MULTI-TASK DEEP LEARNING WITH INCOMPLETE TRAINING SAMPLES FOR THE IMAGE-BASED PREDICTION OF VARIABLES DESCRIBING SILK FABRICS

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

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

Commission II, WG II/8

KEY WORDS: Multi-task learning, Convolutional Neural Networks, cultural heritage, silk fabrics, incomplete training samples

ABSTRACT:

This paper presents a method for the classification of images of silk fabrics with the aim to predict properties such as the place and time of origin and the production technique. The proposed method was developed in the context of the EU project SILKNOW (http://silknow.eu/). In the context of classification, we address the problem of limited as well as not fully labelled data and investigate the connection between the distinct variables. A pre-trained Convolutional Neural Network (CNN) is used for the feature extraction and a classification network realizing Multi-task learning (MTL) is trained based on these features. The training procedure is adapted to enable the consideration of images that do not have a label for all tasks. Additionally, MTL with fully labeled training data is investigated for the classification of silk fabrics. The impact of both MTL approaches is compared to single-task learning based on two different class structures. We achieve overall accuracies of 92-95% and average F1-scores of 88-90% in our best experiments.

1. INTRODUCTION

It is the main goal of the EU project SILKNOW (http://silknow.eu/) to produce a computational system that supports experts in cultural history to improve their understanding of European silk heritage. Information about historic artefacts such as silk fabrics is often collected in databases that are accessible via the internet, e.g. (IMATEX, 2018; MfAB, 2018). Pieces of information that are relevant for cultural heritage experts include the time or place of production of such an artefact, the material it is made of, or the technique that was used for its production. In order to support further scientific analyses by digital ressources, it is essential to have a standardized way of representing this information in a computer. However, many digital collections do not use such a standardized representation. In addition to a digital image showing an artefact, the collections provide metadata, containing the relevant information often as free text. The first step for standardization is the definition of an ontology that describes the pieces of information that may be relevant for the standardized description of an object in the computer and their mutual relations. The second step is the conversion of the available (unstandardized) information into that ontology. Given the fact that a digital collection may contain tens or even hundreds of thousands of records, a manual input of this information, e.g. by cultural historians reading the descriptive texts and extracting the relevant information for the ontology, is tedious, expensive and, consequently, often impossible. Thus, automated procedures have to be developed. Such methods can be based on automated processing of the available descriptive text. However, in many cases, certain pieces of information may not be contained in the textual descriptions, either because they were unknown at the time of writing or because they were considered negligible by the person formulating the text. The only other source of information that can be tapped to obtain the required information automatically are the digital images.

Although work for the classification of artistic pictures exists, e.g. (Blessing & Wen, 2010; Sharif Razavian et al., 2014), we are not aware of a method that does so for images of silk fabrics. Furthermore, these methods usually determine individual variables such as author, style or genre, but they do not take advantage of the inherent relationships between these variables. In this paper, we propose a method that predicts relevant variables for the description of silk fabrics from digital images of such fabrics. We use supervised learning based on deep Convolutional Neural Networks (CNN) (Krizhevsky et al., 2012) to automatically derive the information about the production time, the production place and the production method of silk fabrics. While being very successful in many image classification tasks, CNN are known to require a large amount of training samples, so that even the manual annotation of these training data may be prohibitive. In order to overcome this problem, the training samples are obtained automatically from a publicly available collection of textile images (IMATEX, 2018). Such a procedure may result in an inhomogeneous class distribution and incomplete training samples, i.e. samples for which a part of the class labels required for training are missing. As the amount of training data is limited, our method is built on top of a pre-trained ResNet-152 network (He et al., 2016) for the extraction of generic features from the image. On top of this feature extraction network a classification network is trained for the new specific classification tasks. Based on the assumption that the time and place of production as well as the production technique are not independent, these three tasks are trained together in the course of multi-task learning, e.g. (Leiva-Murillo et al., 2013). Consequently, our CNN learns a joint feature representation for all three tasks while applying task-specific classifiers to predict the class labels for the individual tasks. The end-to-end training procedure is designed such that incompletely labelled samples can also contribute. In this way, the training set can be expanded, and it is expected that the individual tasks and particularly the underrepresented classes in each task can benefit from both, the larger amount of data as well as the con-

*Corresponding author

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-2-W6-47-2019

sideration of potential dependencies between the tasks.

The scientific contribution of this paper can be described as follows. To the best of our knowledge, this is the first method designed to predict multiple variables related to silk fabrics from images. For that purpose, we propose a new CNN architecture that is capable of multitask learning with the goal of learning a joint representation of the images for predicting different variables. We present an end-to-end learning procedure that can deal with incomplete training samples; this is achieved by a specific definition of the training loss function. The developed method is trained and evaluated using samples that were scraped automatically from an available web collection of images of silk fabrics. In the evaluation we show the benefits of multitask learning as opposed to learning individual classifiers for each task.

