ASSESSING THE SEMANTIC SIMILARITY OF IMAGES OF SILK FABRICS USING CONVOLUTIONAL NEURAL NETWORKS

D. Clermont*, M. Dorozynski, D. Wittich, F. Rottensteiner

Institute of Photogrammetry and GeoInformation, Leibniz Universität Hannover, Germany
(clermont, dorozynski, wittich, rottensteiner)@ipi.uni-hannover.de

Commission II, WG II/8

KEY WORDS: Convolutional Neural Networks, Image similarity, Cultural heritage, Silk fabrics, Incomplete training samples

ABSTRACT:

This paper proposes several methods for training a Convolutional Neural Network (CNN) for learning the similarity between images of silk fabrics based on multiple semantic properties of the fabrics. In the context of the EU H2020 project SILKNOW (http://silknow.eu/), two variants of training were developed, one based on a Siamese CNN and one based on a triplet architecture. We propose different definitions of similarity and different loss functions for both training strategies, some of them also allowing the use of incomplete information about the training data. We assess the quality of the trained model by using the learned image features in a k-NN classification. We achieve overall accuracies of 93-95% and average F1-scores of 87-92%.

1. INTRODUCTION

The main goal of the EU H2020 project SILKNOW (http://silknow.eu/) is to support art historians in improving their understanding of European silk heritage, as well as making this knowledge available to the public. Openly accessible databases like (IMATEX, 2018) collect information about such fabrics, but not in a standardized format. Thus, in the context of the project, the information from different collections is collected in a uniform database with standardized annotations. Relevant properties of fabrics include the production time, place or technique. One way to access this knowledge is to make database queries, e.g. to get a list of records related to fabrics produced in the 19^{th} century. The alternative investigated in this paper is to query the records that are most similar to a given *image*, a procedure known as *image retrieval*, e.g. (Zheng et al., 2017). Given an image of a fabric with unknown origin, this would be a way to learn something about the fabric, because the query results also give access to the properties of the most similar images. However, this leads to the question of how to define the similarity of silk fabrics. Existing methods for characterizing the similarity of images are often only based on visual appearance, e.g. (Wang et al., 2016; Jamil et al., 2006). Supervised learning of a model of similarity (Hadsell et al., 2006; Schroff et al., 2015) requires training images such that for each image pair we know whether the images are similar or not. This information is not readily available in a database containing records of fabrics, so that manual annotation would be required, an expensive and very subjective task. An alternative, explored in another context in (Zhao et al., 2015), is to define similarity based on the *similarity of properties* of the depicted fabrics. As information about these properties is available in the database, training samples can be generated automatically using this approach. It may also be more useful in the context of the project SILKNOW, because the most similar images according to this definition may be those from which a user may learn most about the fabric depicted in the query image.

Consequently, this paper presents a method for training a Convolutional Neural Network (CNN) (Krizhevsky et al., 2012) to learn a model of the similarity between images of silk fabrics based on multiple semantic properties of the fabrics, which allows us to model different degrees of similarity. This is different from most existing similarity definitions, which only consider one such property, e.g. (Gordo et al., 2016). The network learns to generate an image descriptor such that the distance of the descriptors of similar images is small. We propose two training strategies for that purpose, one based on a Siamese architecture (Bromley et al., 1994) and another one considering image triplets (Schroff et al., 2015). The training samples are generated automatically from a database of images with annotations. However, existing databases of silk fabrics often contain many samples with incomplete annotations, i.e. information about some semantic properties may be missing. Consequently, for both training scenarios, we will define loss functions that can also cope with such samples, the main problem being that a definition of similarity of image pairs based on annotations will be affected by missing information. In our experiments, we compare the different learning scenarios and we investigate the impact of our new developments on the results. For a quantitative evaluation, we assess the performance of a k-nearest neighbour (k-NN) classifier (Bishop, 2006) based on the Euclidean distance of the feature vectors.

The scientific contribution of this paper is three-fold. Firstly, we define a model of the similarity of images of silk fabrics based on semantic properties. To the best of our knowledge, this is the first work considering multiple properties for that purpose. Secondly, based on this definition of similarity, we develop two strategies for learning a model of image similarity with automatically generated training samples. Finally, for both training strategies, we develop loss functions that can deal with incompletely labelled samples, which gives access to a considerably larger set of training data.

2. RELATED WORK

Learning the similarity of pairs of images is not an entirely new problem in the fields of Photogrammetry and Computer Vision.

*Corresponding author

It is often times faced in the context of feature-based image matching, e.g. (Han et al., 2015), and image retrieval, e.g. (Qi et al., 2016). Usually, the similarity of images is assessed via image descriptors (feature vectors), the idea being that the descriptors of similar images should have a small distance in feature space. While originally hand-crafted descriptors were used, in the meantime the focus of research has been shifted to learning descriptors based on CNNs (Zheng et al., 2017).

In the context of image matching, one task is to find pairs of images that show the same scene and, thus, overlap. Han et al. (2015) train a Siamese CNN to correctly predict whether two image patches are similar or not. They define similarity in a binary way; images are only considered to be similar if they show the same scene. They train their network by minimizing the cross-entropy error of the network's binary predictions of similarity. This training strategy does not use the information that descriptors for similar feature vectors should have a small distance in an explicit way. While the representation may be optimal for binary classification, using the distance of feature vectors for assessing similarity may yield sub-optimal results. Furthermore, the CNN is trained from scratch, whereas Babenko et al. (2014) have shown that image descriptors delivered by pre-trained networks are well suited for image retrieval even if the networks were trained for classification. Using a pre-trained network could improve the matching performance or at least reduce the requirements with respect to training samples.

