Transfer Learning for the Visual Arts: The Multi-Modal Retrieval of Iconclass Codes

NIKOLAY BANAR, University of Antwerp, Belgium
WALTER DAELEMANS, University of Antwerp, Belgium
MIKE KESTEMONT, University of Antwerp, Belgium

Iconclass is an iconographic thesaurus which is widely used in the digital heritage domain to describe subjects depicted in artworks. Each subject is assigned a unique descriptive code, which has a corresponding textual definition. The assignment of Iconclass codes is a challenging task for computational systems, due to the large number of available labels in comparison to the limited amount of training data available. Transfer learning has become a common strategy to overcome such a data shortage. In deep learning, transfer learning consists in fine-tuning the weights of a deep neural network for a downstream task. In this work, we present a deep retrieval framework which can be fully fine-tuned for the task under consideration. Our work is based on a recent approach to this task, which already yielded state-of-the-art performance, although it could not be fully fine-tuned yet. This approach exploits the multi-linguality and multi-modality that is inherent to digital heritage data. Our framework jointly processes multiple input modalities, namely, textual and visual features. We extract the textual features from the artwork titles in multiple languages, whereas the visual features are derived from photographic reproductions of the artworks. The definitions of the Iconclass codes, containing useful textual information, are used as target labels instead of the codes themselves. As our main contribution, we demonstrate that our approach outperforms the state-of-the-art by a large margin. In addition, our approach is superior to the M^3P feature extractor and outperforms the multi-lingual CLIP in most experiments due to the better quality of the visual features. Our out-of-domain and zero-shot experiments show poor results and demonstrate that the Iconclass retrieval remains a challenging task. We make our source code and models publicly available to support heritage institutions in the further enrichment of their digital collections.

CCS Concepts: • **Applied computing** → **Fine arts**; • **Information systems** → *Multilingual and cross-lingual retrieval*; *Image search.*

Additional Key Words and Phrases: Iconclass, Cultural Heritage, Transfer Learning, Deep Learning, Natural Language Processing, Multi-Modal Retrieval, Multi-Lingual Retrieval

INTRODUCTION

Iconography [20, 33] is a branch of art history that seeks to analyze the meaning of artworks, and advance the description and interpretation of them. In contrast to the issues of style and structure, iconography focuses on the content or subjects depicted in artworks. Iconclass [5, 51] is an iconographic classification system that is well-known across the GLAM sector (Galleries, Libraries, Archives and Museums). This resource represents a hierarchical thesaurus of 28,000 codes corresponding to the presence of different subjects (such as people, events and ideas) in artworks. The manual assignment of Iconclass codes requires rare interpretive skills and considerable domain expertise in art history. In addition, the large number of available codes makes the attribution an especially complex and time-consuming task for human annotators.

Authors' addresses: Nikolay Banar, nicolae.banari@uantwerpen.be, University of Antwerp, Lange Winkelstraat 40-42, Antwerp, Belgium; Walter Daelemans, walter.daelemans@uantwerpen.be, University of Antwerp, Lange Winkelstraat 40-42, Antwerp, Belgium; Mike Kestemont, mike.kestemont@uantwerpen.be, University of Antwerp, Lange Winkelstraat 40-42, Antwerp, Belgium.

XXXX-XXXX/2023/3-ART $15.00
https://doi.org/10.1145/3575865

Deep learning [26, 44] increasingly attracts the attention of scholars from the cultural heritage domain, due to its state-of-the-art performance across multiple real-world problems [17]. Iconclass has already drawn attention of the scientific community in the following tasks: neural machine translation [2], image classification [31], object detection [36], image captioning [7, 8], and information retrieval [3]. In this work, we aim to automate the assignment of Iconclass codes using the latest advances in deep learning. Deep learning (or machine learning) offers several general ways to tackle this problem: classification, object detection, and information retrieval. In classification, a model aims to predict an Iconclass code of an object from a set of predefined concepts, based on a visual reproduction of an artwork [31] and/or textual metadata. In addition to the prediction of the Iconclass codes, an object detector can be trained to find the bounding box of a concept in an image [36]. We tackle this problem from the angle of information retrieval, as in related prior work [3]. In this case, a retrieval model computes a similarity score between a source artwork object and all available Iconclass codes. The higher similarity, the better the target Iconclass code corresponds to the source artwork object. We use Dutch and English titles, as well as visual reproductions of artworks as source objects; the definitions of Iconclass codes (instead of the actual codes) are used as target labels. Such a retrieval framework is able to provide the top-K similar Iconclass codes for a given artwork, which can significantly speed up the annotation process for human experts. In addition, it enables us to predict unseen Iconclass codes (which were not encountered in the training phase), and to apply the trained retrieval models out of the box to other, unseen iconographic thesauri with a similar structure.

The training process of deep learning models generally requires large datasets, such as ImageNet [25] or MS-COCO [28]. However, the available datasets in the digital heritage domain are typically much smaller; additionally, artworks are considerably different from real-world, photorealistic images [12–14, 42]. First, the annotation ontologies used for these image datasets do not correspond to the iconographic ones, which may require additional context and knowledge about the subjects under scrutiny. Second, the natural images from the general domain are biased towards photorealism in comparison to the visual arts. The visual arts can depict distorted or fictional realities and focus on less realistic representations. The common remedy in this case is the application of transfer learning [41], an approach that consists in fine-tuning a deep learning model on a task-specific dataset, after the model has been pretrained on a more general dataset. In this situation, the model adapts the knowledge learnt from the general domain for the new, more specific task. Here, we build on the general observation that fine-tuning typically boosts the performance of deep learning models in the face of sparse data. Another challenge, commonly heard in natural language processing research, is the over-representation of the English language in the training material [19], which contributes to the comparative lack of efficient models for less-resourced languages. The digital heritage domain is inherently multi-lingual and multi-modal, and it requires appropriate instruments, which we aim to develop in our work.

In this paper, we present a retrieval framework that takes explicit advantage of the notion of fine-tuning. Our work is based on previous research [3] demonstrating the advantages of multi-modality and multi-linguality to tackle the assignment of Iconclass codes to visual artworks. However, this previous work did not explicitly consider the transfer learning paradigm. We convincingly outperform the previous state-of-the-art model by implementing this important feature. In addition, our model outperforms the multi-lingual CLIP [38] in most experiments. The evaluation supporting this claim is conducted on the same dataset as in the previous work, on data extracted from the database of the Netherlands Institute for Art History. In addition, we use an external dataset for evaluation (as well for zero-shot evaluation) from the well-known digital collection of the Rijksmuseum in Amsterdam. To stimulate the analysis of digital heritage metadata, we make the reference implementation of our framework freely available online, for replication and reuse purposes. [1] Below, we adhere to the following structure for the paper. Firstly, we discuss related work and present our approach in greater detail. Next on,

[1] https://github.com/nikolay-banar/iconize

we describe the datasets under consideration and our experimental settings. Finally, we present our results and discuss our contributions to guide future work.