The remainder of this paper is structured as follows. Section 2 outlines related work on the classification of images of cultural artefacts and methods for multitask learning. The developed methodology for the classification of images of silk fabrics is described in section 3. Section 4 describes the dataset that was used for training and evaluation. The experimental evaluation of our methodology based on these data is presented in section 5. In Section 6 we draw a conclusion and give recommendations for future work.

2. RELATED WORK

Image-based classification of artistic pictures is not an entirely new problem in Photogrammetry and Computer Vision. In Blessing and Wen (2010), different types of hand-crafted features and a multi-class Support Vector Machine (SVM) were used to predict the painter; using images from seven different painters, an overall accuracy of 85.1% was achieved. In Saleh & Elgammal (2015), the performance of different methods of feature extraction and metric learning were compared. The goal was to produce optimal feature vectors for the classification of style, genre and artist by an SVM. The individual tasks (i.e., the prediction of style, genre and artist, respectively) were dealt with independently from each other and using different subsets of training images. The reported quality indices are somewhat lower than those of Blessing and Wen (2010), but Saleh and Elgammal (2016) differentiated more classes for each of the three variables. In Arona & Elgammal (2012), a comparison of different hand-crafted features for predicting the artistic style of a picture is presented. The best results (overall accuracy of 65.4%) were achieved when using Classeme features (Torresani et al., 2010) in combination with a SVM when differentiating between seven classes.

Since the development of AlexNet (Krizhevsky et al., 2012), CNN (LeCun et al., 1989) have revolutionized the field of image classification. Sharif Razavian et al. (2014) demonstrated that features from a pre-trained CNN enable a sufficient representation of images for new recognition tasks, especially in the case of limited training data; this approach was also applied to the classification of artistic paintings. Based on the features of a pre-trained AlexNet, a new classification layer was trained to distinguish 22 art epochs of the Wikiart data set, achieving an accuracy of 55.9% (Hentschel et al., 2016). Other approaches to predict the painter of an artistic picture based on a pre-trained CNN are Sur & Blaine (2017) and Tan et al. (2016), achieving accuracies of 82.5% and 76.1%, respectively. However, these methods only predict one specific variable and their application domain is not related to silk fabrics.

In contrast to the classification of paintings, predicting properties of fabrics is a much less investigated field. A CNN-based classifier was trained to predict different patterns of knitted fabrics in Xiao et al. (2018), achieving an overall accuracy of 98.4% among eight categories of structures. There are also contributions to detect fabric defects based on images (Gao et al., 2018). This leads to a binary classification problem. In Gao et al. (2018), a defect detection accuracy of 96.5% has been achieved. The only one work dealing with the classification of a more abstract fabric property is our own previous work (Dorozynski et al., 2019), where we used a network on top of a pre-trained ResNet-152 to predict the time of production of silk fabrics. This method will serve as a baseline for the multi-task learning framework presented in this paper.

All papers cited so far deal with the prediction of variables of works of art or fabrics based on images, but none of the contributions investigated the joint learning and prediction of several variables. That the joint training of related tasks can be beneficial in comparison to a separate training of the individual tasks was already stated in Caruana (1993), who introduced multitask learning for artificial neural networks and decision trees. In the last couple of years, several approaches for multi-task learning in combination with CNNs were developed for different recognition tasks. For instance, in Li et al. (2014) a shared feature extraction network consisting of convolutional and pooling layers followed by task-specific networks out of dense layers are trained for human pose estimation. A similar architecture is proposed in Long et al. (2017), where all convolutional layers as well as the first fully connected layer are shared for all tasks and the subsequent task-specific dense layers can interact via tensor normal priors. A further option to share information between the tasks is given by cross-stitching units learning a linear combination of the activation maps introduced at different stages between the task-specific CNNs of the tasks (Misra et al., 2016). This architecture leads to a relatively large number of parameters because there are task-specific (though interacting) representations at intermediate layers of the network.

There is only limited work on multi-task learning with partly labelled data. García-Laencina et al. (2008) address the classification with partly incomplete feature vectors in the context of multi-task learning, but all labels are considered to be available. Other multi-task learning approaches that deal with (partly) unlabelled data tackle the problem based on semi-supervised learning (Luo et al., 2013), predicting semi-labels for the missing information, which, however, may be wrong. To the best of our knowledge, there is no contribution that focuses only on the available labels even if some of them are missing. There is also no work on multi-task learning for the classification of fabrics based on CNN.

3. METHODOLOGY

Our approach to predict multiple variables describing silk fabrics is based on CNN architectures that take an image of a fabric and deliver one class label per task. In this context, a task corresponds to one distinct classification problem with the goal of predicting one variable related to the textiles. Exemplarily, the variables *production timespan*, *production place* and *technique* are investigated in this paper.