In the context of image retrieval, Qi et al. (2016) proposed a method to retrieve photos from a database when only a free-hand drawn sketch is available. The authors propose a Siamese CNN architecture consisting of two CNN branches with shared parameters. In training, an image and a sketch are processed by the two branches, respectively, and the loss function tries to minimize the distance of the resultant image descriptors for pairs that are labelled as being similar, and vice versa. Similarity is defined in a binary way: both the photos and the sketches are manually labelled into various shape classes; a photo and a sketch are considered to be similar if their labels are identical. A binary definition of similarity might be disadvantageous because photos and sketches could belong to multiple shape classes. A non-binary definition of similarity could capture different degrees of similarity, e.g. the number of matching shapes, which could increase the retrieval performance.

Gordo et al. (2016) as well as Wang et al. (2014) retrieve photos from a database that are similar to queried photos. Both papers use a network architecture similar to a Siamese one, but extend it by a third network branch also sharing its weights with the other ones. This three-stream architecture is trained using a triplet ranking loss requiring descriptors from similar images to be closer to each other than descriptors from dissimilar images. For training, the authors use a large public dataset of images of famous landmark sites. Again, the similarity is defined in a binary way; pairs of images are considered to be similar if they show the same site. Such a definition would not be directly applicable to the problem considered in this paper, because hardly any pair of images in a database would show the same fabric.

Zhao et al. (2015) propose a method for learning binary image descriptors for multi-label image retrieval. They use CNNs to jointly learn feature representations and their mappings to binary hash codes. Although their research regarding binary descriptors is rather unrelated to our own research, their approach to consider multiple labels for learning image similarity is of great importance for us. In contrast to (Qi et al., 2016) and (Gordo et al., 2016), where *similarity* between two images is a binary variable, Zhao et al. (2015) model similarity in a non-binary way. Working with images with labels for multiple properties, similarity is defined based on the number of matching labels of two images; the higher the number of matching labels, the higher the similarity. While the authors claim that the images used for training can have a varying number of labels, which would be relevant for our problem of having to deal with incomplete samples, we argue that their approach, being based on absolute numbers, might deliver counter-intuitive results if there are large variations in the number of labels per sample. We tackle the problem of having differing numbers of variables by introducing a concept of uncertainty of similarity.

The classification of works of art using Deep Learning has been tackled for some years, too. Whereas some papers deal with the prediction of single properties such as the epoch of a painting (Hentschel et al., 2016), others try to predict multiple properties at once based on the assumption that there are interdependencies between the properties (Long et al., 2017). However, image retrieval for works of art using Deep Learning seems to be a much less investigated field, one example being the matching of papyrus fragments (Pirrone et al., 2019). Interestingly, another work in this field is mostly related to ours in terms of the application and also has the goal of retrieving images of silk fabrics (Jamil et al., 2006). The authors argue that the similarity between those fabrics can be assessed by means of visual appearance, more precisely by the motifs depicted in the fabrics. They focus on the shape of the motifs rather than the colour and chose a set of hand-crafted features to define an image descriptor. Similarly to the assessment of (Zheng et al., 2017), we expect descriptors learned from training data to lead to better results than hand-crafted ones. Furthermore, modelling the similarity of silk fabrics purely based on the motifs' shapes might not capture all aspects of similarity of such silk fabrics.

To the best of our knowledge, there is no publication focussing on image retrieval based on the similarity of multiple properties of works of art. Also, there seems to be hardly any work that deals with missing labels when multiple labels are considered for the similarity of images; we believe that the exception (Zhao et al., 2015) gives counter-intuitive results when there is a large variability of the number of categories to compare in the dataset.

3. METHODOLOGY

It is the goal of our method to train a CNN to deliver similar features for similar input images and dissimilar features for dissimilar ones, so that the Euclidean distance of feature vectors can be used to measure similarity between pairs of images. We assume a database of images with annotations for a series of semantic variables to be available. These annotations are used to define a measure of similarity between pairs of images, so that the training samples, consisting of image pairs with known similarity values, can be generated automatically from the database contents. Having trained the CNN, a feature vector can be derived for every sample of the database by passing the corresponding image through the CNN. The resultant feature vectors are used to build a k-d tree (Pedregosa et al., 2011). Given a query image, the most similar images from the database can be retrieved by applying the CNN to that image and retrieving the k nearest neighbours of the resultant feature vectors from the k-d tree. The results can be presented to a user; optionally, the properties of the query image can be predicted by a majority vote of the nearest neighbours. While this classification is not

the main goal of the proposed method, it will be the basis of the quantitative evaluation in section 4. The details of our method are presented in the subsequent sections.

3.1 Network Architecture

Our CNN architecture based on ResNet-152 (He et al., 2016) is presented in figure 1. The input consists of an RGB image x scaled to 224 x 224 pixels. This image is presented to the ResNet-152, which generates a 2048-dimensional feature vector. This is followed by a fully connected (FC) layer of dimension 1024 with ReLU (Rectified Linear Unit) activations (Nair & Hinton, 2010). The last layer is another FC layer which delivers a 128-dimensional vector. In order to restrict the extents of the feature space, this vector is normalized to unit length, which results in the feature vector $f(x)$ which is the main output of the CNN and which should characterize the input image. As a consequence of normalization, the maximum Euclidean distance of two feature vectors is 2, which will be useful to tune some of the parameters of the loss functions described in section 3.3. Both the choice of the ResNet-152 as a backbone and the architecture of the remaining parts of the network was based on preliminary experiments not reported here for lack of space.

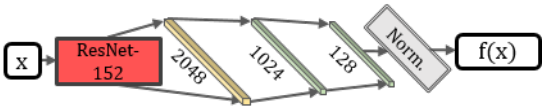

Figure 1. CNN architecture. An image x is passed through a ResNet-152 and two FC layers with 1024 and 128 dimensions, respectively. Normalization (Norm.) of the output of the last FC layer delivers a 128-dimensional feature vector $f(x)$.

