computers

MDPI

Article

Knowledge Graph Embedding-Based Domain Adaptation for Musical Instrument Recognition

Victoria Eyharabide [1,*], Imad Eddine Ibrahim Bekkouch [2] and Nicolae Dragoș Constantin [3]

[1] STIH Laboratory, Sorbonne University, 75005 Paris, France
[2] Sorbonne Center for Artificial Intelligence, Sorbonne University, 75005 Paris, France; imad.bekkouch@etu.sorbonne-universite.fr
[3] Research Institute for Artificial Intelligence, Romanian Academy, 050711 Bucharest, Romania; dragosnicolae555@gmail.com
* Correspondence: maria-victoria.eyharabide@sorbonne-universite.fr

Abstract: Convolutional neural networks raised the bar for machine learning and artificial intelligence applications, mainly due to the abundance of data and computations. However, there is not always enough data for training, especially when it comes to historical collections of cultural heritage where the original artworks have been destroyed or damaged over time. Transfer Learning and domain adaptation techniques are possible solutions to tackle the issue of data scarcity. This article presents a new method for domain adaptation based on Knowledge graph embeddings. Knowledge Graph embedding forms a projection of a knowledge graph into a lower-dimensional where entities and relations are represented into continuous vector spaces. Our method incorporates these semantic vector spaces as a key ingredient to guide the domain adaptation process. We combined knowledge graph embeddings with visual embeddings from the images and trained a neural network with the combined embeddings as anchors using an extension of Fisher's linear discriminant. We evaluated our approach on two cultural heritage datasets of images containing medieval and renaissance musical instruments. The experimental results showed a significant increase in the baselines and state-of-the-art performance compared with other domain adaptation methods.

Keywords: knowledge graph embeddings; knowledge graph; neural network; domain adaptation; cultural heritage; musical iconography; digital humanities

check for updates

Citation: Eyharabide, V.; Bekkouch, I.E.I.; Constantin, N.D. Knowledge Graph Embedding-Based Domain Adaptation for Musical Instrument Recognition. *Computers* **2021**, *10*, 94. https://doi.org/10.3390/computers10080094

Academic Editor: Francesca Fallucchi

Received: 30 June 2021
Accepted: 30 July 2021
Published: 3 August 2021

1. Introduction

It is not a secret that artificial neural networks are nowadays predominant in machine learning applications. As the core component of Deep Learning methods, neural networks have revolutionized the field of predictions [1,2]. The first breakthrough was made in computer vision in 2012 with AlexNet, and since then, neural networks became the focus of research where the field is pushed forward every month with new state-of-the-art methods [3]. Although neural networks are powerful models, they still suffer from many drawbacks. As they are being used in many fields and areas, they face new problems previously unseen.

The bar of expectations for neural networks now is so high that they need to provide human-level performance not only on the data they are trained, but also on different target datasets. However, they cannot do so easily as they are biased towards the dataset they were trained. This problem, known as Domain Gaps, is similar to the problem of overfitting, but in this case, the model generalizes well on the testing data but cannot generalize on new unseen datasets. This challenge increases when dealing with cultural heritage data which is generally difficult to label and acquire, and thus the trained model cannot generalize well.

Transfer Learning is the field of artificial intelligence that focuses on improving a model's performance on a challenging target domain by leveraging data from a well-known source domain. In particular, Domain Adaptation is a subfield of Transfer Learning that aims to minimize the domain gap between the source and target domains, reducing the required data to train a neural network. This idea has become a best practice where it is considered abnormal to retrain a neural network from scratch but instead use a pre-trained model on a large dataset and then only retrain the last layers. This strategy improves the models' performance drastically. However, this is still not enough since the source dataset classes do not necessarily overlap with the target domain. In industrial products, the solution is to collect more data and retrain the target model. Nevertheless, augmenting the dataset size is not feasible for cultural heritage data when only a few original artworks are preserved.

Cultural heritage data are challenging to acquire, demanding to label, and vary in style through different historical periods. First, finding medieval artwork images containing a particular type of musical instrument is difficult; the older the instrument, the fewer artworks that contain it are found. Second, as ancient artworks are generally damaged or deteriorated, experts may have difficulties in classifying the instruments. Finally, when dealing with images containing musical instruments from different historical periods, there are significant differences in how they were painted or sculpted. Besides, the instruments may differ according to the artwork supporting materials, such as paintings, manuscripts, photographs, or sculptures. Those difficulties make the training process on such heterogeneous images a more demanding task.

Knowledge Graphs (KG) played a significant role in preserving cultural heritage and modeling human expert knowledge. In the last ten years, hundreds of semantic data models, vocabularies, and knowledge graphs have been developed for digital humanities, such as MusicKG [4], Sampo [5], NOnt [6], or CLARIAH [7]. Knowledge graphs provide rich semantic context about the images' content that is useful to extract class-informative embeddings. This article's contribution is to add semantic information gathered in knowledge graphs when training neural networks with sparse and heterogeneous image datasets, which is the case of cultural heritage data. In this approach, we use knowledge graphs as an anchor for our deep learning models to organize and direct the model's focus and incorporate not only visual information in the training process but also global and more connected information. We evaluate our method on our collected image dataset of Medieval Musical Instruments, which was annotated and carefully verified by five musicologists specialized in medieval musical instruments. The images we use as source data and their knowledge graph are extracted from the Musiconis dataset (see Figure 1), whereas our target images came from the Vihuelas dataset [8].

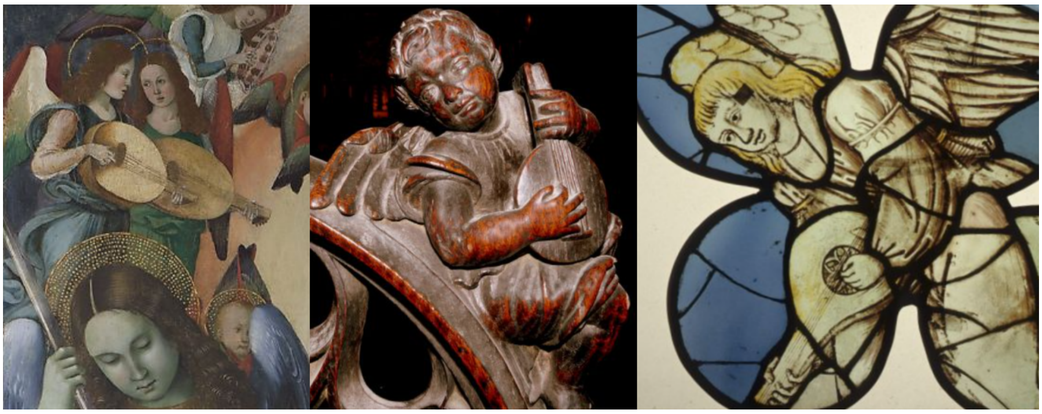


Figure 1. Three images of lutes (luths) in the Musiconis database.