The proposed network architectures require RGB images of the size 224 x 224 pixels as input. They rely on a ResNet-152 (He et al., 2016) as a generic extractor for high-level features, but additionally contain fully connected layers and one or more softmax layers for the actual classification. We choose this network as a feature extractor based on our insights from Dorozynski et al. (2019), where ResNet-152 showed the best performance compared to other feature extractors using similar data. We use the ResNet-152 parameters pre-trained on the ImageNet data set (Deng et al., 2009) and keep them fixed during the training procedure, while we determine the parameters of the other layers to adapt the classifier to the new domain of silk fabrics. Thus, a sufficiently good representation of the images can be obtained (Sharif Razavian et al., 2014), which is beneficial for small data sets (as the one used in this paper) that do not provide enough training data to train the whole ResNet-152. In subsection 3.1, the proposed architectures are described in more detail, while the training procedure is presented in section 3.2.

3.1 Network Architectures

In this work, two network architectures for the prediction of the required variables from images are applied. The first one enables the prediction of the class labels for a single variable and is illustrated schematically in figure 1. We refer to it as our network for single-task learning (STL). As a basis, the ResNet architecture consisting of 152 convolutional and pooling layers from He et al. (2016) is adopted, providing a feature vector with 2048 entries. To map the features to the K task-specific classes, three fully connected (fc) layers are utilized. The first and second fc layers, consisting of 1000 and 100 nodes, respectively, use a Rectified Linear Unit (ReLU) (Nair & Hinton, 2010) as their activation function. The third layer, consisting of as many nodes K as there are classes to be learned, is a softmax layer and delivers the class scores. Figure 1 shows the feature vectors as gray bars and also gives the dimensions of these vectors. The arrows symbolize the network layers that lead to these representations. The number of classes K depends on the variable to be predicted and the available training samples (because classes without samples are not considered). In order to predict multiple variables, multiple instances of this architecture need to be trained, and all of these instances have to be applied to an image independently from each other. In our experiments, this network architecture will mainly serve as a baseline for a comparison to the second network architecture based on MTL.

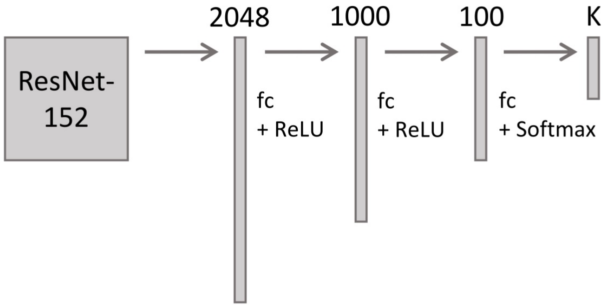

Figure 1. Network architecture for single-task learning. fc: fully connected layers. ReLU: rectified linear unit. K is the number of classes.

We conjecture that the STL architecture results in a considerable overhead in terms of the parameters that have to be determined. We argue that the variables we are interested in are closely related and that there is an intrinsic relationship between the three classification tasks. Consequently, our main goal is to build and test an architecture for multi-task learning (MTL) and classification. Our MTL architecture is illustrated in figure 2. Similarly to the STL architecture, ResNet-152 is chosen as a generic feature extractor, converting the input image into a 2048-dimensional feature vector. On top of ResNet-152, there is one fully connected layer with 1500 nodes that is shared among all tasks and, thus, delivers a task-independent representation of an image to be classified. This layer is followed by two further task-specific fully connected layers, the last one being a softmax layer delivering the classification scores. All dense layers except the last one have a ReLU activation. The number of output nodes is equal to the number of classes that will be predicted for the individual tasks. These numbers are denoted by K_{ts}, K_{pp} and K_{te} for the variables *production timespan*, *production place* and *technique*, respectively. Preliminary experiments showed that neither architectures consisting of more dense layers (either shared among the tasks or task-specific) nor architectures with more nodes per dense layer or both could deliver better results than the one shown in figure 2.

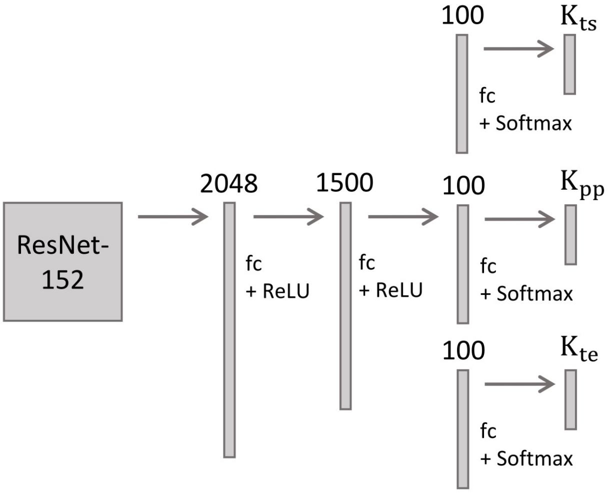

Figure 2. Network architecture for multi-task learning. fc: fully connected layers. ReLU: rectified linear unit. K_{ts}, K_{pp} and K_{te} are the numbers of classes for *production timespan*, *production place*, *technique*, respectively.