RELATED WORK

In the recent past, deep multi-modal (or vision-language) models have achieved impressive results in many tasks, such as image captioning, cross-modal retrieval, and visual question answering. In the general multi-modal domain, there exists a common tendency to use BERT [15]-based models in combination with Faster-RCNN [40]. BERT is a deep model for natural language processing, which achieved state-of-the-art results in many tasks, and Faster-RCNN is a widely used object detector. The SAEM [54] network implements a two-stream model that encodes both modalities separately. Visual features precomputed by Faster-RCNN are processed by a single-layer Transformer [15], and textual features extracted from BERT are mapped by one-dimensional convolutions. The latter works aim to develop pretrained task-agnostic representations for multi-modal problems and, generally, are based on the following pretraining tasks: masked language modeling and masked region modeling. ViLBERT [29] and LXMERT [48] utilize a cross-modal Transformer to combine modalities encoded by two separate uni-modal Transformers. In both cases, the precomputed visual features are extracted by a pretrained Faster-RCNN. Another branch of the recent work (VL-BERT [47], Unicoder-VL [27], UNITER [9]) focuses on simultaneous encoding of vision and language modalities. However, these works focus mostly on English, which is not suitable for the multi-lingual digital heritage domain. Only the most recent works consider multi-linguality. Fei et al. [16] adapted VL-BERT for the multi-lingual scenario, additionally combining it with the tasks of cross-modal text recovery, cross-lingual text recovery, and translation language modeling. M^{3}P [32] learns multi-linguality and multi-modality independently by introducing multi-modal code-switched training, which is applied to a BERT-based model with visual features extracted by Faster-RCNN. UC^2[57] augments existing English datasets with other languages using machine translation, which allows to capture alignment between different languages through shared visual representations. Additionally, they propose two pretraining tasks of masked region-to-token modeling and visual translation language modeling to efficiently use the translated datasets. Most of the methods use precomputed image features from deep models trained on the general domain (without further fine-tuning), which may lead to a sub-par performance for the problems related to the processing of artwork [43]. The multi-lingual implementation [2] of CLIP (MCLIP) [38] does not suffer from this disadvantage, since the textual and visual branches of this model can be fully fine-tuned for a specific task. The model is trained to match captions with the corresponding images. We use this model and features from M^{3}P [32] as additional baselines in our work.

Multi-modal retrieval has recently attracted the attention of scholars working on the computational analysis of artworks [6]. The first efforts were mainly focused on constructing the training material needed for multi-modal retrieval models. The SemArt dataset [18] provides a set of visual artworks linked to textual comments, which enables an application of the retrieval models in the multi-modal scenario. In addition, the authors of the dataset introduce several retrieval models to encode images and text into a shared semantic space. In the best model, visual features are extracted through a pretrained ResNet [21], which is a deep convolutional network architecture for classification, and textual features are obtained through tf-idf. Next on, the features are mapped into a joint semantic space by a feed-forward neural network. Another study presents the BibleVSA dataset [4], which contains illustrations aligned with textual commentaries. To tackle the dataset, the authors investigate supervised and semi-supervised approaches for mapping the images and their corresponding commentaries into a joint semantic space. This work additionally employs the transfer learning paradigm by fine-tuning pretrained networks. They achieve the best result in the supervised manner by training a GRU network [10], a deep recurrent network for natural language processing, with pretrained word embeddings and fine-tuning

[2] https://github.com/FreddeFrallan/Multilingual-CLIP

VGG-19 [45], a pretrained deep convolutional network. In the next work [46], the authors introduce the Artpedia dataset containing images with visual and contextual descriptions. In addition, the authors propose a retrieval model that encodes image regions by the Faster-RCNN object detector and words from the descriptions through a bi-directional GRU network and pretrained word embeddings. Next on, the encoded regions and words are combined through a cross-attention mechanism. In comparison to the previous work, this work demonstrates the advantage of the object detector Faster-RCNN over the classification models for visual encoding in the artwork retrieval. Finally, the authors in [11] extend their semi-supervised approach [4] to tackle the lack of labelled data in the artistic domain. However, none of the methods presented in this section have made use of the latest advances in multi-modal retrieval, which primarily consist in the usage of BERT-based models.

To the best of our knowledge, only one deep model has been developed specifically for the retrieval of Iconclass codes [3] and that also forms the basis for our present work for that reason. The model generally acknowledges the characteristic properties of the digital heritage domain, namely, multi-modality and multi-linguality. However, the main disadvantage of this precursor is lack of fine-tuning as a feature. The model is based on the SAEM framework, which uses precomputed visual and textual representations. We refer to the previously developed model for the retrieval of Iconclass codes as multi-modal SAEM (MSAEM). In this work, we close the gap by introducing the support of fine-tuning (FMSAEM) to the previous state-of-the-art model [3], which has lead to a large boost of performance.

METHODS

In this section, we present our framework in more detail (see Figure 1). The framework consists of two uni-modal branches, which process visual and textual modalities separately, and a multi-modal combiner, which combines the preprocessed modalities. The structure of the framework is the same as in the previous work [3], but with the support of fine-tuning. Below, we describe each part of the framework in greater detail.

Image Branch

The image branch yields a representation of the visual features from an arbitrary image. The branch consists of two main parts, namely, a bottom-up-attention mechanism [1] and a self-attention layer [50]. The bottom-up-attention mechanism detects salient regions in the image and encodes them into vector representations, which later are processed by the self-attention layer to encode complex relations between salient regions. The bottom-up-attention mechanism is proved to be an effective approach in the general image analysis as well as in artwork analysis specifically [46]. The bottom-up-attention mechanism offers a more efficient strategy for feature extraction in comparison to convolutional neural networks, which has been widely used in artwork analysis [4, 18]. The latter approaches construct representations from equal-size spatial fragments, which might contain redundant information. On the contrary, the bottom-up-attention mechanism is able to extract only the most important image regions.

The bottom-up-attention mechanism implements the Faster R-CNN model [40] with a ResNet-101 backbone [21]. The object detector [56] is implemented in PyTorch [34] and pretrained on Visual Genome [24]. The weights of this deep network are publicly available[3]. The Faster R-CNN model requires ground-truth bounding boxes to calculate the loss function for fine-tuning. However, such bounding boxes are difficult to annotate and our datasets do not contain them. We borrow a previously suggested solution from a related paper [47], which consists in precomputing the bounding boxes for each region-of-interest (RoI) using the same pretrained Faster R-CNN model. We use the ResNet-101 backbone to build a feature map from an arbitrary image. Next on, the RoI pooling layer extracts RoIs of different sizes which correspond to the precomputed bounding boxes. Then,

[3]https://github.com/shilrley6/Faster-R-CNN-with-model-pretrained-on-Visual-Genome