The rest of the paper sections are organized as follows. Section 2 is an overview of related works. Section 3 describes our model in detail. The empirical evaluation of our method is shown in Section 4. Finally, Section 5 summarizes the paper.

2. Related Works

Our paper combines transfer learning (domain adaptation) and knowledge graphs to create a novel domain adaptation method that is most suited for cultural heritage datasets. The current state-of-the-art methods rely widely on adversarial and generative models, which provide high performances compared to traditional techniques but require many unsupervised and unstructured data. On the contrary, for cultural heritage data, the number of samples is scarce, but each sample's labeling and annotations are exhaustive. In most state-of-the-art methods in domain adaptation, a discriminator plays a core role in handling the domain independence characteristic for the latent space and is sometimes used to evaluate the performed adaptation's efficiency. Knowledge graphs are powerful tools for modeling cultural heritage data, and their embeddings can be used to provide more information about the samples. In this section, we detail some related approaches on these topics and the advancements in the field.

2.1. Knowledge Graph Embeddings

Knowledge graph embeddings methods aim at mapping a component of a KG, including nodes, entities, and relationships between them, into a continuous vector space using algorithms such as Node2Vec. Machine learning models, mainly neural networks, can use the resulting vectors, as this structure is simpler than a graph structure, while preserving the inherent information and structure of the graph [9]. These techniques have gained popularity due to their wide utility for downstream applications such as KG completion and relation extraction, Recommender Systems, Question Answering, and Relation Extraction from texts.

Knowledge graph embeddings (KGEs) are considered to be a low-dimensional representations of the nodes and relations in a knowledge graph. Knowledge graph embeddings are mappings on different parts of the knowledge graph into a vector space that satisfy certain properties and maintain the information that exists in the graph. Each method defines a score function which measures the distance of two nodes relative to their relation in the mapped embedding space. The goal of these score functions can be summarized as keeping the nodes which are connected to each other in the graph close in the mapped dimension and those which are not connected far from each other. The most famous score functions are TransE, TransR, RESCAL, DistMult, ComplEx, and RotatE [9].

2.2. Deep Learning Applications Applied to Cultural Heritage

In recent years, multiple deep learning applications applied to cultural heritage (CH) have been developed, especially for images of ancient paintings and historical artworks. Most of CH's applications are in the domain of computer vision. One of the major drawbacks of CH applications' implementation is the quality and quantity of annotated datasets available to train and test deep learning algorithms. In general, the data are scarce, and copyrights restrict their reuse and publication. Several historical manuscript image datasets have been proposed [10–12] with the goal of training and evaluating deep learning methods. Digitalized manuscripts datasets are mainly used for document analysis and text recognition [13–15]. As we are interested in ancient musical instruments recognition, we need manuscripts that are illuminated. Among the existing illuminated manuscripts datasets, we can mention the HBA corpus [16], or the HORAE dataset [17], but none of them contain musical instruments. Other datasets containing artistic artworks are PrintArt [18], BAM [19], and OmniArt [20].

Neural networks, ranging from Convolutional Neural Networks (CNNs) to Mask R-CNN [21], are useful in recognizing high-level artworks features from the low-level image features like colors, shapes, and texture. Therefore, they are widely used in CH's applications manipulating images of artworks and paintings, such as object detection [22,23], image classification [18,24,25], image description and captioning [26–28], or answering visual questions [29] in artworks and paintings.

Writer Identification is another field that has attracted attention in this area; it classifies a page of handwritten historical document scans to their original author or artist. There exist two main streams for solving this task: the easier one is building a row-level model followed by a majority classifier using CNNs [30], whereas the other is a facial recognition inspired method which embeds the whole scan and computes distances between the training data and the new samples. Another promising application of deep learning to medieval and historical works is predicting the creating date of artworks, like the approach presented in [31] that used Convolutional Neural Networks and outperformed the traditional rule-based systems used before.

However, the use of deep learning techniques is not always enough to interpret artworks. The vision of a human expert is essential to understand the content and meaning of artworks fully. Therefore, state-of-the-art methods include ontologies and knowledge graphs that model human knowledge to improve neural networks results. These approaches with graph embeddings [32] and graph neural networks [33] aim at creating meaningful vector representations including structural graph information (such as nodes and edges) as well as the content information (such as texts or images) of each node. The method presented in this article combining knowledge graph embeddings with visual embeddings is a clear example of such approaches.

2.3. Domain Adaptation for Image Classification

Domain adaptation (DA) aims at reducing the domain gap between the source and target domains. In mathematical terms, there are two domains: the source domain D^s and the target domain D^t. DA improves the results on a task on the source domain T^s by leveraging the target domain's data and its task T^t. In our case, we are interested in classification tasks, and thus both source and target tasks are classification problems; however, their domains' marginal distributions differ, i.e., $P(X^s) \neq P(X^t)$ [34].

There are two main subfields of domain adaptation which are closed-set and open-set. In closed-set domain adaptation [35,36], the classes of both domains are the same; in other words, $T^t = T^s$ precisely, whereas in open-set domain adaptation, the classes might not be the same, but they share some common labels; in mathematical terms, $T^t \cap T^s \neq \emptyset$. Our work handles the case of open-set domain adaptation. It provides a more challenging scenario and more applicable in real-world cases, especially in cultural heritage datasets that rarely share the same classes between datasets as different tools and instruments were used in different historical periods.