The main advantage of the MTL architecture is the fact that by sharing the larger fc layer, it requires fewer parameters than the STL architecture, which we consider to be beneficial when the number of training samples is restricted. Not considering the final softmax layer, the STL architecture has 2049 × 1000 + 1001 × 100 ≈ 2,149k parameters per variable, thus altogether 6,447k parameters (weights + biases) need to be determined when training three independent STL nets for predicting three different variables. On the other hand, the number of parameters for the MTL network that delivers the same three variables is 2049 × 1500 + 3 × 1501 × 100 ≈ 3,524k (again excluding the softmax layers). The number of parameters for the softmax layers is identical in both cases (100 times the corresponding number of classes; note that in case of STL, the value K will be substituted by K_{ts}, K_{pp} and K_{te}, respectively, in the three instances required to predict the three variables *production timespan*, *production place* and *technique*). By requiring the CNN to learn a joint representation, MTL requires only about 55% of the number of parameters of STL for the fc layers before the softmax layers when all three tasks are to be predicted. In our experiments (section 5) we investigate the effects of this reduction in parameters on the classification results.

3.2 Training

To train a neural network, an objective function that evaluates the quality of the class predictions of the training samples has to be minimized. Such a loss function compares the predicted class label and the ground truth class label to estimate the error in predicting the correct class based on the current network parameters. The believe y_k of a network having the parameter values $\mathbf{w}$ that the sample x_n belongs to the class k results from the softmax activations

$$y_k\left(x_n, \mathbf{w}\right) = \frac{exp\left(w_k^T \cdot \Phi\right)}{\sum_{j=1}^{K} exp\left(w_j^T \cdot \Phi\right)} \tag{1}$$

in the case of a multi-class classification problem with K classes (Bishop, 2006). In equation 1, w_k is a vector of the weights of the k^{th} node of the last layer of the network, while Φ denotes the activations of the preceding network layer. Denoting the weights of all hidden layers of the network by $\overline{\mathbf{w}}$, i.e., $\mathbf{w} = \overline{\mathbf{w}} \cup w_1 \cup \ldots \cup w_K$, we can consider Φ to be a function of the sample x_n and $\overline{\mathbf{w}}$: $\Phi = \Phi\left(x_n, \overline{\mathbf{w}}\right)$. Assuming that N training samples are given and that each one shall be assigned to one of K disjoint classes, a possible loss function for training a CNN is given by the softmax cross-entropy function $E\left(\mathbf{w}\right)$ (Bishop, 2006):

$$E\left(\mathbf{w}\right) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \cdot ln\left(y_k\left(x_n, \mathbf{w}\right)\right). \tag{2}$$

In equation 2, t_{nk} is an indicator variable that is equal to one if the n^{th} sample belongs to the k^{th} class and zero otherwise, and ln denotes the natural logarithm. Thus, the loss is small for a parameter selection $\mathbf{w}$ that assigns each sample x_n to its correct class k (for which $t_{nk} = 1$). By minimizing equation 2, the parameters $\mathbf{w}$ of a CNN such as the one depicted in figure 1 can be determined. We use a variant of stochastic minibatch gradient descent (Krizhevsky et al., 2012) using backpropagation for an efficient computation of the gradients (Bishop, 2006). More details on optimization are given in section 5. Note that in all experiments, we only determine the parameters of the fc layers, while for the ResNet-152 we use pre-trained parameters that are frozen in the optimization process.

Having to predict several variables per samples, the related tasks can be learned simultaneously in the framework of MTL. The implicit assumption of MTL is that there is some intrinsic relation between these variables. For our MTL architecture, we compare two training strategies that differ by the restrictions they impose on the training samples. The first strategy is based on the assumption that for every training sample x_n we know the correct class label for *all* of the M tasks (i.e., for all variables to be predicted). We refer to such samples as *complete* training samples, and in the first strategy we assume that we only use such complete samples. Under these assumptions, the softmax cross-entropy loss for determining the parameters of the MTL network shown in figure 2 can be extended to

$$E\left(\mathbf{w}\right) = -\sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{k=1}^{K_m} t_{nmk} \cdot ln\left(y_k\left(x_n, \mathbf{w}\right)\right). \tag{3}$$

In equation 3, $m \in \{1, \ldots, M\}$ is the index of the task and K_m is the corresponding number of classes. Again, t_{nmk} is an indicator variable that is equal to one for the correct class k of the n^{th} sample for variable m and zero otherwise. That is, for task m, t_{nmk} is one only for one of the K_m different classes differentiated in that task, so that the variables t_{nmk} have to fulfill the following constraint:

$$\sum_{k=1}^{K_m} t_{nmk} = 1 \quad \forall m \in M. \tag{4}$$

Note that for M=1, equation 3 is equivalent to equation 2. If we assume all training samples in the process of minimizing equation 3 to be complete samples, every task will contribute equally to the determination of the parameters of the joint (first) fc layer of the network in figure 2. Thus, we expect our network to learn a good common representation for all tasks at that layer. Furthermore, in every training iteration, all parameters of all fc and softmax layers will be updated.