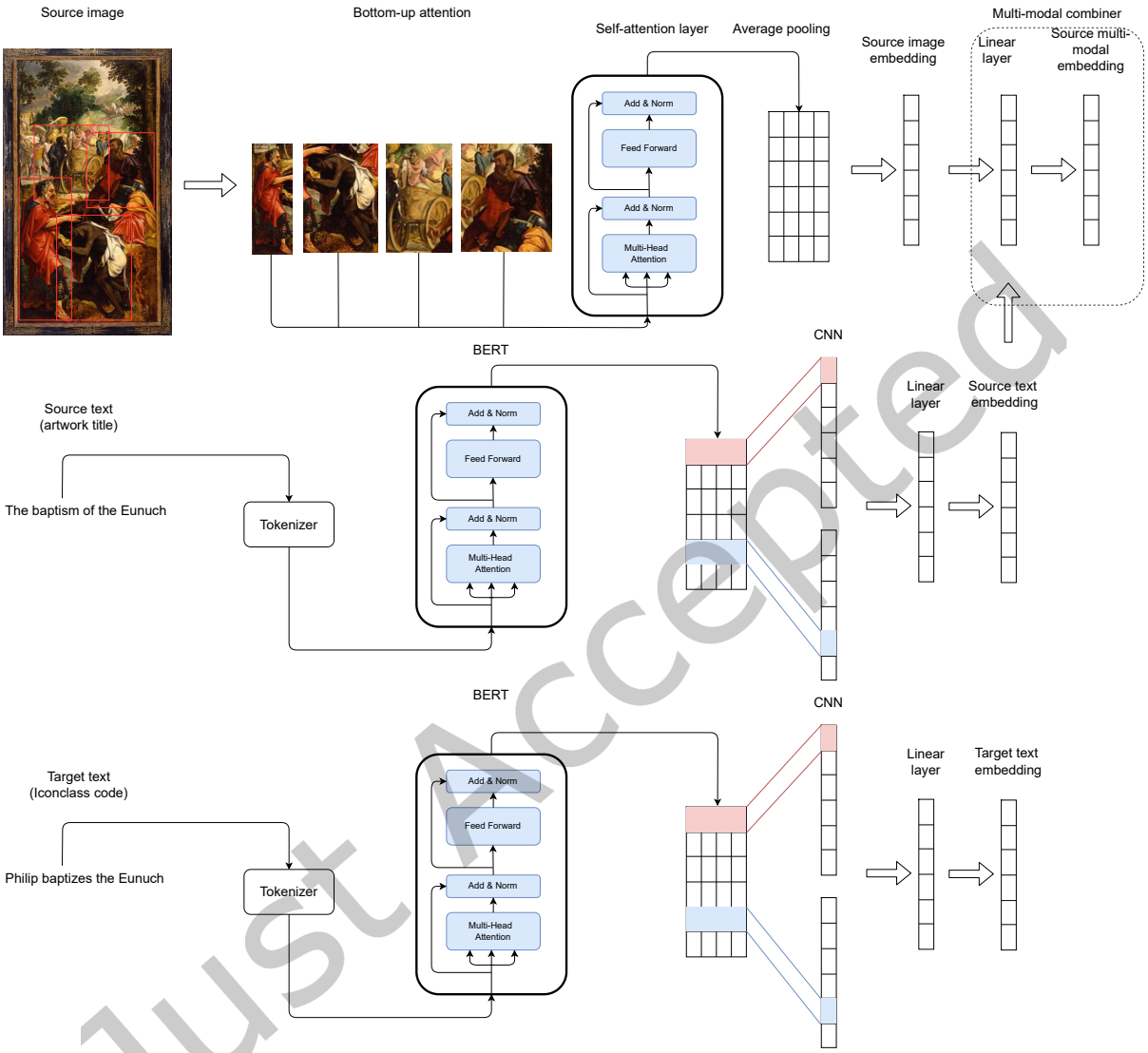

Fig. 1. The scheme of our framework FMSAEM. The upper branch builds a visual embedding from the input image. The lower branch separately processes the textual source (titles) and target (an Iconclass code definition) inputs. Next on, the multi-modal combiner uses the visual embedding and textual source embedding to construct a source multi-modal embedding. Finally, the source multi-modal embedding and target embedding are used to calculate the loss function or the matching score.

these RoIs are resized into vectors of the same size. Finally, these vectors are combined into a feature matrix and processed by a position-wise fully connected layer.

The Faster R-CNN model is not capable of encoding an order of the RoIs and complex relations between them. The self-attention layer helps to overcome this issue. It is able to attend all RoIs simultaneously and extract useful information from them. Hence, the self-attention layer encodes the relationships between the extracted RoIs. Finally, the feature matrix is averaged into the image embedding and gets L2-normalized. The image branch is implemented in PyTorch and can be fully fine-tuned. In previous work [3], the visual features were extracted from the Faster-RCNN model implemented in Caffe [22]. Hence, the structure of our visual branch fully corresponds to the visual branch from MSAEM when we freeze the Faster-RCNN in our model.

Text Branch

The text branch consists of a BERT model [15] (Bidirectional Encoder Representations from Transformers), used in conjunction with the WordPiece tokenizer and one-dimensional convolutions. BERT is a widely used architecture in natural language processing, which achieved state-of-the-art results across multiple tasks. It provides pretrained context-aware representations of words based on bi-directional context, which proved to outperform static word embeddings [30, 35]. In our task, the BERT processes the input text sequence (a title or an Iconclass code definition) and provides context-aware word embeddings. Then, one-dimensional convolutions (followed by max-pooling) are applied in order to capture the local context. Finally, all features are concatenated and processed by a fully connected layer with L2-normalization. In previous work [3], the BERT was used without fine-tuning to extract textual features. We use the uncased multi-lingual version of BERT [15] from the Simple Transformers library [39] based on [53], which can be fully fine-tuned. Our text branch is identical to that used in MSAEM if the layers of the BERT model are frozen.

Multi-Modal Combiner

The multi-modal combiner fuses the outputs of the image and text branches if multiple sources are available. If only one is available, the multi-modal combiner represents a linear transformation of the image/text branch output. The network processes first the available textual and/or visual information with the corresponding branches and, then, it concatenates the obtained representations into a single vector. In the multi-lingual scenario, each language is processed separately by the text branch as an independent source input. Next on, the vector is resized into the original size of the target embeddings by a fully connected layer. Hence, the input size of the fully connected layer depends on the number of available sources. Finally, L2-normalization is applied. Hence, the network can exploit the multi-modality and multi-linguality that is integral to the data.

Loss Function

In this work, we use a loss function from the SAEM framework, which we describe in this subsection. To calculate the loss function, we utilize the source embedding e^s, which is provided by the multi-modal combiner, and the target embedding e^t, which represents the corresponding Iconclass code processed by the text branch. We use inner product $s(e^s, e^t) = e^s \times (e^t)^\intercal$ to score the candidate Iconclass codes in the training and retrieval phases. As the embeddings are L2-normalized, the scoring function is equal to cosine similarity.

The loss function consists of a bi-directional triplet loss [55] and a bi-directional angular loss [52] with hard negative mining. The bi-directional triplet loss increases the difference between a ground-truth matching pair (s, t) and negative points:

$$L_{triplet}(s,t) = max[0, m - s(e^s, e^t) + s(e^s, e^{\hat{t}})] + max[0, m - s(e^s, e^t) + s(e^{\hat{s}}, e^t)]$$

where m denotes the margin parameter, $\hat{t}$ denotes a negative target Iconclass code for the multi-modal source object s, $\hat{s}$ denotes a negative multi-modal source object for the target Iconclass code t. In the loss function, we use hard negatives over a mini-batch to increase performance and computational efficiency of the model.