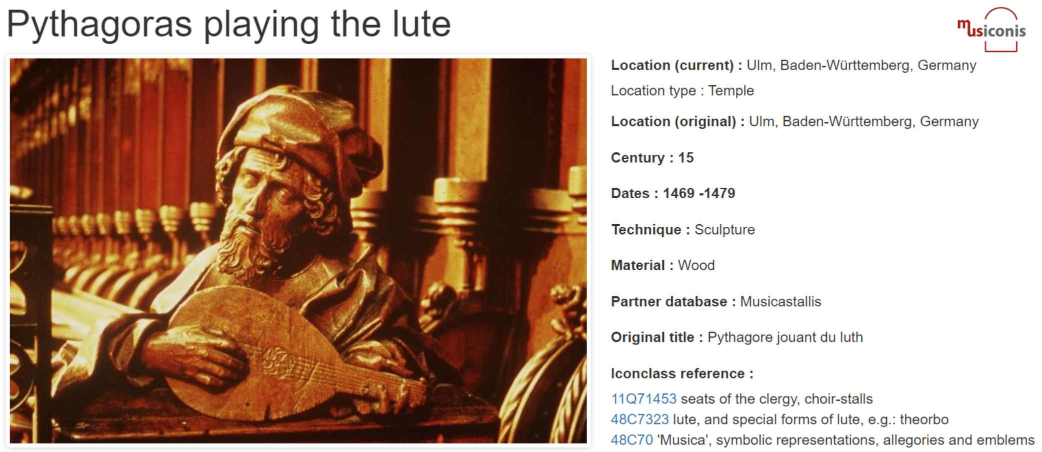
3. Cultural Heritage Datasets

We evaluated our KG embedding-based domain adaptation approach on music iconographical data. The analysis of ancient artworks containing musical instruments brings valuable information on the instruments' nature, physical characteristics, or playing methods. In this section, we present the image datasets and the knowledge graph used to test our proposal.

3.1. Medieval and Renaissance Musical Iconography as Source and Target Domains

As we mentioned before, transfer learning aims to improve a model's performance on a target domain by reusing a trained model on an already-known source domain. This article uses an image dataset of medieval musical instruments as the source domain and a renaissance musical instruments database as the target domain. Previously in [37], we presented a new manually annotated image dataset of historical musical instruments and a non-intrusive Transfer Learning method for object detection. While in [38], we proposed another method for unsupervised domain adaptation, which starts by applying style transformations to the input images and train a transformation discriminator module to predict these style changes. Based on these previous articles, we reused the lessons learned to detect chordophones in medieval artworks, to detect herein vihuelas (a Spanish renaissance chordophone) from a small collection of images. In this article, we combined knowledge graph embeddings with visual embeddings from the images and trained a neural network with the combined embeddings to take our methods a step forward.

Musiconis database (http://musiconis.huma-num.fr/en/, accessed on 30 July 2021): Musiconis is a meta-database containing a vast collection of artworks images from the Middle Ages. Currently, it is the largest collection of musical iconography representing sound and music between the 5th century and the 15th century. The original images come from several international partner institutions or museums, such as the National French Library in Paris, the Courtauld Institute of Art in London, or the Metropolitan Museum of Art in New York. The Musiconis database contains 2154 iconographic representations whose scenes not only contain musical but also vocal, acrobatic or choreographic performances. Musicologists deeply analyzed these scenes, and each musical instrument was carefully described with organological details. Figure 2 depicts a Musiconis' example of a wood sculpture in a choir (area of a church that provides seating for the clergy and church choir) in which Pythagoras plays the lute (http://musiconis.huma-num.fr/en/fiche/198/pythagore.html, accessed on 30 July 2021). From all the available images in Musiconis, we annotated a subset of 662 chordophones as the objective is to detect vihuelas, a small Renaissance chordophone. The image distribution is as follows: 112 are Citharas, 132 harps, 56 lutes, 75 lyres, and 327 vielles (217 vielles played with a bow).



Pythagoras playing the lute

Musiconis

Location (current) : Ulm, Baden-Württemberg, Germany

Location type : Temple

Location (original) : Ulm, Baden-Württemberg, Germany

Century : 15

Dates : 1469 -1479

Technique : Sculpture

Material : Wood

Partner database : Musicastallis

Original title : Pythagore jouant du luth

Iconclass reference :

11Q71453 seats of the clergy, choir-stalls
48C7323 lute, and special forms of lute, e.g.: theorbo
48C70 'Musica', symbolic representations, allegories and emblems

Figure 2. An example of artwork in the Musiconis database.

Vihuela database (https://vihuelagriffiths.com/, accessed on 30 July 2021): This database [8] is a collection of artwork images depicting vihuelas from the Renaissance period. The vihuela was a popular string instrument in Spain and its colonies between the 15th and 16th centuries. This musical instrument has a shape similar to the modern guitar but is tuned like a lute. There is also a bowed version called "vihuela de arco", which could be considered a modern violin precursor. Vihuelas also became popular in Italy and Portugal as they were easier to play and tune than other bowed string instruments. Even if there are several similarities between vihuelas and previous medieval chordophones, the principal difference is the vihuela's body shape (like the number eight). Besides, the vihuela's body is made from thin flat pieces of wood, which differs from previous string instruments whose bodies were carved from a single block of wood. From this database, we annotated 165 chordophones in total. As it is a dataset of vihuelas, most of the annotated instruments are from the lutes family. The image distribution is as follows: 130 lutes, 31 vielles (27 with a bow), five harps, and two lyres.

3.2. *MusicKG: A Knowledge Graph of Medieval Musical Iconography*

MusicKG is a multilingual knowledge graph (KG) of medieval musical iconography. This KG models as RDF triples the representations of sound and music in the Musiconis database. Following the W3C recommendations, MusicKG is linked to other popular Knowledge Graphs, such as Wikidata, Getty Vocabularies, Iconclass, MIMO, and Geonames. In MusicKG, not only the artwork characteristics (such as artist, material used, or inception) are modeled, but also all the different scenes inside that artwork (for example, a couple dancing and a musician playing behind). In turn, each scene is described exhaustively by depicting the performer's characteristics (type, genre, clothing, and position), the musical instrument's characteristics (type, family, and material), the sound created, and the analogies, if any. Even though MusicKG is an extensive graph of relationships between performances and iconographic entities, we decided to use only a subset of all the RDF triples to create embeddings. Using a smaller graph allows us to better visualize and interpret the results. Once our approach efficiency has been proven, we plan to use all the available RDF triples to exploit the full potential of the KG.