However, requiring all training samples to be complete is a rather strong restriction. Hand-labelling training samples is a tedious and time-consuming task. As we will describe in section 4, we generate training samples automatically by harvesting online collections that are available in the WWW. Under these circumstances, one will get access to much larger training datasets, but at the cost that many samples will be incomplete. That is, for many samples one might only know the correct labels for a subset of the tasks, e.g. due to a poor or incomplete annotation of the online collections. The second training strategy investigated in this paper relaxes the assumptions of the first stragegy by also allowing incomplete samples to be used in the training process. Denoting the subset of tasks for which a class label is known for sample x_n by M_n, the second training strategy is based on minimizing

$$E\left(\mathbf{w}\right) = -\sum_{m \in M_n}\left(\sum_{n=1}^{N}\sum_{k=1}^{K_m} t_{nmk} \cdot ln\left(y_k\left(x_n, \mathbf{w}\right)\right)\right), \tag{5}$$

which is equivalent to equation 3 in the case of complete samples; in the case of incomplete samples, only samples with known labels contribute to the loss. If the label for a task is unknown, the weights and biases of the related task-specific fc and softmax layers are not affected by a sample. On the one hand, this may lead to a lower quality of the joint representation, in particular if there is a set of variables for which the training label is unknown for a very large number of samples. On the other hand, this procedure allows the inclusion of a larger number of samples and, thus, the differentiation between more classes per task (because some labels may not occur at all if one discards all incomplete samples). In our experiments we compare the two training strategies for MTL.

4. GENERATION OF TRAINING SAMPLES

The presented methodology is applied to data that were obtained from the online collection of the Centre de Documentació i Museu Tèxtil in Terrassa (Spain) (IMATEX, 2018). Among several museum collections we checked this collection contains the largest amount of images that are useful for our application and it also had the most consistent annotations. The collection contains a total of 30916 RGB images of fabrics, clothes and accessories as well as some paper designs for textiles. Figures 3 and 4 illustrate examples for images of fabrics and design papers. As we are only interested in properties of mere textiles, all images of clothes and accessories were discarded. All images are scaled such that the larger dimension (could be width or height) is equal to 400 pixels. Consequently, the other dimension may be smaller and varies between 25 and 400 pixels. For each image, descriptive texts concerning various characteristics of the depicted objects are given in the online collection. In order make the data available for our purposes, we implemented a web crawler in Java; its source code is available publicly at `https://github.com/SILKNOW/crawler/` . The web crawler performs specific search requests on the specified museum web site and serializes the web site responses in the Java Script Object Notation (JSON) data format.

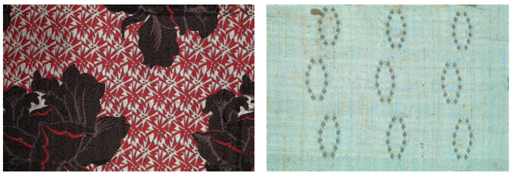

Figure 3. Examples for images of fabrics we used for our work. Classes: *2nd half 19th century, Catalonia, Jacquard.* ©Quico Ortega (IMATEX, 2018)

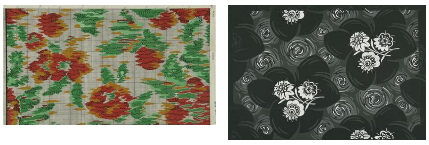

Figure 4. Examples for images of paper drawings we used for our work. Classes: *Unkown production time and place, Drawing.* ©Quico Ortega (IMATEX, 2018)

These JSON files contain the meta information about all samples in corresponding pairs of `label` and `value` fields. This information was further processed to make it useful for our application. First, we had to map the names of properties (given in Catalán language in the `label` fields of the JSON files) to the variables we wanted to predict. As pointed out earlier, we focussed on the three variables *production timespan* (referred to as *CRONOLOGIA* in the JSON files), the *production place* (*ORIGEN*) and the (production) *technique* (*TECNICA*) of fabrics. Furthermore, for each of these variables, the corresponding descriptions (given in the `value` fields of the JSON files, again in Catalán language) had to be mapped to class labels. After defining the class structure, we generated one conversion table per variable. For each variable, this conversion table contained all descriptive strings that occur in the JSON files. The mapping table also contains the class labels, which are assigned to the corresponding descriptions. For instance, the strings *Itália* - and *Itália -/ Biella* - for *ORIGEN* were both mapped to the class *Italy* for variable *production place*. The class definitions and the conversion tables were generated such that all classes were mutually exclusive and there was a sufficiently large number of samples for each class (at least 200).