In combination with the bi-directional triplet loss, the angular loss can accelerate the training procedure [54]. The angular loss constrains the angle at the negative point of triplet triangles that leads to (i) the improved robustness of objective against feature variance; (ii) capturing additional local structure of triplet triangles; (iii) better convergence in comparison to conventional methods. We follow [54] and use the hard negative mining over a mini-batch to optimize the performance of the model:

$$L_{angular}(s,t) = log[1 + exp(f(e^s, e^t, e^{\hat{t}}))] + log[1 + exp(f(e^s, e^t, e^{\hat{s}}))]$$

where $f(e_1, e_2, e_3) = 4tan^2\alpha(e_1 + e_2)e_3^\intercal - 2(1 + tan^2\alpha)e_1e_2^\intercal$, where e_1, e_2, and e_3 correspond to embeddings, α denotes the angular margin parameter, $\hat{t}$ denotes a hard negative target Iconclass code for the multi-modal source object s, $\hat{s}$ denotes a hard negative multi-modal source object for the target Iconclass code t.

Finally, the bi-directional triplet loss and angular loss are combined into the final loss:

$$L_(s,t) = L_{triplet}(s,t) + \theta L_{angular}(s,t)$$

where θ denotes the weight of the angular loss. Following [54], we decay θ over epoch number as the angular loss plays a less important role than the triplet loss.

RESULTS AND DISCUSSION

In this section, we first describe the datasets, which we used in our experiments. Next on, we outline our experimental settings. Finally, we present and discuss our results, which we compare to previous work [3].

Datasets

Iconclass is an hierarchical classification system (see Figure 2) in the domain of iconography used for the description and retrieval of content depicted in visual artworks. Iconclass contains 28,000 codes with their corresponding textual definitions. In our work, we use Iconclass codes with a depth of 5, giving us 10,418 codes in total. We use Iconclass definitions in English as the target labels. We use the Iconclass Python package[4] to convert the Iconclass codes to definitions. The Iconclass hierarchy starts with the following 10 main categories: (0) abstract art; (1-5) general topics; (6) history; (7) Bible; (8) literature; (9) classical mythology and ancient history. Further descriptive depth can be added by the three options presented in Table 1.

N	**Extension**	**Definition**	**Keywords**
1	31A2	anatomy (non-medical)	anatomy
	31A2**2**	parts of the human body (skeleton excepted)	limb
2	25F33	predatory birds	predatory birds
	25F33**(EAGLE)**	predatory birds: eagle	eagle
3	23K22	May and its 'labours'	May
	23K22**(+1)**	May and its 'labours' (+ with zodiacal signs)	May, Tierkreis, zodiac, zodiaco

Table 1. Division of Iconclass codes: (1) a letter or a digit enhances specificity; (2) text in parentheses adds the name of a specific entity; (3) text in parentheses with a plus-sign adds 'shade of meaning'.

[4]https://labs.brill.com/ictestset/

(a) 73B4 (b) 11HH(CATHERINE) (c) 73A624

Fig. 2. Examples of images attributed with Iconclass codes [37]. The following definitions correspond to the presented codes: (a) 'presentation of the Christ-child in the temple, usually Simeon (and Anna) present (Luke 2:22-39)'; (b) 'the virgin martyr Catherine of Alexandria; possible attributes: book, crown, emperor Maxentius, palm-branch, ring, sword, wheel'; (c) 'annunciation of Christ's birth to the shepherds (and/or shepherdesses) at night; a host of singing angels in the air'.

RKD Dataset. We utilize the same dataset that was extracted and used in the immediate precursor to this work [3]. The dataset was obtained from the database of the Netherlands Institute for Art History [5]. It contains 26,725 visual artworks with the following corresponding metadata: Iconclass codes, Dutch-language artwork titles, and English-language artwork titles. We use 22,725 objects (1,843 unique codes) in the training set, 2,000 objects (617 unique codes) are used in the development set, and another 2,000 objects (612 unique codes) represent the test set. The training set contains multiple Iconclass codes per object while the development/test sets has only one Iconclass code per object. The training set and the development set have 587 unique codes in common, and the training set and the test set overlap by 577 unique codes. Hence, the model should (minimally) be able to predict the labels that are observed in the training set (94-95 percent). However, the set of the possible labels is still large (10,418 codes).

Rijksmuseum Dataset. In addition, we use an external collection to evaluate our best models. We extracted this dataset from the website of Rijksmuseum.[6] The dataset contains 2,000 objects with 939 unique codes, which we use for evaluation. The extracted metadata consist of Iconclass codes and Dutch-language artwork titles. The presented dataset is more challenging for evaluation than the RKD test set for the following reasons. This dataset overlaps with the training set by 674 unique codes. Hence, only 72 percent of the labels from the Rijksmuseum dataset are attested in the RKD training set. In addition to the general difference in content between the collections,

[5]https://rkd.nl/en/explore/images
[6]http://www.rijksmuseum.nl/collectie

the annotation of these collections are conducted by the different institutions and individuals, who might have adopted different annotation rationales, even if they were using the same thesaurus.

Zero-Shot Rijksmuseum Dataset. This dataset consists of 452 objects from the Rijksmuseum dataset with 265 unique Iconclass codes that are not contained in the training set.

Experimental Settings

We use MCLIP [38] and features from M^{3}P [32] as additional baselines to FMSAEM. The text branch of MCLIP consists of fined-tuned multi-lingual BERT[7]. The the visual branch contains the Res50x4 vision encoder, which is a ResNet50 network scaled in the EfficientNet style [49]. M^{3}P is a BERT-based model with visual features extracted by Faster-RCNN, and we use this model as a feature extractor (which is not fine-tuned). MCLIP and M^{3}P provide textual and visual embeddings, which are fed to the multi-modal combiner in the same way as in FMSAEM.

In our experiments, we mimic the experimental settings from the previous work [3] and use the default hyper-parameters of the SAEM framework to train all models. The Adam optimizer [23] minimizes a combination of a bi-directional triplet loss [55] and a bi-directional angular loss [52] with hard negative mining. The learning rate decays by a factor of 0.1 each 10 epochs starting from the initial value of 0.0001. We set a batch size to 64 objects (images or/and sentences) and adjust gradient accumulation steps to fit our model into two GeForce GTX TITAN X with 12 GB RAM. We evaluate our models using the standard metric Recall@K (for K=1, 5, 10). We train models for 30 epochs and select the best performer (using an average of the adopted metrics) on the validation dataset. In addition, we select the optimal number of frozen layers (*see below*) in our models, again on the basis of the validation dataset. We divide the ResNet-101 (with 33 layers) from the image branch of FMSAEM into 5 similar chunks, and the BERT (with 12 layers) from the text branch is split into 6 equal chunks. When the ResNet-101 and BERT are completely frozen, the structure of our framework FMSAEM fully corresponds to the previously published framework MSAEM [3]. We conduct the same experiments with the Res50x4 encoder (with 4 layers) from MCLIP to find the optimal amount of the frozen layers.