The Musiconis example presented before in Figure 2 is depicted as a MusicKG artwork instance in Figure 3. The main class is **Visual artwork** (herein "artwork") which is connected to the original sources through several predicates: **official website**, **collection**, **inventory number**, and **described at URL**. Furthermore, each artwork instance has a **title** from Musiconis and a title from its original database in the **stated as** property. Generally, several **images** are associated with an artwork to capture all the details from different angles and different resolutions. Regarding dates, each artwork has **start time**, **end time**, and **time period** that indicate the century, the date on which the artist began and finished creating the artwork, respectively. The relation **material used** describes the material an artwork is made of, such as **Wood** or **Ivory** for sculptures; or **Textile** for embroideries and tapestry weavings. The relation **fabrication method** relates an artwork with its **Artistic technique**, such as **Sculpture** or **Painting**.

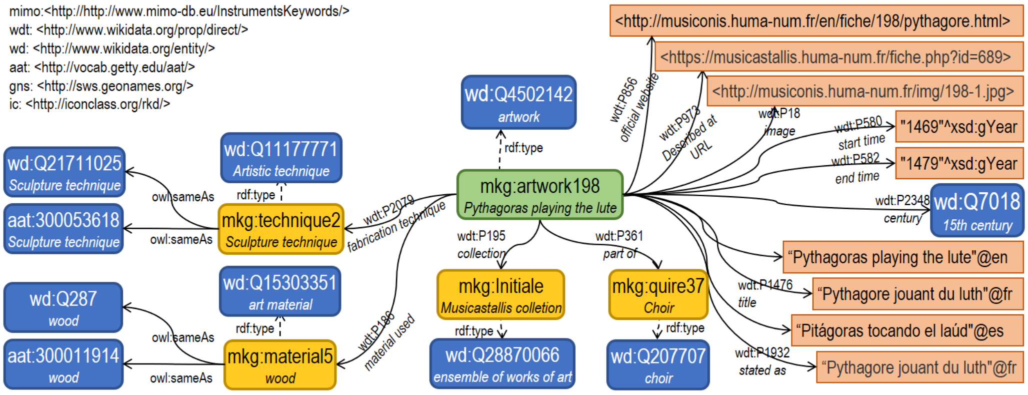

mimo:<http://http://www.mimo-db.eu/InstrumentsKeywords/>
wdt: <http://www.wikidata.org/prop/direct/>
wd: <http://www.wikidata.org/entity/>
aat: <http://vocab.getty.edu/aat/>
gns: <http://sws.geonames.org/>
ic: <http://iconclass.org/rkd/>
<http://musiconis.huma-num.fr/en/fiche/198/pythagore.html>
<https://musicastallis.huma-num.fr/fiche.php?id=689>
<http://musiconis.huma-num.fr/img/198-1.jpg>
wd:Q4502142 artwork
rdf:type
wdt:P856 official website
wdt:P973 Described at URL
wdt:P18 image
wdt:P580 start time
"1469"^xsd:gYear
wdt:P582 end time
"1479"^xsd:gYear
wdt:P2348 century
wd:Q7018 15th century
wd:Q21711025 Sculpture technique
wd:Q11177771 Artistic technique
owl:sameAs
rdf:type
aat:300053618 Sculpture technique
owl:sameAs
mkg:technique2 Sculpture technique
wdt:P2079 fabrication technique
mkg:artwork198 Pythagoras playing the lute
wdt:P195 collection
wdt:P361 part of
"Pythagoras playing the lute"@en
wdt:P1476 title
"Pythagore jouant du luth"@fr
"Pitágoras tocando el laúd"@es
wdt:P1932 stated as
"Pythagore jouant du luth"@fr
wd:Q15303351 art material
wd:Q287 wood
owl:sameAs
rdf:type
aat:300011914 wood
owl:sameAs
mkg:material5 wood
wdt:P186 material used
mkg:Initiale Musicastallis colletion
rdf:type
wd:Q28870066 ensemble of works of art
mkg:quire37 Choir
rdf:type
wd:Q207707 choir

Figure 3. Representation of the artwork instance describing the example of Figure 2.

4. Methodology

This section describes our approach for domain adaptation using knowledge graph embeddings as anchors for our encoders. We start by presenting the terminology used, then detailing the components of our method, and finally describing the losses to train the different components.

Before providing the mathematical functions and different losses, we establish the terminology and annotations used throughout the paper. We denote the source domain as $X^s = (x_i^s, y_i^s)_{i=1}^N$ where x_i^s represent the input images with variant sizes, y_i^s is their respective classes, and N being the size of the source dataset. In this approach, it is important to note that the source domain is associated with a preexisting knowledge graph that connects the images of the dataset with concepts C^s in the graph, creating clusters of data around each concept. Thus, the images are linked together with not just the class information but also along several axes.

The target domain data are referred as $X^t = (x_i^t, y_i^t)_{i=1}^M$, where x_i^t represents the target images, y_i^s represents their respective classes, and M being the size of the target dataset. Domain Adaptation deals with the case where these two datasets share similar classes but usually come from a different distribution (meaning they have some large differences in terms of style, which lead neural networks to be unable to generalize). As our research focuses on open-set domain adaptation, the classes of both domains overlap but do not necessarily have to be the same.

Our model contains two main components typically found in every computer vision classifier: the *Encoder* and the *Classifier* with an additional latent space mapper (the *DimensionMapper*) which converts the visual embeddings into the same dimensions as the knowledge graph embeddings (after training, it is considered to be a final layer of the Encoder). They are both present during the training and the inference phase. We can define our model function f as the composition of the Encoder function e and the classifier function c such that $f = e \circ c$ where $e : \mathcal{X} \longrightarrow \mathcal{Z}$ maps the input images into a vectorial-1D latent space which is considered as the embedding of the visual information of the image, used later to classify it into its corresponding class using $c : \mathcal{Z} \longrightarrow \mathcal{Y}$ which maps the embeddings space into the label space.

Our model's strength comes from a mathematical heuristic used to guide the neural network weight optimization through embedding anchors generated from the associated knowledge graph, which is based on Linear Discriminant Analysis (LDA).

