vision. Springer, pp 721–737
13. Cui P, Liu S, Zhu W (2018) General knowledge embedded image representation learning. IEEE Trans Multimed 20(1):198–207
14. Davies DL, Bouldin DW (1979) A cluster separation measure. IEEE Trans Pattern Anal Mach Intell 2:224–227
15. Deng J, Ding N, Jia Y, Frome A, Murphy K, Bengio S, Li, Y, Neven H, Adam H (2014) Large-scale object classification using label relation graphs. In: European conference on computer vision. Springer, pp 48–64
16. Dwyer T, Marriott K, Stuckey P.J (2005) Fast node overlap removal. In: International symposium on graph drawing. Springer, pp 153–164
17. Fergus R, Bernal H, Weiss Y, Torralba A (2010) Semantic label sharing for learning with many categories. In: European conference on computer vision. Springer, pp 762–775
18. Garcia N, Renoust B, Nakashima Y (2019) Context-aware embeddings for automatic art analysis. In: Proceedings of the 2019 on international conference on multimedia retrieval. ACM, pp 25–33
19. Garcia N, Renoust B, Nakashima Y (2019) Understanding art through multi-modal retrieval in paintings. arXiv preprint arXiv:1904.10615
20. Garcia N, Vogiatzis G (2018) How to read paintings: semantic art understanding with multi-modal retrieval. In: Proceedings of the European conference in computer vision workshops
21. Goyal P, Ferrara E (2018) Graph embedding techniques, applications, and performance: a survey. Knowl Based Syst 151:78–94
22. Grover A, Leskovec J (2016) node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 855–864
23. Hachul S, Jünger M (2004) Drawing large graphs with a potential-field-based multilevel algorithm. In: International symposium on graph drawing. Springer, pp 285–295
24. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition
25. Johnson CR, Hendriks E, Berezhnoy IJ, Brevdo E, Hughes SM, Daubechies I, Li J, Postma E, Wang JZ (2008) Image processing for artist identification. IEEE Signal Process Mag 25(4):37–48
26. Johnson J, Krishna R, Stark M, Li L.J, Shamma D, Bernstein M, Fei-Fei L (2015) Image retrieval using scene graphs. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3668–3678
27. Karayev S, Trentacoste M, Han H, Agarwala A, Darrell T, Hertzmann A, Winnemoeller H (2014) Recognizing image style. In: Proceedings of the British machine vision conference. BMVA Press
28. Khan FS, Beigpour S, Van de Weijer J, Felsberg M (2014) Painting-91: a large scale database for computational painting categorization. In: Machine vision and applications
29. Krishna R, Zhu Y, Groth O, Johnson J, Hata K, Kravitz J, Chen S, Kalantidis Y, Li LJ, Shamma DA, Bernstein M, Fei-Fei L (2016) Visual genome: Connecting language and vision using crowdsourced dense image annotations. arXiv:1602.07332
30. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
31. Lambert A, Bourqui R, Auber D (2010) Winding roads: routing edges into bundles. Comput Graph Forum 29(3):853–862
32. Long M, Wang J (2015) Learning multiple tasks with deep relationship networks, vol 3. CoRR, arXiv:abs/1506.02117
33. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110
34. Ma D, Gao F, Bai Y, Lou Y, Wang S, Huang T, Duan LY (2017) From part to whole: who is behind the painting? In: Proceedings of the 2017 ACM on multimedia conference. ACM
35. Mao H, Cheung M, She J (2017) Deepart: learning joint representations of visual arts. In: ACM on multimedia conference
36. Marino K, Salakhutdinov R, Gupta A (2017) The more you know: Using knowledge graphs for image classification. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 20–28
37. Mensink T, Van Gemert J (2014) The rijksmuseum challenge: Museum-centered visual recognition. In: Proceedings of international conference on multimedia retrieval. ACM
38. Miller GA (1995) Wordnet: a lexical database for English. Commun ACM 38(11):39–41
39. Renoust B, Oliveira Franca M, Chan J, Garcia N, Le V, Uesaka A, Nakashima Y, Nagahara H, Wang J, Fujioka Y (2019) Historical and modern features for Buddha statue classification. In: Proceedings of 2019 ACM multimedia conference, SUMAC workshop. Association for Computing Machinery (ACM), pp 1–8