Experiments

In this subsection, we provide the main results of our experiments. First, we vary the number of the layers that should be frozen (expressed as a proportion of the total number of layers) in both branches of our framework, which is likely to affect the performance of the fine-tuned architecture. Secondly, we compare our framework with the previously obtained results [3], M^{3}P, and MCLIP. Next on, we evaluate our models on the external dataset and evaluate their zero-shot performance. Finally, we experiment with fusion techniques to combine the available textual and visual sources of FMSAEM.

Fine-tuning on the RKD Dataset. Figure 3 investigates an optimal number of layers for BERT and ResNet-101 that can be frozen to obtain the best performance. Figure 3 (on the left) corresponds to the cross-modal experiment (a) with FMSAEM from Table 2. We can observe that fine-tuning of the last ResNet-101 layers drastically improves the performance of FMSAEM while fine-tuning of BERT is almost not beneficial. According to this figure, the optimal number of layers for ResNet-101 corresponds to 19 layers (out of 33 layers). Next on, we use 19 layers for ResNet-101 in all fine-tuning experiments containing the visual input. We conducted the same experiment (a) with MCLIP and found that the optimal number of layers for ResNet-50x4 corresponds to 3 layers (out of 4 layers), which we use in the further experiments. Figure 3 (on the right) corresponds to the text-based experiments (b, c, f) with FMSAEM from Table 2. The optimal number of layers for BERT on average corresponds to 10 layers (out of

[7]https://github.com/FreddeFrallan/Multilingual-CLIP

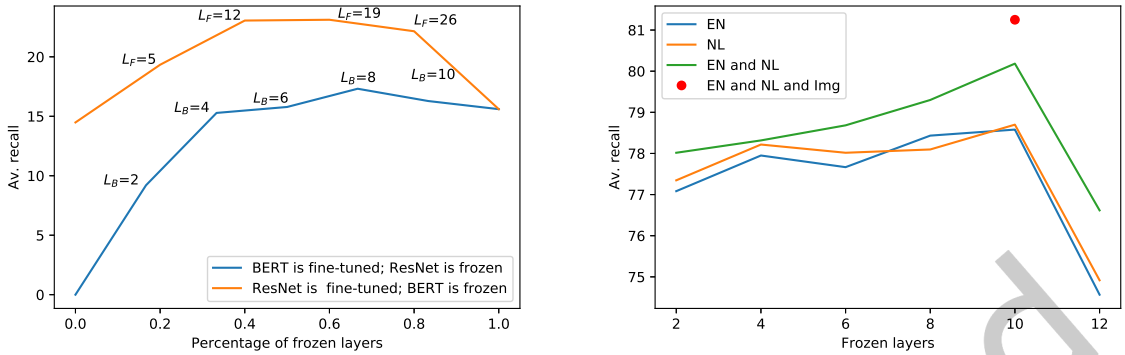

Fig. 3. Experiments, conducted on the RKD validation set, for different proportions of frozen layers. The figure to the left corresponds to the experiment, where BERT (or ResNet-101) from FMSAEM is completely frozen while ResNet-101 (or BERT) is fine-tuned. The figure to the right corresponds to the text-based experiments with BERT, where various sources are used.

12 layers). Further, we use the optimal number of layers from this figure for BERT in all fine-tuning experiments with the textual input for both FMSAEM and MCLIP.

In Table 2, we provide results that enable the comparison of our model without fine-tuning MSAEM$_{reimp}$ to the similar model from the previous work MSAEM. In the cross-modal experiment (a), the original model MSAEM outperforms our model MSAEM$_{reimp}$ by a large margin. Next on, we observe the same trends in the multi-modal experiments with the visual input (d, e, g). Hence, the initial quality of the visual features provided by our framework is lower than in the related model from the previous work. Despite the full correspondence of the models' structure, it is not uncommon that different implementations would provide different results. Similarly to the previous work, we observe that visual features mostly improve results of our MSAEM$_{reimp}$ model in text-based experiments (d, e, g). However, the difference is less convincing because the margin between results is moderate, and, in the experiment (e) for MSAEM$_{reimp}$, we can see even small decrease of the results, which can be explained by the low quality of the initial visual features.

In the text-based experiments (b, c, f) without fine-tuning, we can observe that the results fluctuate from case to case in favor of either MSAEM from the previous work or the our MSAEM$_{reimp}$. Similarly to the previous work, we can see that MSAEM$_{reimp}$ with the Dutch input in the experiment (c) performs at the same level as the corresponding model with the English input in the experiment (b). Finally, we observe that the increasing number of sources in the experiments with MSAEM$_{reimp}$, which is not fine-tuned, moderately improves the overall performance, bringing it on par with the previous results.

MSAEM$_{reimp}$ demonstrates comparable performance to M^{3}P, which is similarly not fine-tuned. MSAEM$_{reimp}$ provides superior visual features that can be seen from the experiment (a). In all other experiments, the difference between the models is subtle.

As can be seen from Table 2, the fine-tuning of FMSAEM improves the performance by a large margin in comparison to MSAEM$_{reimp}$. We can observe that the quality of the visual features in the experiment (a) is considerably increased, which leads to the improved results of the FMSAEM models with the visual input in the experiments (d, e, g) in the comparison to the corresponding text-based experiments (b, c, f). Similarly to the previous work, we can observe that FMSAEM with the Dutch titles in the experiment (c) performs on par with the corresponding FMSAEM model in the experiment (b) based on the English titles. This outcome will be useful for GLAM institutions that work with local languages in the resource-scarce scenario to retrieve Iconclass codes