3.2 Semantic Similarity

There is no unique definition of the term *similarity* of images, let alone of images of silk fabrics. For reasons already pointed out in section 1, we prefer a definition of similarity based on semantic properties over a definition based on visual appearance. Such a definition allows us to automatically generate training samples from a database of images with semantic annotations, while a definition based on visual similarity would require manual labelling of pairs of images as being similar or not. Manual labelling is highly subjective and might lead to inconsistent annotations, so that we consider a definition based on semantic properties also to be more objective than the other option.

Our definition of similarity requires the availability of a set of images x with annotations for a set of semantic properties such as their production timespan or production place. Further, let $l_i(x)$ be the class label of the i^{th} property for image x. Then, for a pair of images x_1, x_2, we can define a *similarity function* $Y(x_1, x_2)$ which returns a value of 1 if the images are similar and 0 otherwise. This function can be defined in a straightforward way if only a single property l is considered:

$$Y(x_1, x_2) = \delta(l(x_1) = l(x_2)). \tag{1}$$

In eq. 1, $\delta(\cdot)$ is the Kronecker delta function, which returns 1 if the argument is true and zero otherwise. Thus, $Y(x_1, x_2) = 1$ if the class labels for property l are indentical and $Y(x_1, x_2) = 0$ otherwise. A naïve way of considering multiple properties at once would be to check whether *all* property labels are equal. Similarly to the definition in eq. 1, this would lead to a binary-valued similarity function $Y(x_1, x_2) \in \{0, 1\}$. However, we prefer to be able to model different degrees of similarity. We argue that a pair of images with identical annotations in all but a few properties should be considered to be more similar than a pair without any identical annotations. Consequently, we define a real-valued similarity function $Y(x_1, x_2) \in [0, 1]$ whose value is proportional to the number of identical annotations:

$$Y(x_1, x_2) = \frac{1}{I} \sum_{i=1}^{I} \delta(l_i(x_1) = l_i(x_2)), \tag{2}$$

where I is the number of semantic properties. Note that this is equivalent to eq. 1 for $I = 1$.

Eq. 2 is the basic definition of similarity which will be used for training our CNN. However, it requires annotations to be available for all properties, which is not necessarily the case. We could apply this function to incomplete samples, i.e., pairs of images for which a part of the properties under consideration is unknown, by just considering the properties for which annotations are available for both images and setting I to the number of such properties. However, under these circumstances, a pair of images for which only one property is annotated would be considered to be similar with $Y(x_1, x_2) = 1$, although in fact they might differ in all the other (unknown) properties. Obviously, the fact that some properties are unknown introduces some uncertainty into our definition of similarity. The more properties are unknown, the larger this uncertainty is. Thus, in order to be able to include incomplete samples in the training process, we expand our similarity function in order to incorporate this uncertainty. For that purpose, we note that the similarity function $Y(x_1, x_2)$ can serve as an indicator Y_p for *positive similarity*, i.e., $Y_p(x_1, x_2) \equiv Y(x_1, x_2)$. Similarly, we can define an indicator Y_n for *negative similarity*. If x_1 and x_2 are complete samples, i.e., if all properties are known for these images, we can define $Y_n(x_1, x_2) = 1 - Y(x_1, x_2)$. Under these circumstances, we have $Y_p + Y_n = 1$, and there is no uncertainty. In order to expand our notion of similarity to incomplete samples, we use a new definition of $Y_p(x_1, x_2)$ and $Y_n(x_1, x_2)$:

$$\begin{aligned} Y_p(x_1, x_2) &= \frac{1}{I} \sum_{i}^{I} \delta(l_i(x_1) = l_i(x_2)) \cdot \pi_i^1 \cdot \pi_i^2 \\ Y_n(x_1, x_2) &= \frac{1}{I} \sum_{i}^{I} \delta(l_i(x_1) \neq l_i(x_2)) \cdot \pi_i^1 \cdot \pi_i^2, \end{aligned} \tag{3}$$

where

$$\pi_i^n = \begin{cases} 0 & l_i(x_n) \text{ is not known} \\ 1 & l_i(x_n) \text{ is known} \end{cases} \tag{4}$$

indicates whether an annotation for property i exists for image x_n or not. This definition is equivalent to eq. 2 for pairs of complete samples. For incomplete samples, the relative sizes of Y_p and Y_n still express whether two images are more or less similar, but in this case, eq. 3 results in $Y_p + Y_n < 1$. We can interpret $1 - (Y_p + Y_n)$ as a measure of the uncertainty of our knowledge about the similarity of an image pair. The definition of similarity according to eq. 3 is used for training the CNN in the presence of incomplete samples.

3.3 Training

We initialize the parameters of ResNet-152 using the model pre-trained on the ImageNet data set (Deng et al., 2009). The weights of the FC layers are initialized using Variance Scaling (He et al., 2015). In training, we freeze all parameters of

ResNet-152 except those of the last layer. We argue that by using this pre-trained network a good generic feature representation for the images can be obtained (Razavian et al., 2014). The parameters of the last layer of ResNet-152 and the two FC layers of our CNN (cf. fig. 1) are determined in the training procedure. For this purpose, we propose two different strategies, one based on a two-stream Siamese architecture (Bromley et al., 1994) and another one based on a triplet architecture, e.g. (Gordo et al., 2016). For both presented strategies, the training of a neural network consists of minimizing an objective loss function that measures the network's ability for producing similar features for similar input images and dissimilar features for dissimilar input images. We present different loss functions for both training strategies and compare them in our experiments. In all cases, training is based on stochastic minibatch gradient descent with momentum (SGD), using backpropagation for computing the gradients. More details about SGD are given in section 4. The training strategies and the related loss functions are presented in sections 3.3.1 and 3.3.2.