4.1. Architecture

In this subsection, we detail the components of our method from a topology perspective. As shown in Figure 4, the principal components are the *Encoder*, *Classifier*, and *DimensionMapper*.

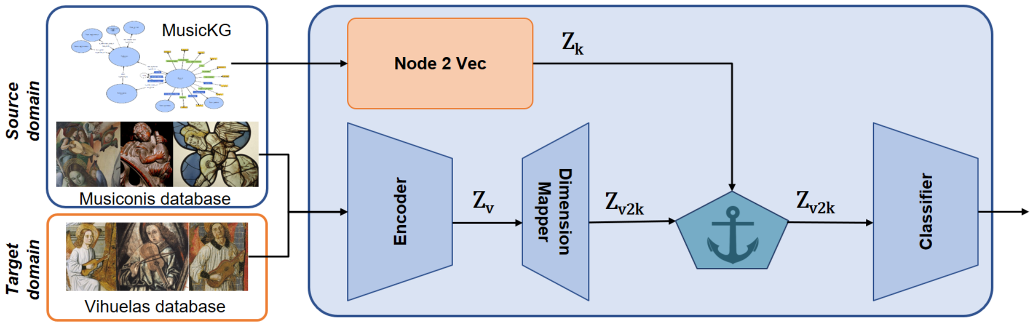


Figure 4. An overview of the proposed approach.

Encoder: Our feature extractor $E(.)$ is a pre-trained pure Convolutional Neural Network extractor with weights W^E. In most cases it should contain only elements such as convolutional layers and max-pooling with a Flattening layer in the end, but as different pre-trained model architectures exist, the components of our model might vary to include things such as residual blocks. The Encoder aims at transferring the shape of the input images from a 2D array with 3 channels to a 1D vector that fully connected neural networks can use, as shown in the following formula:

$$z_v = E(x), x \in X^s \cup X^t, z_v \in R^v \tag{1}$$

In our formulation, $z_v \in Z_v$ is the annotation for the 1D vector space extracted by the pre-trained Encoder, which represents (extracts) the visual embedding of the image. For differentiation between the visual embedding and the knowledge graph embedding, we use a lower case v to index the visual embeddings. The embedding space is a flat vector of the same size for all images. In this case, it is a set of v float numbers such that $v \in [256, 4096]$ depending on the model. The Encoder's output is sent to the space-mapper, which is later used for training the Classifier. The desired behavior for the visual embeddings is to be a class-informative, meaning it extracts information about the label of the object in the image, and domain-independent, meaning images of the same class should be clustered together regardless of their original domain. In the rest of the section, we annotate the Encoder's output of source domain samples as $z^s = E(x^s)$ and the target samples latent space is $z^t = E(x^t)$. In the results section we use an encoder extracted from the pre-trained weights of Resnet 18 trained on ImageNet.

Dimension Mapper: We used a pre-trained model in order to improve the results of our models; this leads to having different latent spaces dimensions. The Dimension Mapper outputs the Encoder embeddings (which is in R^v) into the dimension of the knowledge graph embeddings (which is in R^k) as defined in the following operation:

$$z_{v2k} = DM(z_v) = DM(E(x)), \; x \in X^s \cup X^t \tag{2}$$

Such that our Dimension Mapper is a function that maps the values as follows $dm : \mathcal{R}^v \longrightarrow \mathcal{R}^k$.

Classifier: We are using an Encoder/Classifier scenario. While the Encoder is the only pre-trained part and might include different layer types, the Classifier is a simplified fully-connected neural network (Multi-Layer Perceptron) $C(.)$. This type of Classifier is commonly employed for all image classification cases, such as multi-class classification and binary classification, depending on the chosen final loss function (which might be a binary cross-entropy or cross-entropy). In this work, as our dataset contains several classes, we chose cross-entropy as the loss function. As indicated above, the input of our Classifier is the mapped visual latent space, which was transformed from R^v to R^k where $k < v$ in our case. In contrast, the output is a 1-D probability vector that provides the model's prediction and its probability of belonging to each class $\hat{y}$. The classifier function is defined as follows:

$$\hat{y} = C(z_{v2k}) = C(DM(E(x))), \; x \in X, \; X = X^s \cup X^t \tag{3}$$

In our annotations, we use $\hat{y}$ to reference vector of per-class probabilities, which is the output of the classifier such that $\hat{y} \in \hat{Y}, \; \hat{Y} = \hat{Y}^s \cup \hat{Y}^t$ meaning it is shared between both domains regardless. Before starting the domain adaptation process, we first train the Encoder, Dimension Mapper, and Classifier on the source domain until we reach convergence. Later, we use the Classifier's output for the target domain as pseudolabels, which will help us to better train the model, but only if the Classifier is confident in his decision. We set the threshold θ such that $\theta > 0.95$.

4.2. Losses

This section describes in detail the loss functions that our model uses to train its components.

Classification Loss: As expected, the first loss our model is trained on is the classification loss, which in our case is the cross-entropy loss function $H(,.,)$ that reduces the differences between the probability distributions of the output and the labels, as they are between 0 and 1, according to the following formula:

$$\mathcal{L}_c(W^E, W^C, W^{DM}) = \left(1 * \sum_{x^s \in X^s} H(\hat{y}^s, y^s) + \lambda_t \sum_{x^t \in X^t} H(\hat{y}^t, y^t) \right) \tag{4}$$

We use λ_t as a hyperparameter to control the contribution to the loss between the two domains. This loss affects the training of the Encoder, Classifier, and Dimension Mapper.

Anchoring Loss: The core contribution of our method relies on the anchoring loss, which takes the knowledge graph embeddings as anchors and brings the mapped visual embeddings closer to them, creating richer visual embeddings. This loss aims to embed more information in the Encoder's training without using the data itself as input to the Classifier as it might be missing on some datasets and, most importantly, unavailable on new images. This loss allows our model to combine the increased accuracy of fusion models (which use multiple sources of input data, commonly, images and text, or images and a vector form the tabular data) with the speed and generalizability of one source model used for image classification. This loss is based on a traditional machine learning model, namely, Linear Discriminant Analysis (LDA) and Fisher's linear discriminant aiming to map input data into a space that linearly separates the samples.