Selecting the images that have a valid description and thus a valid class label for at least for one of the three targeted variables leads to a dataset of 10383 images. All class definitions as well as the number of samples that are available for the individual classes can be seen in table 1. In total, 44% of the images have information about *production timespan*, the *production place* is known for 81% of the depicted fabrics and the manufacturing *technique* is recorded for 66% of the data. These statistics do already show the relevance of the proposed methodology: for incomplete samples the missing information could be predicted by a CNN.

Enabling the prediction of all classes with a roughly equal quality, an equal number of samples per class is desired for training. Furthermore, a sufficient number of representative training samples is required. Whereas the distribution of images among the classes is relatively homogeneous for the task *production timespan*, the other two tasks have a quite unbalanced number of images per class. In addition to this challenge, a further problem arises when the dataset is restricted to contain complete samples only. For the task *production timespan* this is only the case for 3137 images or less than 50% of all samples; the distributions of the samples among the classes are also shown in table 1. Note that regarding the complete samples, no samples are available for some of the classes. Thus, when only complete samples are considered, these classes cannot be considered.

	Class name	Complete samples	Incomplete samples
TS	2nd half 19th c.	1022	1160
	1st half 20th c.	1611	2258
	2nd half 20th c.	488	1201
PL	Spain	394	2671
	Catalonia	2727	4322
	Italy	-	551
	Non-western	-	880
TE	drawing	1386	3854
	embroidery	336	359
	jacquard	1160	1276
	weaving	239	307
	damask	-	579
	velvet	-	500

Table 1. Overview of the class distributions for all tasks. TS: *production timespan*. PL: *production place*. TE: *technique*.

5. EXPERIMENTS

In this section, the proposed network architectures and the according learning procedures of section 3 are applied to the data introduced in section 4. The setup of the experiments and the evaluation strategy are described in section 5.1. Afterwards, the results are presented and discussed in section 5.2.

5.1 Test setup and evaluation strategy

The procedure for all experiments involves the splitting of the respective data into training, validation and test sets, where 60% of the data contribute to the training and 20% of the data are used for validation and for testing, respectively. The feature extraction part of the ResNet-152 is initialized with the pre-trained weights based on the ImageNet dataset (Deng et al., 2009) and the parameters of the classification networks are initialized randomly by drawing them from zero-mean normal distributions. Before an image is propagated through the network, it is scaled to the required input size of 224 × 224 pixels.

In the course of a five-fold cross validation, the respective classification network is trained until a saturation of the validation accuracies can be observed. 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 $1 \cdot 10^{-4}$. Finetuning of hyper-parameters for all experiments would have led to a considerable amount of computations and is beyond the scope of this paper. We believe that the results achieved using these settings are already quite good; further improvements by exploring better hyper-parameter settings are left for future work.

We report on two series of experiments. In the first set of experiments, the performance of STL, MTL with complete samples and MTL with incomplete samples will be determined to enable a comparison of these three approaches. As the class structure has to be the same for all three experiments to allow for a fair comparison, only the classes that occur in complete samples (cf. table 1) can be considered. All incomplete samples that contribute to other classes are discarded. Thus, the classes *Italy* and *Non-western* are not considered for the variable *production timespan* and the classes *damask* and *velvet* are not considered for the variable *technique* in the first set of experiments. The second set of experiments makes use of all classes listed in table 1. In this way, the MTL approach with incomplete samples can be compared to the STL approach on the basis of a more complex class structure. The MTL method with complete samples could not be applied in this case because there were no samples for some classes.

The first experiment of the first series, referred to as *STL*, addresses the training of the variables *production timespan*, *production place* and *technique* independently from each other. All images of fabrics that provide class labels for one of the classes to be discerned are considered in the course of STL, applying the architecture depicted in figure 1. This means that incomplete samples could also be applied in this case, because learning a single task needs samples that have a label for the task to be learned and incompleteness of labels for all other tasks does not matter. Thus, the number of samples per class is the one in the column for incomplete samples in table 1, but samples for classes for which no complete samples are available were discarded.

The second experiment in the first series investigates the impact of multi-task learning with completely labelled training data (*MTL-C*) on the classification performance, where exclusively samples providing a label for all of the three tasks are taken into account. Thus, the dataset is reduced to all samples according to the column of complete samples in table 1. Consequently, the architecture depicted in figure 2 is trained by minimizing equation 3.

The last experiment in the first series realizes multi-task learning, too, but samples having an incomplete labelling (*MTL-I*) are additionally considered, which is achieved by minimizing the loss according to equation 5 in the training procedure. Thus, the samples enumerated in the column for incomplete samples in table 1 form the basis for this experiment. In comparison to STL, all variables are relevant in the same training procedure, whereas STL considers only one variable per training of an STL network.

The second set of experiments compares the *STL* and the *MTL-I* approaches. The difference to the first set of experiments is the modified data basis. In contrast to the first set of experiments, the classes *Italy* and *Non-western* are additionally considered for the variable *production timespan* and the classes *damask* and *velvet* are additionally considered for the variable *technique*.