3.3.1 Siamese Training: Our first strategy uses the two-stream Siamese architecture depicted in fig. 2. The training procedure requires pairs of input images x_1, x_2 with known similarity value $Y(x_1, x_2)$. The network takes the two input images and propagates them through two identical copies of our basic CNN architecture to deliver two feature vectors $f(x_1), f(x_2)$ for x_1 and x_2, respectively. Both CNN branches share the same network weights w. The network calculates the L2 distance $\Delta(f(x_1), f(x_2))$ between the two feature vectors, which forms the basis for calculating the loss L. It is the goal of the training procedure to make this distance small for similar image pairs and large for dissimilar image pairs. To achieve this goal, the *contrastive loss* can be used (Hadsell et al., 2006):

$$\begin{aligned} L(x_1, x_2) = Y(x_1, x_2) \cdot max(0, \Delta(f(x_1), f(x_2)) - M_p) \\ +(1 - Y(x_1, x_2)) \cdot max(0, M_n - \Delta(f(x_1), f(x_2))). \end{aligned} \quad (5)$$

In eq. 5, $Y(x_1, x_2)$ is one of the similarity functions described earlier. M_p is the positive distance margin, i.e. the maximum allowed distance of feature vectors of similar inputs, while M_n is the negative distance margin, i.e. the minimum allowed distance of feature vectors of dissimilar inputs. The goal of minimising the loss in eq. 5 is to produce feature vectors having a distance smaller than M_p for samples with $Y = 1$ and larger than M_n for samples with $Y = 0$. In the standard case of the contrastive loss, the function Y is binary, which is also the case for the similarity function in eq. 1; it can be used to train a model based on the similarity of a single property. As we want all of our training samples to always contribute to the training procedure, unless otherwise noted we always pull the distance of features between similar inputs towards the minimum possible distance of 0 and always pushing the distance of features between dissimilar inputs towards the maximum possible distance of 2; this corresponds to choosing $M_p = 0$ and $M_n = 2$. The normalization of the feature vectors helps to define M_n.

The contrastive loss according to eq. 5 also works for a definition of similarity based on multiple properties. In this case, we have to use the similarity function from eq. 2. In the case of a binary similarity function, every training sample will either pull the distance towards M_p or outside of M_n; here, for each image pair, we consider the loss to be a trade-off of two competing forces. One force, weighted by Y, pulls the distance towards M_p, while the other force, weighted by $(1 - Y)$ tries to make the distance larger than M_n. The similarity defines the

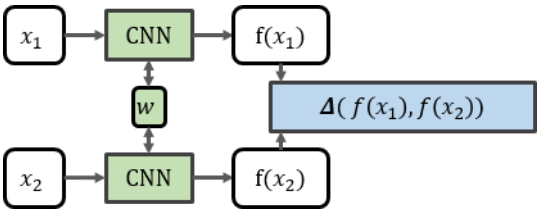

Figure 2. Siamese architecture. Two images are propagated through the same CNN architecture. Both CNNs share their weights w. The L2 distance Δ of the resulting feature vectors $f(x_1), f(x_2)$ is used to calculate the training loss.

weights and, thus, the relative size of Y and $(1 - Y)$ will indicate an equilibrium distance that minimizes the loss. The larger the similarity, the closer this distance will be to M_p. Consequently, we expect the CNN to learn to produce feature vectors whose distances will correspond to the degree of similarity of image pairs according to eq. 2.

The contrastive loss in eq. 5 assumes the values of the similarity function Y to be certain, which is only the case for complete samples (cf. section 3.2). To be able to use incomplete training samples as well, we have to adapt that loss function to consider the uncertainty of the similarity:

$$\begin{aligned} L(x_1, x_2) = Y_p \cdot max(0, \Delta(f(x_1), f(x_2)) - M_p) \\ +Y_n \cdot max(0, M_n - \Delta(f(x_1), f(x_2)). \end{aligned} \quad (6)$$

The only difference between eqs. 5 and 6 is that in the latter, we use the indicators for positive (Y_p) and negative (Y_n) similarity according to eq. 3 instead of Y and $(1 - Y)$ as weights for the two terms in the loss function. Again, the relative size of the weights will determine the point of equilibrium for minimizing the loss, but as the sum of the weights, $(Y_p + Y_n)$, is smaller than 1 (cf. section 3.2), the total impact of a sample on the gradients will be smaller, which means that more uncertain samples have a smaller influence on the training process, which is rather intuitive. We push this thought further by also adapting the margins M_p and M_n from eq. 6, also making them dependent on the degree of uncertainty of the similarity of a sample:

$$\begin{aligned} M_p(x_1, x_2) &= \frac{1}{I} \sum_i^I (1 - \pi_i^1 \cdot \pi_i^2) \\ M_n(x_1, x_2) &= 2 - \frac{1}{I} \sum_i^I (1 - \pi_i^1 \cdot \pi_i^2), \end{aligned} \quad (7)$$

where π_i^1 and π_i^2 are defined according to eq. 4. For complete samples, this results in $M_p = 0$ and $M_p = 2$, as in the earlier case. For incomplete samples, M_p and M_n will be placed symmetrically around 1, because $M_n = 2 - M_p(x_1, x_2)$. The larger the number of properties without annotations, the larger M_p and the smaller M_n. The force pulling the distance towards 0 will only act if the distance is larger than M_p, and the one pushing the distance away from 0 only acts as long as it is smaller than M_n. The larger M_p and, thus, the smaller M_n, the smaller the impact of a sample on the minimization process. Thus, by adapting the two radii according to eq. 4, the uncertainty of the similarity information is again used to modulate the impact of a training sample on the resultant parameters.