Exp.	Model	Source			Metrics			
		Image	EN	NL	Recall@1	Recall@5	Recall@10	Average
a	MSAEM	✓			13.10	19.60	23.20	18.63
	MSAEM$_{reimp}$	✓			9.50	16.20	19.30	15.00
	M^{3}P	✓			6.05	10.95	14.20	10.40
	FMSAEM	✓			**14.50**	**24.40**	**28.05**	**22.32**
	MCLIP	✓			11.20	17.25	20.85	16.43
b	MSAEM		✓		62.80	77.30	80.75	73.62
	MSAEM$_{reimp}$		✓		66.95	77.30	79.75	74.67
	M^{3}P		✓		66.75	76.90	79.95	74.53
	FMSAEM		✓		70.95	81.55	85.05	79.18
	MCLIP		✓		**71.60**	**82.50**	**85.25**	**79.78**
c	MSAEM			✓	66.45	77.05	80.55	74.68
	MSAEM$_{reimp}$			✓	67.05	77.20	79.20	74.48
	M^{3}P			✓	66.10	76.45	78.85	73.80
	FMSAEM			✓	68.25	**81.95**	**84.95**	**78.38**
	MCLIP			✓	**69.70**	79.90	83.50	77.70
d	MSAEM	✓	✓		67.85	79.20	82.30	76.45
	MSAEM$_{reimp}$	✓	✓		66.20	77.65	80.85	74.90
	M^{3}P	✓	✓		68.05	77.95	80.85	75.62
	FMSAEM	✓	✓		71.10	**83.80**	**87.00**	**80.63**
	MCLIP	✓	✓		**71.25**	81.95	84.85	79.35
e	MSAEM	✓		✓	66.60	78.80	81.80	75.73
	MSAEM$_{reimp}$	✓		✓	66.55	76.45	79.70	74.23
	M^{3}P	✓		✓	67.30	77.00	80.00	74.77
	FMSAEM	✓		✓	68.55	**81.65**	**85.60**	**78.60**
	MCLIP	✓		✓	**69.25**	80.45	83.40	77.70
f	MSAEM		✓	✓	68.95	80.30	82.90	77.38
	MSAEM$_{reimp}$		✓	✓	68.10	78.50	81.25	75.95
	M^{3}P		✓	✓	68.25	78.95	81.35	76.18
	FMSAEM		✓	✓	71.65	82.15	85.45	79.75
	MCLIP		✓	✓	**72.30**	**82.80**	**85.65**	**80.25**
g	MSAEM	✓	✓	✓	70.05	80.35	83.10	77.83
	MSAEM$_{reimp}$	✓	✓	✓	70.05	79.15	82.20	77.13
	M^{3}P	✓	✓	✓	69.35	79.10	81.35	76.60
	FMSAEM	✓	✓	✓	**72.35**	**84.35**	**87.70**	**81.47**
	MCLIP	✓	✓	✓	71.25	81.95	84.85	79.35

Table 2. Results of the experiments for different matching sources. The experiments with MSAEM correspond to the results from previous work [3]. The experiments with MSAEM$_{reimp}$ correspond to the framework under scrutiny, but without any fine-tuning of the ResNet-101 or BERT. The experiments with FMSAEM correspond to fine-tuning of the current framework.

without translating their metadata to English. In addition, we demonstrate that the increasing number of the

Model	Test Source	Training Source			Metrics			
		Image	EN	NL	Recall@1	Recall@5	Recall@10	Average
MSAEM$_{reimp}$	NL		✓		31.15	42.95	47.10	40.40
M^{3}P	NL		✓		27.40	37.30	40.85	35.18
FMSAEM	NL		✓		39.45	56.20	60.20	51.95
MCLIP	NL		✓		**42.70**	**58.40**	**62.35**	**54.48**
MSAEM$_{reimp}$	EN			✓	30.10	42.30	47.10	39.83
M^{3}P	EN			✓	24.95	34.40	38.50	32.62
FMSAEM	EN			✓	42.20	61.20	68.30	57.23
MCLIP	EN			✓	**48.75**	**62.35**	**68.70**	**59.93**

Table 3. Results of the cross-lingual experiments on the RKD test set. In these experiments, the language of the textual input in the training phase does not correspond to the language in the test phase.

sources in the experiments with fine-tuning boosts the overall performance, which corresponds to the results from the previous paper.

Finally, we compare our model FMSAEM with M^{3}P and MCLIP. From Table 2, we can observe that FMSAEM outperforms M^{3}P in all experiments by a large margin. Our model is better than MCLIP on average in 5 out of 7 cases. MCLIP demonstrates better results only in the text-based experiments (b, f). However, quality of the visual features from MCLIP is less promising than from FMSAEM, which can be seen from the experiment (a). In addition, adding the visual features to the textual ones does not improve or even worsens the performance of MCLIP.

Cross-Lingual Experiments. We conduct cross-lingual experiments, the results of which can be seen in Table 3. We can observe that if the evaluation is not conducted on the same language as training, the performance drops considerably. However, the models still are able to match Iconclass codes with decent results, due to the multi-lingual nature of the BERT model considered here. The fine-tuned FMSAEM model outperforms MSAEM$_{reimp}$ by a large margin. MCLIP demonstrates the best performance in the cross-lingual experiments, which confirms the high quality of the textual representations from MCLIP. The difference in performance between the English and Dutch fine-tuned models from Table 3 may be explained by presence of English at the target side. The Dutch models, which outperforms the English models, are trained on English Iconclass definitions while the English models have not encountered any Dutch during the training procedure.

Evaluation on the Rijksmuseum Dataset. From Table 4, we can observe that the performance drops by a large margin for the evaluation conducted on the external dataset. Such results are expected due to the general differences between the RKD and Rijksmuseum datasets, which we discussed earlier. The annotation is performed by different individuals, working in different contexts and adopting different annotation rationale; additionally, the nature and and composition of the collections are different, which can be expected to degrade the performance for deep networks [41]. In addition, the Rijksmuseum dataset has a more elevated number of unseen labels in the evaluation set, which makes them more difficult to predict. We can observe from Table 4 that fine-tuning largely improves performance of FMSAEM in comparison to MSAEM$_{reimp}$ in all experiments. We can additionally see that the increasing number of matching sources boosts the final results of FMSAEM. M^{3}P is outperformed by our model in the cross-modal scenario, however, it unexpectedly demonstrates the best performance among the baselines. Similarly to the results from Table 4, MCLIP provides less promising visual features than our model, but outperforms it in the text-based experiment. Finally, FMSAEM outperforms the baseline models in the multi-modal experiment.

Exp.	Model	Source			Metrics			
		Image	EN	NL	Recall@1	Recall@5	Recall@10	Average
a	$MSAEM_{reimp}$	✓			3.03 (3.75)	5.02 (6.30)	6.35 (8.00)	4.80 (6.02)
	M^3P	✓			3.46 (4.15)	6.62 (7.95)	8.46 (10.40)	6.18 (7.50)
	FMSAEM	✓			**4.08 (4.85)**	**7.55 (9.00)**	**9.89 (11.90)**	**7.17 (8.58)**
	MCLIP	✓			3.07(3.60)	5.71 (7.00)	7.48 (9.30)	5.42 (6.63)
c	$MSAEM_{reimp}$			✓	14.73 (17.95)	21.59 (27.00)	24.56 (30.80)	20.29 (25.25)
	M^3P			✓	12.78 (15.70)	18.37 (22.40)	21.08 (25.90)	17.41 (21.33)
	FMSAEM			✓	18.59 (22.95)	26.70 (33.15)	30.26 (37.05)	25.18 (31.05)
	MCLIP			✓	**19.61 (23.75)**	**28.53 (34.95)**	**31.53 (38.45)**	**26.56 (32.38)**
e	$MSAEM_{reimp}$	✓		✓	15.65 (18.85)	22.32 (27.00)	26.04 (31.90)	21.34 (25.92)
	M^3P	✓		✓	14.11 (16.95)	20.24 (24.30)	23.00 (27.70)	19.12 (22.98)
	FMSAEM	✓		✓	**18.76 (22.85)**	**27.69 (34.55)**	**32.58 (40.40)**	**26.34 (32.60)**
	MCLIP	✓		✓	17.84 (21.20)	26.34 (31.95)	30.90 (37.55)	25.04 (30.23)

Table 4. Evaluation on the Rijksmuseum dataset, which contains the Dutch titles and visual reproductions of the artworks. The numbers in the brackets correspond to a setting with one ground-truth label per object, while the numbers without brackets correspond to the multi-label evaluation.