Our goal is not to make the latent space linearly separable as this is impossible in the case of image embeddings and might lead to performance degradations. In fact, the objective is only to reduce the distance between the mean of the source latent space z_{v2k} which are linked to that specific concept allowing the latent space of the mapped visual embeddings to contain more rich information about the source images that the Encoder will be forced to focus on and extract. Our loss follows the following formula:

$$\mathcal{L}_{anc}(W^E, W^{DM}) = \left(\frac{\sum_{i\in Y}\sum_{c\in C}(d(\mu^c_{v2k}, a^c_k) + d(\mu^c_{v2k}, z^c_{v2k-i}))}{\sum_{ci\in C}\sum_{cj\in C} d(\mu^{ci}_{v2k}, \mu^{cj}_{v2k})} \right) \times \lambda_{BF} \tag{5}$$

$$\lambda_{BF} = \frac{\min_i |Y^t_i|}{\max_i |Y^t_i|}$$

The goals of this loss are (i) to reduce the distance between the center of mapped visual embeddings of a concept μ^c_{v2k} and their corresponding anchor a^c_k, which is represented in the loss as $(d(\mu^c_{v2k}, a^c_k)$, and (ii) to reduce the distance between the center of mapped visual embeddings of a concept μ^c_{v2k} and its corresponding mapped visual embeddings $z^c_{v2k-i})$, which represented in the loss as $d(\mu^c_{v2k}, z^c_{v2k-i})$. This formulation of the target proved to be faster in training than directly reducing the distance between the embeddings and their anchors one by one, and yet it provides the same gradients in the end. Our anchoring loss's second aim is to augment the distance between the centers of the mapped visual embeddings to create more space between them such that $ci \neq cj$. This loss is only used on different concepts and not the same one.

One downside of using this loss is that it usually has higher values than the classification loss and can influence the direction of classification. Besides, this loss is usually very imbalanced and depends on the random samples taken by the data loader and their classes/concepts, so we added a balancing factor λ_{BF} to reduce its effect when the classes/concepts are not balanced enough.

4.3. Optimization

We can imagine our model trained on the weighted sum of both losses. Considering the different ratios of loss weights (Classification Loss is usually 10–15 times smaller than the Anchoring Loss), the importance of the Classification Loss (which is vital for the classification), and the performances of our final model; we decided to add a balancing parameter β_A, which is usually around [0.01, 0.03]. This optimization is summarized in the following formula:

$$\mathcal{L} = \min_{W_E, W_C, W_D M} 1 * \mathcal{L}_C + \beta_A \mathcal{L}_{anc} \tag{6}$$

For better understanding of the steps of our model, in Algorithm 1 we provide an algorithmic description that depicts the step-by-step operations needed for our method. The first step of our method starts by leveraging a knowledge graph that describes our source dataset, such as the century or the material used to create an artwork. Later, we create an embedding of each artwork based on its connections with the other nodes in the graph using the node2vec algorithm. These artwork level embeddings help us generate concept level embeddings which will be used as the anchors for training our neural networks. The second step is to train the neural network on Classification Loss, and minimize the overall distance between the center of visual concept embeddings and the normalized center of the knowledge graph concept embeddings. This method enables the Encoder part of the network to extract class-informative and structured latent space, allowing the Classifier to generalize better to other domains.

Algorithm 1: Knowledge Graph Embedding based Domain Adaptation.

```
Input: $X^s$—Source domain images.
       $Y^s$—Source domain image labels.
       $KG^s$—Source domain knowledge graph.
       $X^t$—Target domain images.
       $Y^t$—Target domain image labels.
       $\beta_A$—Balancing factor—hyperparameter
Output: $\theta^E$—Weights of the Encoder
        $\theta^{DM}$—Weights of the domain Mapper
        $\theta^C$—Weights of the classifier
// Creating the anchor embeddings $a_k^c$ using node2vec
Sample walks using a random walk from the $KG^s$. ;
Embed the nodes of $KG^s$ using the skip gram model. ;
Generate the $a_k^c$ as the mean of the art work embeddings related to the concept;
// Pre-training The Encoder, Mapper, and classifier on the source
   domain.
for $i \leftarrow 1$ to epochs do
    for $j \leftarrow 1$ to nb_batches do
        Sample a batch of source images $(x_{1s}^j, y_{1s}^j), (x_{2s}^j, y_{2s}^j), ..., (x_{Ns}^j, y_{Ns}^j)$;
        $\theta^E = \theta^E - \alpha \frac{\partial L_C}{\partial \theta^E}$                    Equation (4);
    end
end
// Anchoring the source visual concepts and adapting to the target
for $i \leftarrow 1$ to $I$ do
    Sample a batch of images for both domains $(x^s, y^s, c^s), (x^t, y^t)$;
    Update $W^E$ by deriving $\mathcal{L}_C + \mathcal{L}_{anc}$;
    Update $W^E$ by deriving $\mathcal{L}_C + \mathcal{L}_{anc}$;
    Update $W^C$ by deriving $\mathcal{L}_C$;
end
return $\theta^E, \theta^C$
```

5. Results

This section evaluates our method's ability to embed knowledge graph extracted information to improve the results of image classifiers on complex datasets that suffer from class imbalance and small sample sizes. We used two datasets: the Musiconis dataset with its images, labels, and knowledge graph (MusicKG) and the vihuelas dataset with its images and labels. First, we describe the model's abilities to generalize and present the per-class accuracy metrics against several baselines that do not use knowledge graphs and show that our model improves their results, proving the efficiency of adding knowledge graph data to computer vision deep learning-based models. Second, we evaluate our model's performance when we change the source dataset's size for training to show its sensitivity and resilience. Throughout the results section, all the reported results are the average of 5 runs of the model on the best hyperparameters found using k-fold cross-validation with k = 10. The train–test split is a stratified split with 80% for training and 20% for testing.

5.1. Class Level Evaluation

This subsection presents our enhanced performances against several baselines and shows that knowledge graphs can add value to computer vision models without altering the classification pipeline of classical deep learning-based image classification. We compare our model against three baselines: (1) *SourceOnly*: a deep learning model sharing our architecture but trained only with the images and labels of the source dataset; (2) *TargetOnly*: a deep learning model sharing our architecture but trained only with the images and labels of the target dataset; and (3) *SourceTarget*: a deep learning model sharing our architecture but trained with both the source and the target datasets' images and labels.