3.3.2 Triplet Training: Our second training strategy uses the triplet architecture depicted in fig. 3. The network takes

three input images x_a, x_p, x_n and propagates each of them through a CNN branch. Again, all branches share their weights w. Consequently, this approach requires triplets of training samples, but like the first strategy, it only requires similarity information for image pairs. In this context, x_a is the anchor sample, x_p is a 'positive' sample, meaning that $Y(x_a, x_p) = 1$, and x_n is a 'negative' sample, meaning that $Y(x_a, x_n) = 0$. The CNN branches deliver three feature vectors $f(x_a), f(x_p), f(x_n)$, from which the L2 distances $\Delta(f(x_a), f(x_p))$ and $\Delta(f(x_a), f(x_n))$ are calculated. For determining the parameters of the network, the *triplet loss function* (Schroff et al., 2015) can be applied:

$$L(x_a, x_p, x_n) = \\ max(0, M + \Delta(f(x_a), f(x_p)) - \Delta(f(x_a), f(x_n))), \quad (8)$$

where $L(x_a, x_p, x_n)$ is the loss and M is the margin, which can in principle be chosen freely; during training, the difference of feature distances between $\Delta(f(x_a), f(x_p))$ and $\Delta(f(x_a), f(x_n))$ is pushed to be at least M. In other words, a network should learn to deliver feature vectors $f(x_a), f(x_p)$ that are more similar to each other than the feature vectors $f(x_a), f(x_n)$, meaning that the feature vectors for similar image pairs only need to be *more* similar than the features for a pair being not similar. Note that this loss function only works for a binary definition of similarity according to eq. 1. In other words, it can only be applied when a single property is considered for defining similarity.

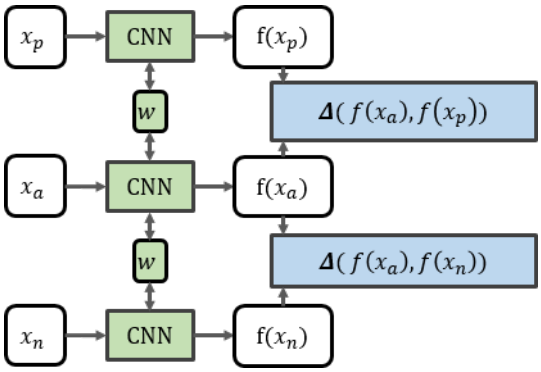

Figure 3. Triplet architecture. Three input images are propagated through the same CNN architecture. The CNN branches share their weights w. The L2 distances $\Delta(f(x_a), f(x_p))$ and $\Delta(f(x_a), f(x_n))$ of the resulting feature vectors are the basis for calculating the loss to be minimized in training.

In order to consider multiple properties per sample, we have to adapt the selection of the samples x_p and x_n and the definition of the margin M. This can be expressed by defining M to be a function of the input samples, thus $M = M(x_a, x_p, x_n)$:

$$M(x_a, x_p, x_n) = Y(x_a, x_p) - Y(x_a, x_n) \overset{!}{>} 0. \quad (9)$$

In eq. 9, the function $Y(\cdot)$ is the similarity function defined in eq. 2. The restriction $M \overset{!}{>} 0$, meaning that M **must** be larger than 0, is important for a triplet of input samples to be a valid triplet, i.e. the anchor and positive samples have to be more similar to each other than the anchor and negative samples. The margin ensures that the distance between descriptors of similar images is smaller than that of descriptors of dissimilar images. We can use the triplet loss function defined by eqs. 8 and 9 only with samples for which all labels are known. In order to also use incomplete samples, we have to redefine the margin M:

$$M(x_a, x_p, x_n) = min(Y_p(x_a, x_p), Y_n(x_a, x_n)) \overset{!}{>} 0, \quad (10)$$

using the definitions from eq. 3 for Y_p and Y_n. In this case, the margin M also represents the uncertainty for the similarity; the larger the uncertainty (i.e. the more labels are unknown), the smaller the margin. This means that the descriptors for pairs of similar and dissimilar images are allowed to be close to each other if the number of available annotations is small.

An important aspect of the training procedure is the definition of the image triplets. Given a minibatch of size B, we calculate the margin $M(x_i, x_j, x_k)$ for every triplet of samples $\{x_i, x_j, x_k\}$ with $i, j, k \in \{1, ..., B\}$ using either eq. 10 or eq. 9, depending on whether incomplete samples are used or not. In this process, we make sure that all images of a triplet are different. Of the remaining triplets, we retain those fullfilling the restriction $M \overset{!}{>} 0$ and use them for training.

4. EXPERIMENTS

4.1 Dataset and test setup

4.1.1 Dataset: To evaluate our proposed methods we use data extracted from the publicly available database of the Centre de Documentació i Museu Tèxtil in Terrassa (Spain) (IMATEX, 2018). This database consists of thousands of RGB images of silk fabrics with annotations about their semantic properties; we exemplarily consider the three variables *production place*, *production technique* and *production timespan*. The annotations for these properties are incomplete, so that there is a considerable number of incomplete samples. The dataset used in this paper is identical to the one used by Dorozynski et al. (2019). It was generated automatically from the online collection (IMATEX, 2018); in this process, the raw annotations were mapped to a standardized class structure. For details of the procedure the reader is referred to (Dorozynski et al., 2019). The dataset consists of 8192 images for which at least one property is known. We call it the *comprehensive* set, because it contains both the 5071 incomplete and 3121 complete samples (for which *all* properties are known). All images are scaled such that the larger dimension (height or width) is exactly 400 pixels; the other, possibly smaller, dimension varies between 25 and 400 pixels. The class structure as well as the number of samples that are available for the individual classes are shown in tab. 1.

	Class name	Complete	Comprehensive
PL	*Catalonia* (*C*)	2727	4322
	Spain (Rest) (*S*)	394	2671
TE	*drawing* (*D*)	1386	3854
	embroidery (*E*)	336	359
	jacquard (*J*)	1160	1276
	weaving (*W*)	239	307
TS	2^{nd} 19^{th}	1022	1160
	1^{st} 20^{th}	1611	2258
	2^{nd} 20^{th}	488	1201

Table 1. Overview of the class distributions for all properties. TS: *production timespan*. PL: *production place*. TE: *technique*. The classes for TS refer to half-centuries, e.g. 2^{nd} 19^{th} means *second half of the* 19^{th} *century*. The characters in parenthesis are abbreviations used in the tables below.