Zero-Shot Evaluation on the Rijksmuseum Dataset. As it can be seen from Table 5, all models have a poor performance in the zero-shot evaluation setting. M^3P shows the worst results among the presented models in this setting. MCLIP moderately outperforms FMSAEM in the multi-modal and text-based experiments while FMSAEM is slightly better in the cross-modal experiment. In addition, we can see that fine-tuning improves the performance of FMSAEM in comparison to $MSAEM_{reimp}$.

Exp.	Model	Source			Metrics			
		Image	EN	NL	Recall@1	Recall@5	Recall@10	Average
a	$MSAEM_{reimp}$	✓			0 (0)	0 (0)	0.66 (0.66)	0.22 (0.22)
	M^3P	✓			0 (0)	0.22 (0.22)	0.22 (0.22)	0.15 (0.15)
	FMSAEM	✓			0 (0)	0.22 (0.22)	**1.11 (1.11)**	**0.44 (0.44)**
	MCLIP	✓			0 (0)	**0.33 (0.44)**	0.33 (0.44)	0.22 (0.29)
c	$MSAEM_{reimp}$			✓	0.22 (0.22)	1.33 (1.55)	2.51 (2.88)	1.35 (1.55)
	M^3P			✓	0.44 (0.44)	0.77 (0.88)	1.88 (1.99)	1.03 (1.11)
	FMSAEM			✓	0.55 (0.66)	2.99 (3.32)	4.54 (5.09)	2.69 (3.02)
	MCLIP			✓	**1.44 (1.55)**	**5.42 (5.75)**	**7.04 (7.52)**	**4.63 (4.94)**
e	$MSAEM_{reimp}$	✓		✓	0.22 (0.22)	1.11 (1.11)	1.77 (1.77)	1.03 (1.03)
	M^3P	✓		✓	0 (0)	0.74 (0.88)	1.73 (1.99)	0.82 (0.96)
	FMSAEM	✓		✓	**1.22 (1.33)**	3.83 (4.20)	5.83 (6.42)	3.63 (3.98)
	MCLIP	✓		✓	0.66 (0.88)	**4.50 (4.87)**	**7.26 (8.19)**	**4.14 (4.65)**

Table 5. Zero-shot evaluation on the Rijksmuseum dataset. The Iconclass codes in this evaluation setting are not observed by the models during the training procedure. The numbers in the brackets correspond to a setting with one ground-truth label per object, while the numbers without brackets correspond to the multi-label evaluation.

Fusion Techniques. Table 6 investigates different fusion techniques for the multi-modal input of FMSAEM. We can observe a small difference in performance with the superiority of the concatenation technique.

Fusion	Metrics			
	Recall@1	Recall@5	Recall@10	Average
average	**72.40**	84.00	86.80	81.06
weighted average	71.80	83.80	86.95	80.85
concatenation	72.35	**84.35**	**87.70**	**81.47**

Table 6. Effect of different fusion techniques for the multi-modal input of FMSAEM. The evaluation is conducted on the RKD dataset. The average fusion corresponds to calculating the average representation of the available sources before feeding it to the last linear layer (the multi-modal combiner). In the weighted average fusion, the weights are learnt from data. In the concatenation technique, all source representations are concatenated before feeding them to the multi-modal combiner.

CONCLUSION

In this work, we developed a framework for the multi-lingual and multi-modal attribution of Iconclass codes using a deep neural network, with the additional support of full fine-tuning. Because our framework is based on previous work [3] in this domain, both works use the same, multiple sources for the attribution, namely, the English and Dutch textual features from artwork titles and the visual features from artwork images. The main difference consists in the added fine-tuning phase, which largely boosts the performance of the proposed model. Our experiments (re)confirm the following findings. Image-text matching is less promising than the text-text matching even after the extensive fine-tuning. However, visual features still help to improve performance of the multi-modal model. The attribution based on the Dutch-language artwork titles is as promising as the attribution based on the English-language artwork titles. Hence, the framework can in theory be used in the GLAM sector around the world to attribute Iconclass codes in local, perhaps lesser resourced languages, without

N	code	definition
1	11F71	specific aspects of Christ-child ~ Madonna-representations (N.B. secondary notation only)
2	11F73	specific aspects of Mary ~ Madonna-representations (N.B. secondary notation only)
3	11F42	Madonna: Mary sitting or enthroned, the Christ-child in her lap (or in front of her bosom) (Mary sometimes represented half-length)
4	11F43	Madonna: Mary sitting on the ground, the Christ-child in her lap
5	11F72	specific aspects of Christ-child and Mary ~ Madonna-representations (N.B. secondary notation only)
6	11F45	Mary kneeling (on the ground), the Christ-child in front of her
7	11F74	other specific aspects of Madonna-representations
8	11F41	**Madonna: Mary standing (or half-length), Christ-child close to her bosom**
9	73B83	representations derived from Holy Family
10	41A65	water-works in garden

Table 7. Example of the top 10 Iconclass codes offered by the best performing model FMSAEM based on the artwork 'Virgin and Child with book of prayers'. The ground-truth Iconclass code is highlighted in bold.

having to translate their metadata to English. FMSAEM demonstrated the better performance than M3P in all experiments, and our model outperformed MCLIP in most experiments due the better quality of the visual features. Interestingly, we observe that MCLIP outperforms our framework in most text-based experiments.

In addition, we conducted an experiment where the input language of the test set is different from the input language of the training set. The experiment demonstrates that the performance is still decent, notwithstanding a large drop in average recall. MCLIP has a superior performance over other models, which again confirms the high quality of the provided textual features.

We observe a large degradation of performance for evaluation on the external collection, which is different from the training set. Such results may be explained by differences between the collections as well as by the smaller size of the training set (22,725 objects and 1,843 unique codes) in comparison to the number of possible labels, which is 10,418. However, we observed that fine-tuning still helps to improve performance even in the cross-lingual setting. In the zero-shot experiment, all models fail to demonstrate any meaningful results because of the high complexity of the task. From Table 7, we can see that the Iconclass system contains many similar labels, which makes this task extremely challenging. In the provided example, the first eight codes offered by our model have the same ancestor node (11F). To overcome the issues listed above, the framework will need more annotated data that covers more unique artworks with the corresponding labels. Hence, the future work should focus on continuing efforts regarding data gathering in the field. In addition, we observed that the visual branch remains a weak part of the framework, which can be improved in the future work by using an object detector already adapted for the art domain.

REFERENCES

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 6077–6086.

[2] Nikolay Banar, Walter Daelemans, and Mike Kestemont. 2020. Neural Machine Translation of Artwork Titles Using Iconclass Codes. In *Proceedings of the The 4th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*. 42–51.

[3] Nikolay Banar, Walter Daelemans, and Mike Kestemont. 2021. Multi-modal Label Retrieval for the Visual Arts: The Case of Iconclass.. In *ICAART (1)*. 622–629.