40. Renoust B, Oliveira Franca M, Chan J, Le V, Uesaka A, Nakashima Y, Nagahara H, Wang J, Fujioka Y (2019) Buda.art: a multimodal content-based analysis and retrieval system for Buddha statues. In: Proceedings of 2019 ACM multimedia conference. Association for Computing Machinery (ACM), pp 1–3
41. Rudd EM, Günther M, Boult TE (2016) Moon: a mixed objective optimization network for the recognition of facial attributes. In: European conference on computer vision. Springer, pp 19–35
42. Ruder S (2017) An overview of multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098
43. Salakhutdinov R, Torralba A, Tenenbaum J (2011) Learning to share visual appearance for multiclass object detection. In: 2011 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 1481–1488
44. Saleh B, Elgammal AM (2015) Large-scale classification of fine-art paintings: learning the right metric on the right feature. CoRR
45. Sanakoyeu A, Kotovenko D, Lang S, Ommer B (2018) A style-aware content loss for real-time HD style transfer. In: Proceedings of the European conference on computer vision, vol 2
46. Schubert E, Sander J, Ester M, Kriegel HP, Xu X (2017) Dbscan revisited, revisited: why and how you should (still) use DBSCAN. ACM Trans Database Syst (TODS) 42(3):19
47. Seguin B, Striolo C, Kaplan F et al (2016) Visual link retrieval in a database of paintings. In: Hua G, Jégou H (eds) European conference on computer vision workshops. Springer, Cham, pp 753–767

48. Sener O, Koltun V (2018) Multi-task learning as multi-objective optimization. In: Advances in neural information processing systems, pp 525–536
49. Shamir L, Macura T, Orlov N, Eckley D.M, Goldberg I.G (2010) Impressionism, expressionism, surrealism: automated recognition of painters and schools of art. ACM Trans Appl Percept 6(2)
50. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: International conference on learning representations
51. Speer R, Havasi C (2012) Representing general relational knowledge in conceptnet 5. In: Proceedings of the eighth international conference on language resources and evaluation (LREC-2012), pp 3679–3686
52. Strezoski G, van Noord N, Worring M (2019) Learning task relatedness in multi-task learning for images in context. In: Proceedings of the 2019 on international conference on multimedia retrieval. ACM, pp 78–86
53. Strezoski G, Worring M (2018) Omniart: a large-scale artistic benchmark. ACM Trans Multimed Comput Commun Appl (TOMM) 14(4):88
54. Tan WR, Chan CS, Aguirre HE, Tanaka K (2016) Ceci n'est pas une pipe: a deep convolutional network for fine-art paintings classification. In: ICIP
55. Wang X, Ye Y, Gupta A (2018) Zero-shot recognition via semantic embeddings and knowledge graphs. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6857–6866
56. Wattenberg M, Vigas F, Johnson I (2016) How to use t-sne effectively. Distill. https://doi.org/10.23915/distill.00002. http://distill.pub/2016/misread-tsne
57. Xu K, Ba J, Kiros R, Cho K, Courville A, Salakhudinov R, Zemel R, Bengio Y (2015) Show, attend and tell: neural image caption generation with visual attention. In: International conference on machine learning, pp 2048–2057
58. Yang Y, Hospedales T (2016) Deep multi-task representation learning: a tensor factorisation approach. arXiv preprint arXiv:1605.06391
59. Zhang H, Shang X, Luan H, Wang M, Chua TS (2016) Learning from collective intelligence: feature learning using social images and tags. ACM Trans Multimed Comput Commun Appl., pp 1:1–1:23. https://doi.org/10.1145/2978656
60. Zhang T, Ghanem B, Liu S, Ahuja N (2013) Robust visual tracking via structured multi-task sparse learning. Int J Comput Vis 101(2):367–383
61. Zhang Z, Luo P, Loy CC, Tang X (2014) Facial landmark detection by deep multi-task learning. In: European conference on computer vision. Springer, pp 94–108

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.