4.1.2 Test setup and evaluation strategy: We evaluate the method in two ways. First, we apply a quantitative evaluation based on a k-NN classification. For that purpose, after training we build a k-d tree from the descriptors of all training samples and query the descriptors of all test samples to the tree; we retrieve the $k = 5$ nearest neighbours and predict the properties of the queried samples by taking the majority vote of the properties of the nearest neighbours. We evaluate this classification by comparing the predicted labels to the reference labels. We report the overall accuracy (i.e., the percentage of correct predictions) and the F1-score for every class, i.e. the harmonic mean of precision and recall. Precision is defined as TP / (TP + FP) and recall as TP / (TP + FN), where TP is the number of samples of a class that was classified correctly (true positives), FP is the number of samples that was assigned to that class but belongs to another one in the reference (false positives), and FN is the number of samples assigned to another class than the one it belongs to in the reference (false negatives). As our definition of similarity is based on similarity of properties, in this way we can assess if for a test sample the nearest neighbours among the training samples in feature space really have the same properties, and the results can also be compared to those of Dorozynski et al. (2019).

Secondly, we compute the distances between the feature vectors of all test samples and compare them to the similarity values. Due to the normalization of the feature vectors, this cannot be done directly, but in the way of a regression analysis. Here, for lack of space we only report the correlation coefficient, which gives us the degree of linear dependency between the feature distance and the similarity according to our definition.

In all experiments we split the data into training, validation and test sets consisting 60%, 20% and 20% of the data, respectively. The evaluation is based on a five-fold cross validation. In each cross validation iteration, a different set of images is used for testing, so that over the course of all iterations each sample is used for testing once. We initialize the CNN as described in 3.3. All images are scaled to the required input size of 224×224 pixels before being propagated through the network. We train our proposed networks for 300 iterations with a batchsize of 100. Training is based on stochastic gradient descent using Adaptive Moments (Kingma & Ba, 2014) and the standard parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\hat{\epsilon} = 1 \cdot 10^{-8}$), except for the learning rate of $1 \cdot 10^{-4}$. During training, we also fine-tune the last layer of the ResNet-152 network. For regularization purposes, we apply early stopping based on the validation loss.

For the k-NN analysis, we carried out six different experiments to compare different network variants and different definitions of similarity. First, we investigate a single-property scenario. Thus, in experiment I, we trained three individual networks (one per property) using the Siamese architecture and the standard contrastive loss (eq. 5) based on the similarity according to eq. 1; in experiment II we also trained three such networks, but using the triplet architecture and the triplet loss (eq. 8) based on the definition of the margin from eq. 9. Both experiments are carried out using the comprehensive set of samples, for each property using all samples with a annotation for that property.

In experiments III and IV, we evaluate the performance of k-NN classification when considering all properties at once and using only complete samples. In both cases, only one CNN is trained. In experiment III, we use the Siamese architecture and the standard contrastive loss (eq. 5) based on the similarity according to eq. 2, while experiment IV is based on the triplet architecture and the triplet loss using the margin from eq. 9.

Finally, in experiments V and VI, we want to assess the impact of using incomplete samples. Thus, in experiment V, we train a CNN using the Siamese architecture with the variant of the contrastive loss according to eq. 6, using the definition of similary according to eq. 7; in experiment VI, we use the triplet architecture and the triplet loss based on the definition of the margin according to eq. 10. In both cases, the comprehensive set of samples is used for training. Nevertheless, the evaluation is based on the complete samples only so that the comparison to experiments III and IV is based on the same data.

4.2 Results and Discussion

4.2.1 K-NN Classification: The overall accuracies for all semantic properties of the six experiments are shown in tab. 2; the highest values are highlighted in bold font. These results show that considering multiple semantic properties at once (experiments III-VI) generally performs better than just considering one property at a time (experiments I, II); the improvement is in the order of 5%-10%. However, they also show that considering incomplete samples in the training procedure (experiments V, VI) leads to a drop in performance compared to using only complete samples (experiments III, IV). This drop is more prominent when comparing the results obtained when using the triplet loss (cf. experiment IV vs. VI): in this case, considering also incomplete samples leads to a drop of 4.9% in the mean overall accuracy. It is less prominent (1.1% in mean overall accuracy) when comparing the results obtained when using the Siamese architecture (cf. experiment III vs. V). The results thus show that when using multiple properties for defining similarity, the triplet loss performs better than the contrastive loss, but only when exclusively complete samples are used for training. This indicates that the training procedure based on the triplet loss is less robust to incomplete information than the procedure based on the Siamese architecture and the contrastive loss. A possible explanation is that for incomplete samples, the margin between a positive and a negative pair is larger when the contrastive loss is used than when the triplet loss is used. Consequently, descriptors of dissimilar images may be closer to each other in feature space, leading to a worse performance of the k-NN classification when the triplet loss is used.

Independently from the specific problems of the triplet loss with incomplete samples, there might be another reason why the results achieved when only considering complete samples are better than those achieved when including incomplete ones. In the training procedures of experiments V and VI, only about 38% of the samples are complete, which means that 62% are incompletely labelled and, thus, dominate the training process. As those incomplete samples do not reflect the interdependencies between the properties as complete samples arguably do, this learning procedure might lead to a loss of generality for the learned features, thus resulting in decreased quality measures.

Property	I	II	III	IV	V	VI
PL	85.8	83.8	94.1	**95.0**	93.2	93.2
TE	91.5	89.2	92.8	**94.1**	92.0	87.6
TS	84.6	85.2	91.8	**93.0**	90.1	86.6
Average	87.3	86.1	92.9	**94.0**	91.8	89.1

Table 2. Overall accuracies [%] of the six experiments for the properties *production place* (PL), *production technique* (TE) and *production timespan* (TS) and average values.