[4] Lorenzo Baraldi, Marcella Cornia, Costantino Grana, and Rita Cucchiara. 2018. Aligning text and document illustrations: towards visually explainable digital humanities. In *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 1097–1102.

[5] Hans Brandhorst. 2019. A Word is Worth a Thousand Pictures: Why the Use of Iconclass Will Make Artificial Intelligence Smarter. https://labs.brill.com/ictestset/ICONCLASS_and_AI.pdf.

[6] Giovanna Castellano and Gennaro Vessio. 2021. Deep learning approaches to pattern extraction and recognition in paintings and drawings: an overview. *Neural Computing and Applications* (2021), 1–20.

[7] Eva Cetinic. 2021. Iconographic image captioning for artworks. *arXiv preprint arXiv:2102.03942* (2021).

[8] Eva Cetinic. 2021. Towards Generating and Evaluating Iconographic Image Captions of Artworks. *Journal of Imaging* 7, 8 (2021), 123.

[9] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation learning. In *European conference on computer vision*. Springer, 104–120.

[10] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[11] Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, Massimiliano Corsini, and Rita Cucchiara. 2020. Explaining digital humanities by aligning images and textual descriptions. *Pattern Recognition Letters* 129 (2020), 166–172.

[12] Elliot J Crowley, Omkar M Parkhi, and Andrew Zisserman. 2015. Face painting: querying art with photos. (2015).

[13] Elliot J Crowley and Andrew Zisserman. 2014. In search of art. In *European conference on computer vision*. Springer, 54–70.

[14] Elliot J Crowley and Andrew Zisserman. 2014. The state of the art: Object retrieval in paintings using discriminative regions. (2014).

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT (1)*.

[16] Hongliang Fei, Tan Yu, and Ping Li. 2021. Cross-lingual Cross-modal Pretraining for Multimodal Retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 3644–3650.

[17] Marco Fiorucci, Marina Khoroshiltseva, Massimiliano Pontil, Arianna Traviglia, Alessio Del Bue, and Stuart James. 2020. Machine Learning for Cultural Heritage: A Survey. *Pattern Recognition Letters* 133 (2020), 102–108.
[18] Noa Garcia and George Vogiatzis. 2018. How to read paintings: semantic art understanding with multi-modal retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. 0–0.
[19] Ismael Garrido-Muñoz, Arturo Montejo-Ráez, Fernando Martínez-Santiago, and L Alfonso Ureña-López. 2021. A Survey on Bias in Deep NLP. *Applied Sciences* 11, 7 (2021), 3184.
[20] Angelika Grund. 1993. ICONCLASS. On subject analysis of iconographic representations of works of art. *KO KNOWLEDGE ORGANIZATION* 20, 1 (1993), 20–29.
[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
[22] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*. 675–678.
[23] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[24] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision* 123, 1 (2017), 32–73.
[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.
[26] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.
[27] Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. 2020. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 11336–11344.
[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, 740–755.
[29] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. ViLBERT: pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. 13–23.
[30] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.
[31] Federico Milani and Piero Fraternali. 2020. A Data Set and a Convolutional Model for Iconography Classification in Paintings. *arXiv preprint arXiv:2010.11697* (2020).
[32] Minheng Ni, Haoyang Huang, Lin Su, Edward Cui, Taroon Bharti, Lijuan Wang, Dongdong Zhang, and Nan Duan. 2021. M3p: Learning universal representations via multitask multilingual multimodal pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3977–3986.
[33] Erwin Panofsky. 2018. *Studies in iconology: humanistic themes in the art of the Renaissance*. Routledge.
[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf
[35] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.
[36] Nicolò Oreste Pinciroli Vago, Federico Milani, Piero Fraternali, and Ricardo da Silva Torres. 2021. Comparing CAM Algorithms for the Identification of Salient Image Features in Iconography Artwork Analysis. *Journal of Imaging* 7, 7 (2021), 106.
[37] Etienne Posthumus. 2020. Brill Iconclass AI Test Set. https://labs.brill.com/ictestset/.
[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*. PMLR, 8748–8763.
[39] T. C. Rajapakse. 2019. Simple Transformers. https://github.com/ThilinaRajapakse/simpletransformers.
[40] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*. 91–99.
[41] Ricardo Ribani and Mauricio Marengoni. 2019. A Survey of Transfer Learning for Convolutional Neural Networks. In *2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)*. IEEE, 47–57.

[42] Matthia Sabatelli, Nikolay Banar, Marie Cocriamont, Eva Coudyzer, Karine Lasaracina, Walter Daelemans, Pierre Geurts, and Mike Kestemont. 2021. Advances in Digital Music Iconography: Benchmarking the detection of musical instruments in unrestricted, non-photorealistic images from the artistic domain. *Digital Humanities Quarterly* 15, 1 (2021).

[43] Matthia Sabatelli, Mike Kestemont, Walter Daelemans, and Pierre Geurts. 2018. Deep transfer learning for art classification problems. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*. 0–0.

[44] Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural networks* 61 (2015), 85–117.

[45] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*.

[46] Matteo Stefanini, Marcella Cornia, Lorenzo Baraldi, Massimiliano Corsini, and Rita Cucchiara. 2019. Artpedia: A new visual-semantic dataset with visual and contextual sentences in the artistic domain. In *International Conference on Image Analysis and Processing*. Springer, 729–740.

[47] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2019. VL-BERT: Pre-training of Generic Visual-Linguistic Representations. In *International Conference on Learning Representations*.

[48] Hao Tan and Mohit Bansal. 2019. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 5100–5111.

[49] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*. PMLR, 6105–6114.

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[51] G. Vellekoop, E. Tholen, and L. D. Couprie. 1973. *Iconclass : an iconographic classification system.* North-Holland Pub. Co., Amsterdam.

[52] Jian Wang, Feng Zhou, Shilei Wen, Xiao Liu, and Yuanqing Lin. 2017. Deep metric learning with angular loss. In *Proceedings of the IEEE International Conference on Computer Vision*. 2593–2601.

[53] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, 38–45. https://www.aclweb.org/anthology/2020.emnlp-demos.6

[54] Yiling Wu, Shuhui Wang, Guoli Song, and Qingming Huang. 2019. Learning fragment self-attention embeddings for image-text matching. In *Proceedings of the 27th ACM International Conference on Multimedia*. 2088–2096.

[55] Yiling Wu, Shuhui Wang, Guoli Song, and Qingming Huang. 2019. Online asymmetric metric learning with multi-layer similarity aggregation for cross-modal retrieval. *IEEE Transactions on Image Processing* 28, 9 (2019), 4299–4312.

[56] Jianwei Yang, Jiasen Lu, Dhruv Batra, and Devi Parikh. 2017. A Faster Pytorch Implementation of Faster R-CNN. *https://github.com/jwyang/faster-rcnn.pytorch* (2017).

[57] Mingyang Zhou, Luowei Zhou, Shuohang Wang, Yu Cheng, Linjie Li, Zhou Yu, and Jingjing Liu. 2021. UC2: Universal Cross-lingual Cross-modal Vision-and-Language Pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4155–4165.