The evaluation of the experiments is based on five-fold cross validation. In each test run, a different group of the images is used for testing, so that in an entire experiment, each sample appears in the test set once. We compare the predictions to the reference and report the overall accuracy (the average percentage of correctly classified samples over all test runs). Because of the unbalanced distributions of the samples among the classes, we also report the F1-score

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{6}$$

as a class-specific quality measure. In equation 6, the recall is the percentage of the samples of a class according to the reference that is also assigned to that class by the CNN, while precision is the percentage of the samples assigned to a certain class that actually corresponds to that class in the reference. The F1 score is one possible measure that combines these two measures.

5.2 Results and Discussion

5.2.1 Test series 1: The overall accuracies for all variables of all experiments of the first set of experiments are shown in table 2. In general, it can be observed that introducing MTL leads to improvements for all variables compared to the STL approach, but only in the case of fully labelled samples. Training the MTL architecture on the basis of incompletely labelled samples decreases the overall accuracies for the task *production timespan* by 0.4% and by 1.2% for the task *production place*. Only the accuracy of the *technique* task can be improved slightly by 0.4%. In contrast, improvements for all three tasks can be obtained using samples with class labels for all tasks. The variable *production timespan* has an increase of 6.8% in overall accuracy, while the improvement for *production place* and *technique* is 8.2% and 2.0%, respectively.

The F1-scores are presented in the tables 3 to 5, where each table contains the F1-scores for one task. As already observed by considering the overall accuracies, the MTL approach with completely labelled samples outperforms the two other approaches according to the F1-scores, too. Except for the class *Spain*, all classes of all tasks show improved results compared to the separate learning of the individual variables (the exception certainly occurs because of the enormous reduction of samples for that class). On average, the F1-scores of *production timespan* is increased by 4.7%, those of *production place* and *technique* by 1.2% and 5.1%, respectively. The MTL approach with incompletely labelled training samples achieves on average lower or similar F1-scores in comparison to the STL approach.

A possible reason for the superiority of MTL with complete labels over MTL with incomplete labels may be the training

Task	STL	MTL-C	MTL-I
production timespan	85.8	**92.3**	85.4
production place	87.2	**95.4**	86.0
technique	90.9	**92.9**	91.3
Average	88.0	**93.5**	87.6

Table 2. Overall accuracies [%] of the first series of experiments.

Class name	STL	MTL-C	MTL-I
2nd half 19th c.	81.5	**91.5**	81.4
1st half 20th c.	91.3	**95.2**	91.3
2nd half 20th c.	80.0	**80.4**	78.6
Average	84.3	**89.0**	83.8

Table 3. F1-Scores [%] for the task *production timespan* (first series of experiments).

Class name	STL	MTL-C	MTL-I
Spain	**83.2**	78.3	81.3
Catalonia	89.7	**97.2**	88.8
Average	86.5	**87.7**	85.1

Table 4. F1-Scores [%] for the task *production place* (first series of experiments).

Class name	STL	MTL-C	MTL-I
drawing	95.1	**96.5**	95.2
embroidery	83.5	**90.3**	82.9
jacquard	83.6	**92.5**	84.1
weaving	78.9	**82.4**	79.2
Average	85.3	**90.4**	85.3

Table 5. F1-Scores [%] for the task *technique* (first series of experiments).

Task	STL	MTL-I
production timespan	**85.8**	85.1
production place	76.0	**78.9**
technique	87.4	**89.0**
Average	83.1	**84.3**

Table 6. Overall accuracies [%] of the second series of experiments.

procedure for the joint dense layer. In the case of partly unknown labels, only 38% of the samples have a class assignment for all three tasks. This means that 62% of the samples lead to a weight update of the joint layer in training even though they do not reflect all interdependencies of the tasks. Thus, the representation learned by the joint fc layer is most probably dominated by the variables having missing labels. This leads to a loss of generality for the representation, resulting in decreased quality measures.

The improvements in the case of MTL with complete samples are probably caused by the gain in generality obtained in the training of the joint layer. Due to the contribution of all variables to the joint layer's weight updates, the representation becomes more generic for all tasks.

5.2.2 Test series 2:

The overall accuracies of the second set of experiments can be seen in table 6. In contrast to the accuracies of the first series, MTL with incomplete samples leads to an improvement for the variable *production place* of 2.9% and of 1.6% for *technique*. Only the variable *production timespan* has a decrease of 0.7% in overall accuracy.

Class name	STL	MTL-I
2nd half 19th c.	**81.5**	81.0
1st half 20th c.	**91.3**	90.6
2nd half 20th c.	**80.0**	78.7
Average	**84.3**	83.4

Table 7. F1-Scores [%] for the task *production timespan* (second series of experiments).