Tab. 3 presents the F1-scores for the individual properties and their classes. As we have already observed when analyzing the overall accuracies in tab. 2, the approaches considering multiple properties outperform the approaches considering only single properties. We again observe that the triplet loss outperforms the contrastive loss, but again only when no incomplete samples are used. Additionally, the F1-scores in tab. 3 indicate that the triplet loss performs better at classifying classes with very few training samples. For example, the F1-score for the *production place* class *Catalonia* (C) (2727 complete samples) is only 0.4% better for the triplet loss compared to the contrastive loss, while the improvement of 4.4% for the class *Spain (Rest)* (S) (only 394 complete samples) is considerably larger.

	Class	I	II	III	IV	V	VI
PL	C	88.6	86.9	96.8	**97.2**	96.0	96.2
	S	**81.3**	78.6	73.3	77.6	74.7	70.0
	Average	84.9	82.8	85.0	**87.4**	85.4	83.1
TE	D	96.0	93.8	95.7	**96.3**	95.5	90.7
	E	83.2	80.5	87.6	**91.8**	86.6	78.6
	J	83.3	79.7	**96.9**	93.5	91.7	87.3
	W	75.8	78.2	84.4	**86.2**	79.5	80.6
	Average	84.7	83.1	90.2	**92.0**	88.3	84.3
TS	2^{nd} 19^{th}	80.4	91.2	92.9	**93.5**	92.0	85.2
	1^{st} 20^{th}	90.8	81.2	94.8	**95.6**	93.7	91.3
	2^{nd} 20^{th}	76.7	77.8	79.3	**83.4**	72.8	81.6
	Average	82.6	83.4	89.0	**90.8**	86.2	82.7

Table 3. F1-scores [%] for the properties *production place* (PL), *production technique* (TE) and *production timespan* (TS). For the abbreviations of class names, cf. tab. 1.

We also compare our proposed approach to those achieved by multi-task learning by Dorozynski et al. (2019), which are based on the same dataset. We focus on the best variants in both papers, thus comparing the results achieved for completely labelled samples, i.e. experiment IV in this paper and variant MTL-C in (Dorozynski et al., 2019). The quality metrics for the experiments of both approaches are shown in tab. 4. The comparison shows that both approaches are on par with each other. While multi-task learning performs better in predicting the production place, our approach has a slightly better overall performance, reflected in an improvement of both the mean overall accuracy (0.5%) and the mean F1-score (1.1%). We can conclude that when using our approach to learn the semantic similarity between images of silk fabrics, the k-NN analysis is very likely to retrieve images having similar properties; if we predict the properties from the k-NN, the quality is as good as the one of a CNN trained to predict the semantic properties.

Figures 4 and 5 show two examples for k-NN analysis, i.e. two query images and their respective five nearest neighbours.

4.2.2 Analysis of correlation: This analysis is only carried out for our best learning strategy (experiment IV). For each iteration of five-fold cross correlation, we calculated the $I \cdot J$ Euclidean distances Δ_{ij} between all pairs (i, j) of $I = 1872$ training samples and $J = 624$ test samples. For every pair, we also calculated the similarity indicator Y_{ij} based on the annotations using eq. 2. Based on the resultant $I \cdot J$ tuples (Δ_{ij}, Y_{ij}) of distances and similarity indicators, we calculated the correlation coefficient $\rho_{\Delta Y}$ between the two variables. The average correlation coefficient from all cross-validation iterations was $\rho_{\Delta Y} = -0.90$. This high negative correlation shows there is a high degree of linear dependency between the two variables: the larger the difference between two feature vectors Δ_{ij}, the smaller their respective similarity Y_{ij} (and vice versa). We take this as an indicator that our proposed method can in fact be used to train a CNN to produce feature vectors such that their distances can be used to measure the similarity of images.

	Overall Accuracy		F1-score	
Property	MTL-C	IV	MTL-C	IV
PL	**95.4**	95.0	**87.7**	87.4
TE	92.9	**94.1**	90.4	**92.0**
TS	92.3	**93.0**	89.0	**90.8**
Average	93.5	**94.0**	89.0	**90.1**

Table 4. Comparison of (mean) overall accuracies [%] and (mean) F1-scores [%] between our approach and the approach presented in (Dorozynski et al., 2019).

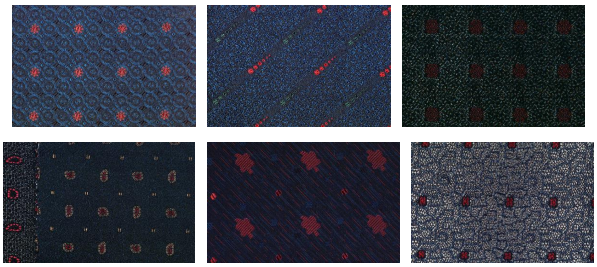

Figure 4. Exemplary result for k-NN analysis. The top left image is the query image, the other images are the five nearest neighbours. All properties of the query image were predicted correctly. Classes for all images: *Catalonia, 2^{nd} 19^{th}, jacquard.* © Centre de Documentació i Museu Tèxtil (IMATEX, 2018); photographer: Quico Ortega.

Figure 5. Exemplary result for k-NN analysis. The top left image is the query image, the other images are the five nearest neighbours. Here, the k-NN classification predicted only the production place (*Catalonia*) correctly. The majority vote from the nearest neighbours for the production timespan (1^{st} 20^{th}) and technique (*drawing*) lead to false predictions. © Centre de Documentació i Museu Tèxtil (IMATEX, 2018); photographer: Quico Ortega.

5. CONCLUSION

In this paper we have presented several approaches for CNN-based learning the similarity of images of silk fabrics based on semantic properties. The advantage of a definition of similarity based on semantic properties is that the training data can be generated automatically if a database with annotated images is available. We proposed two methods for training a CNN, based on a Siamese and on a triplet architecture, respectively. We compared different variants of the loss function designed to deal with different definitions of similarity based on semantic annotations. We evaluated our methods using a k-NN classification. Our experiments showed that considering multiple se-

mantic properties simultaneously is beneficial for learning the similarity between images, but only if completely labelled training samples are used. Our experiments also indicated that the triplet loss is less robust against incomplete labels than the contrastive loss. In general, k-NN classification based on our definition of similarity performed on par with a task-specific classifier (Dorozynski et al., 2019).