We report the f1-scores for every main class in our dataset (Viele, Lute, Bow) and the macro F1-score for the models in Table 1, as it strikes a good balance between precision and recall and evaluates the models much better than accuracy as the sample distribution among the classes differ broadly. We chose to use f1-scores for evaluation instead of accuracy as our dataset suffers from class imbalance and hence accuracy metrics are not very informative about the model's performances. The table clearly shows that our method improves over the three baselines used for comparison. We can also see that the *SourceTarget* model outperforms both baselines as it uses the two datasets. Surprisingly the *SourceOnly* model outperformed the *TargetOnly* model on some classes even though it was not trained on the target data.

Table 1. Per class F1-score comparison between our model and three baselines.

Method	Source Only	Target Only	Source Target	KGE-DA (Ours)	Metric
Viele	64.62	52.16	69.1	**72.18**	F1-score
	58.54	48.09	72.02	73.36	Precision
	72.1	56.98	66.4	71.03	Recall
Lute	53.92	67.14	74.96	**85.63**	F1-score
	50.1	62.96	73.22	82.25	Precision
	58.36	71.91	53.92	89.29	Recall
Bow	57.89	46.03	71.44	**73.06**	F1-score
	62.29	48.19	79.46	76.66	Precision
	54.06	44.05	64.88	69.77	Recall
Avg	58.81	55.11	71.83	**76.96**	F1-score
	56.8	60.55	75.8	74.37	Precision
	60.94	50.56	68.24	79.72	Recall

5.2. *Target Size Evaluation*

In the previous subsection, we proved that our method outperforms the baselines. This subsection shows how our method performs against the *TargetOnly* and *SourceTarget* baselines when the target data's size varies. This comparison is important since our method's principal goal is to use datasets with tiny sample size (the general case of cultural heritage datasets) and still manage to get good results.

As shown in Table 2, our model's performances are always higher than the baselines, even in extreme cases. More importantly, our model's performances were not affected as much as the baselines when reducing the target dataset's sample size. We can also see that the *TargetOnly* baseline was the most affected even though it is the most used technique for small data cases. We can also see that the *SourceTarget* model still gives better performances than *SourceOnly* and *TargetOnly*. However, the drop of performance was significant, especially when going from 100% to 75% where it dropped from 71.83% to a 68.7%, unlike our model that only dropped 1.57% proving our method's flexibility and efficiency even in extreme cases.

Table 2. Performance evaluation based on f1-score of KGE-DA method while varying target data sizes.

Method	Source Only	Target Only	Source Target	KGE-DA (Ours)
30%	58.81	36.14	60.31	**67.03**
45%	58.81	43.49	64.28	**70.26**
60%	58.81	49.26	64.42	**74.86**
75%	58.81	52.97	68.7	**75.39**
100%	58.81	55.11	71.83	**76.96**

6. Conclusions

We presented a new approach to improve state-of-the-art domain adaptation methods using knowledge graph embeddings. We combined knowledge graph embeddings with visual embeddings from the images and trained a neural network with the combined embeddings as anchors. This method is particularly appropriate when dealing with sparse and heterogeneous datasets, like those we generally face in the digital humanities and cultural heritage domain. We evaluated our approach on two cultural heritage datasets of images containing medieval and renaissance musical instruments. The experimental results showed a significant increase in the baselines and state-of-the-art performance compared with other domain adaptation methods. Besides, our model's performances were not affected as much as the baselines when reducing the target dataset's size.

Author Contributions: Conceptualization, V.E. and I.E.I.B.; methodology, V.E. and I.E.I.B.; software, I.E.I.B. and N.D.C.; validation, V.E. and I.E.I.B.; writing—original draft preparation, V.E. and I.E.I.B.; writing—review and editing, V.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yakovlev, K.; Bekkouch, I.E.I.; Khan, A.M.; Khattak, A.M. Abstraction-Based Outlier Detection for Image Data. In *SAI Intelligent Systems Conference*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 540–552.
2. Rivera, A.R.; Khan, A.; Bekkouch, I.E.I.; Sheikh, T.S. Anomaly Detection Based on Zero-Shot Outlier Synthesis and Hierarchical Feature Distillation. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, 1–11. [CrossRef] [PubMed]
3. Bekkouch, I.E.I.; Aidinovich, T.; Vrtovec, T.; Kuleev, R.; Ibragimov, B. Multi-Agent Shape Models for Hip Landmark Detection in MR Scans. In *Medical Imaging 2021: Image Processing*; SPIE: Washington, DC, USA, 2021; Volume 11596, pp. 153–162.
4. Eyharabide, V.; Lully, V.; Morel, F. MusicKG: Representations of Sound and Music in the Middle Ages as Linked Open Data. In *International Conference on Semantic Systems. The Power of AI and Knowledge Graphs*; Acosta, M., Cudré-Mauroux, P., Maleshkova, M., Pellegrini, T., Sack, H., Sure-Vetter, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 57–63.
5. Hyvönen, E. "Sampo" Model and Semantic Portals for Digital Humanities on the Semantic Web; In Proceedings of the 5th Conference of the Digital Humanities in the Nordic Countries (DHN), Riga, Latvia, 21–23 October 2020; pp. 373–378.
6. Meghini, C.; Bartalesi, V.; Metilli, D. Representing narratives in digital libraries: The narrative ontology. In *Semantic Web*; IOS Press: Amsterdam, The Netherlands, 2021; pp. 1–24.
7. Meroño-Peñuela, A.; De Boer, V.; Van Erp, M.; Zijdeman, R.; Mourits, R.; Melder, W.; Rijpma, A.; Schalk, R. CLARIAH: Enabling Interoperability Between Humanities Disciplines with Ontologies. *Appl. Pract. Ontol. Des. Extr. Reason.* **2020**, *49*, 73.
8. Griffiths, J. At Court and at Home with the Vihuela de mano: Current Perspectives on the Instrument, its Music, and its World. *J. Lute Soc. Am.* **1989**, *22*, 1–27.
9. Wang, Q.; Mao, Z.; Wang, B.; Guo, L. Knowledge Graph Embedding: A Survey of Approaches and Applications. *IEEE Trans. Knowl. Data Eng.* **2017**, *29*, 2724–2743. [CrossRef]