In future work we would like to include additional collections of images of silk fabrics. This would give us additional training samples and, possibly, a more balanced class distribution. However, introducing data from additional collections might pose a problem regarding the transferability between these collections. One way to solve this potential problem would be to use domain adaptation (Wang & Deng, 2018). Apart from introducing new data from additional collections, we would also like to consider additional semantic properties, such as *motif* or *production material*. As our results indicate that exploiting potential interdependencies between the properties is beneficial for learning the similarity, we assume that considering additional properties could still improve the process. We would also like to investigate a combination of multi-task classification and similarity learning, e.g. by combining our proposed network architecture and its (similarity-based) loss functions with the (classification) loss function of (Dorozynski et al., 2019). This approach could be used in the context of multi-task classification, where the network uses learned features for the prediction of multiple semantic variables. We think that guiding the network to producing dissimilar features for dissimilar inputs will improve the classification performance.

Another expansion could be to apply weights to the individual properties in the similarity functions. This weighting can be based on information provided by art historians in order to give more importance to certain properties, as the domain experts might consider them to be be of greater relevance for assessing the similarity of fabrics. In this context, we would also like to investigate whether those weights could instead be learned by the network if domain experts can provide us with labelled pairs of images of similar / dissimilar fabrics.

ACKNOWLEDGEMENTS

The research leading to these results is in the frame of the "SILKNOW. Silk heritage in the Knowledge Society: from punched cards to big data, deep learning and visual/tangible simulations" project, which has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No. 769504. We would also like to thank the Centre de Documentació i Museu Tèxtil, in particular Sílvia Saladrigas Cheng, for providing the data for this research and for giving us the permission to reproduce some of their images.

References

Babenko, A., Slesarev, A., Chigorin, A., Lempitsky, V., 2014. Neural codes for image retrieval. *European Conference on Computer Vision (ECCV)*, 584–599.

Bishop, Christopher M., 2006. *Pattern Recognition and Machine Learning*. 1^{st} edn, Springer, New York (NY), USA.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R., 1994. Signature verification using a "Siamese" time delay neural network. *Advances in Neural Information Processing Systems (NIPS)*, 737–744.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 248–255.

Dorozynski, M., Clermont, D., Rottensteiner, F., 2019. Multi-task deep learning with incomplete training samples for the image-based prediction of variables describing silk fabrics. *ISPRS Annals of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, IV-2/W6, 47–54.

Gordo, A., Almazán, J., Revaud, J., Larlus, D., 2016. Deep image retrieval: Learning global representations for image search. *European Conference on Computer Vision (ECCV)*, 241–257.

Hadsell, R., Chopra, S., Lecun, Y., 2006. Dimensionality reduction by learning an invariant mapping. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1735–1742.

Han, X., T., Leung, Jia, Y., Sukthankar, R., Berg, A. C., 2015. MatchNet: Unifying feature and metric learning for patch-based matching. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3279–3286.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *International Conference on Computer Vision (ICCV)*, 1026–1034.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Identity mappings in deep residual networks. *European Conference on Computer Vision (ECCV)*, 630–645.

Hentschel, Ch., Wiradarma, T. P., Sack, H., 2016. Fine tuning CNNS with scarce training data – Adapting imagenet to art epoch classification. *International Conference on Image Processing (ICIP)*, 3693–3697.

IMATEX, 2018. Centre de Documentació i Museu Tèxtil, CMDT's textilteca online. `http://imatex.cdmt.cat` (accessed 14 February 2019).

Jamil, N., Bakar, Z.A., Tengku Sembok, T.M., 2006. Image retrieval of songket motifs using simple shape descriptors. *Geometric Modeling and Imaging–New Trends (GMAI)*, 171–176.

Kingma, D. P., Ba, J., 2015. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations (ICLR 2015)*.

Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. ImageNet classification with deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems (NIPS)*, 1097–1105.

Long, M., Cao, Z., Wang, J., Yu, P. S., 2017. Learning multiple tasks with deep relationship networks. *Advances in Neural Information Processing Systems (NIPS)*, 1594–1603.

Nair, V., Hinton, G. E., 2010. Rectified linear units improve restricted boltzmann machines. *27th International Conference on Machine Learning (ICML)*, 807–814.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., B., Thirion, Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. 12 (Oct), 2825–2830.

Pirrone, A., Beurton Aimar, M., Journet, N., 2019. Papy-S-Net : A Siamese network to match papyrus fragments. *5th International Workshop on Historical Document Imaging and Processing*, 78–83.

Qi, Y., Song, Y.-Z., Zhang, H., Liu, J., 2016. Sketch-based image retrieval via Siamese convolutional neural network. *International Conference on Image Processing (ICIP)*, 2460–2464.

Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S., 2014. CNN features off-the-shelf: An astounding baseline for recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 806–813.

Schroff, F., Kalenichenko, D., Philbin, J., 2015. A unified embedding for face recognition and clustering. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 815–823.

Wang, H., Feng, L., Zhang, J., Liu, Y., 2016. Semantic discriminative metric learning for image similarity measurement. *IEEE Transactions on Multimedia*, 18 (8), 1579–1589.

Wang, J., Song, Y., T., Leung, Rosenberg, C., Wang, J., Philbin, J., Chen, B., Wu, Y., 2014. Learning fine-grained image similarity with deep ranking. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1386–1393.

Wang, M., Deng, W., 2018. Deep visual domain adaptation: A survey. *Neurocomputing*, 312, 135–153.

Zhao, F., Huang, Y., Wang, L., Tan, T., 2015. Deep semantic ranking based hashing for multi-label image retrieval. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1556–1564.

Zheng, L., Yang, Y., Tian, Q., 2017. SIFT meets CNN: A decade survey of instance retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40 (5), 1224–1244.