10. Rabaev, I.; Barakat, B.K.; Churkin, A.; El-Sana, J. The HHD Dataset. In Proceedings of the 2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR), Dortmund, Germany, 8–10 September 2020; pp. 228–233.
11. Simistira, F.; Seuret, M.; Eichenberger, N.; Garz, A.; Liwicki, M.; Ingold, R. Diva-hisdb: A precisely annotated large dataset of challenging medieval manuscripts. In Proceedings of the 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), Shenzhen, China, 23–26 October 2016; pp. 471–476.
12. Shen, Z.; Zhang, K.; Dell, M. A Large Dataset of Historical Japanese Documents with Complex Layouts. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 548–549.
13. Hu, P.; Xu, M.; Wu, M.; Chen, G.; Zhang, C. Handwritten Style Recognition for Chinese Characters on HCL2020 Dataset. In Proceedings of the Chinese Conference on Pattern Recognition and Computer Vision (PRCV), Nanjing, China, 16–18 October 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 138–150.
14. Pondenkandath, V.; Alberti, M.; Eichenberger, N.; Ingold, R.; Liwicki, M. Cross-Depicted Historical Motif Categorization and Retrieval with Deep Learning. *J. Imaging* **2020**, *6*, 71. [CrossRef]
15. Valy, D.; Verleysen, M.; Chhun, S.; Burie, J.C. A new khmer palm leaf manuscript dataset for document analysis and recognition: Sleukrith set. *Int. Workshop Hist. Doc. Imaging Process.* **2017**, 1–6. [CrossRef]
16. Mehri, M.; Héroux, P.; Mullot, R.; Moreux, J.P.; Coüasnon, B.; Barrett, B. HBA 1.0: A pixel-based annotated dataset for historical book analysis. In Proceedings of the 4th International Workshop on Historical Document Imaging and Processing, Kyoto, Japan, 10–11 November 2017; pp. 107–112.
17. Boillet, M.; Bonhomme, M.L.; Stutzmann, D.; Kermorvant, C. HORAE: An annotated dataset of books of hours. *arXiv* **2019**, arXiv:2012.00351.
18. Carneiro, G.; Da Silva, N.P.; Del Bue, A.; Costeira, J.P. Artistic image classification: An analysis on the printart database. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; Springer: Berlin/Heidelberg, Germany, 2012; pp. 143–157.
19. Wilber, M.J.; Fang, C.; Jin, H.; Hertzmann, A.; Collomosse, J.; Belongie, S. Bam! the behance artistic media dataset for recognition beyond photography. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1202–1211.
20. Strezoski, G.; Worring, M. Omniart: Multi-task deep learning for artistic data analysis. *arXiv* **2017**, arXiv:1708.00684.
21. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R.B. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
22. Gonthier, N.; Gousseau, Y.; Ladjal, S.; Bonfait, O. Weakly supervised object detection in artworks. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018.
23. Kadish, D.; Risi, S.; Løvlie, A.S. Improving Object Detection in Art Images Using Only Style Transfer. *arXiv* **2021**, arXiv:2102.06529.
24. Milani, F.; Fraternali, P. A Dataset and a Convolutional Model for Iconography Classification in Paintings. *J. Comput. Cult. Herit. JOCCH* **2021**, *14*, 1–18. [CrossRef]
25. Castellano, G.; Vessio, G. Deep convolutional embedding for digitized painting clustering. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 2708–2715.
26. Wang, A.; Hu, H.; Yang, L. Image captioning with affective guiding and selective attention. *ACM Trans. Multimed. Comput. Commun. Appl. TOMM* **2018**, *14*, 1–15. [CrossRef]
27. Sheng, S.; Moens, M.F. Generating captions for images of ancient artworks. In Proceedings of the ACM International Conference on Multimedia, Nice, France, 21–25 October 2019; pp. 2478–2486.
28. Cetinic, E. Iconographic image captioning for artworks. *arXiv* **2021**, arXiv:2102.03942.
29. Garcia, N.; Ye, C.; Liu, Z.; Hu, Q.; Otani, M.; Chu, C.; Nakashima, Y.; Mitamura, T. A dataset and baselines for visual question answering on art. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 92–108.
30. Cilia, N.; De Stefano, C.; Fontanella, F.; Marrocco, C.; Molinara, M.; Di Freca, A.S. An end-to-end deep learning system for medieval writer identification. *Pattern Recognit. Lett.* **2020**, *129*, 137–143. [CrossRef]
31. Hamid, A.; Bibi, M.; Moetesum, M.; Siddiqi, I. Deep Learning Based Approach for Historical Manuscript Dating. In Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, Australia, 20–25 September 2019; pp. 967–972.
32. Gesese, G.A.; Biswas, R.; Alam, M.; Sack, H. A survey on knowledge graph embeddings with literals: Which model links better literally? In *Semantic Web*; IOS Press: Amsterdam, The Netherlands, 2019; pp. 1–31.
33. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Philip, S.Y. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 4–24. [CrossRef] [PubMed]
34. Pan, S.J.; Yang, Q. A Survey on Transfer Learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [CrossRef]
35. Bekkouch, I.E.I.; Youssry, Y.; Gafarov, R.; Khan, A.; Khattak, A.M. Triplet Loss Network for Unsupervised Domain Adaptation. *Algorithms* **2019**, *12*, 96. [CrossRef]
36. Batanina, E.; Bekkouch, I.E.I.; Youssry, Y.; Khan, A.; Khattak, A.M.; Bortnikov, M. Domain Adaptation for Car Accident Detection in Videos. In Proceedings of the 2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA), Istanbul, Turkey, 6–9 November 2019; pp. 1–6.

37. Bekkouch, I.; Eyharabide, V.; Billiet, F. Dual Training for Transfer Learning: Application on Medieval Studies. In Proceedings of the International Joint Conference on Neural Networks, Virtual Event, 18–22 July 2021.
38. Bekkouch, I.; Constantin, N.D.; Eyharabide, V.; Billiet, F. Adversarial Domain Adaptation for Medieval Instrument Recognition. In Proceedings of the SAI Intelligent Systems Conference, Amsterdam, The Netherlands, 2–3 September 2021.