As the overall accuracies already indicated, the F1-scores that can be seen in the tables 7 to 9 are also decreased due to the introduction of MTL for the *production timespan*, whereas an increase of the F1-score can be observed for nearly all classes of the variables *production place* and *technique*. The only exceptions are the classes *Non-western* and *embroidery*. A possible reason for these exceptions could be the relative low number of samples (especially for *embroidery*) and the decrease in the case of *Non-western* may occur because of the potential diversity within that class. All other *production places* denote one country whereas *Non-western* contains influences from Egypt, China, Japan, India and Iran.

Class name	STL	MTL-I
Spain	76.0	**76.8**
Catalonia	80.8	**84.2**
Italy	55.9	**57.5**
Non-western	**70.9**	69.1
Average	70.9	**71.9**

Table 8. F1-Scores [%] for the task *production place* (second series of experiments).

Class name	STL	MTL-I
drawing	94.1	**94.9**
embroidery	**83.6**	83.1
jacquard	80.6	**83.0**
weaving	76.5	**76.6**
damask	80.7	**83.4**
velvet	72.9	**75.0**
Average	81.4	**82.7**

Table 9. F1-Scores [%] for the task *technique* (second series of experiments).

One general observation is that the F1-scores of the STL approach as well as MTL with incomplete samples are lower than the achieved scores in test series 1. This may be caused by the more complex class structure that requires the estimation of more complex decision boundaries in the feature space by means of the trained fc layers. Be that as it may, Test series 2 shows that the utilization of MTL with incompletely labelled training data can improve the classification performance when considering all available classes. Even though MTL with incomplete samples lowers the quality measures in the experiments of Test series 1 (having a reduced number of classes), the approach is able to improve two of the three targeted tasks in the case of a more complex class structure.

6. CONCLUSION

Two multi-task learning approaches for the prediction of the *production timespan*, the *production place* as well as the production *technique* of images of fabrics were introduced in this paper. One approach needs a class label for all three tasks per training sample and the other approach needs a class assignment for at least one of the tasks. In two test series, based on different class structures, the two approaches were compared to a single task learning approach. The first test series with a reduced number of classes showed that the best classification results can be achieved with multi-task learning when class labels for all tasks are available. Thus, overall accuracies of 92.3% to 92.9% and average F1-scores of 87.7% to 90.4% were obtained. The second test series demonstrated that multi-task learning with an incomplete labelling still can improve some of the tasks in the case of a more complex class structure.

In future work, further collections of images of fabrics should be included to obtain a more balanced distribution of the samples among the classes on the one hand and to maintain all

https://doi.org/10.5194/isprs-annals-IV-2-W6-47-2019

classes in MTL requiring a class label for all tasks. Possible improvements of the MTL approach allowing an incomplete labelling might be achieved by using the samples with incomplete labels only for a fine-tuning of the task-specific layer and to exclude them from the training of the joint layer. Another option could be a weighting of the samples; those providing a label for all tasks get a higher weight than those with missing labels. Furthermore, another realization of combining the tasks could be investigated for both MTL approaches, so that the network may learn inherent dependencies between the tasks. In that regard additional variables might be included for the classification, e.g. the author of a fabric/drawing or the subject depicted. Because of the inherent dependencies between variables, including new variables may even improve the classification results for the variables already considered.

Beyond that, a modification of the feature extraction would be of interest. The last layers of the pre-trained ResNet-152 could be fine-tuned and thus be adapted for the classification of silk fabrics or an own architecture could be investigated. Both modifications of the feature extraction require an expansion of the dataset to enable the training of additional parameters. This expansion could be achieved either by integrating further textile collections or by artificially enlarging the dataset by means of data augmentation.

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 IMATEX for providing the annotated images for this research.

References

Arona, R. S., Elgammal, A. M., 2012. Towards automated classification of fine-art painting style: A comparative study. *International Conference on Pattern Recognition*, 3541–3544.

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

Blessing, A., Wen, K., 2010. Using machine learning for identification of art paintings. Technical report, Stanford University, USA.

Caruana, R. A., 1993. Multitask learning: A knowledge-based source of inductive bias. *International Conference on Machine Learning*, 41–48.

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*, 248–255.

Dorozynski, M., Wittich, D., Rottensteiner, F., 2019. Deep Learning zur Analyse von Bildern von Seidenstoffen für Anwendungen im Kontext der Bewahrung des kulturellen Erbes. *Publikationen der DGPF, Band 28, 387-399.*

Gao, C., Zhou, J., Wong, W. K., Gao, T., 2018. Woven fabric defect detection based on convolutional neural network for binary classification. *International Conference on Artificial Intelligence on Textile and Apparel*, 307–313.

García-Laencina, P. J, Figueiras-Vidal, AR, Sancho-Gómez, JL, 2008. Incomplete pattern classification using a multi-task approach. *e-Proceedings of the 12th world multi-conference on systemics, cybernetics and informatics*, 1–6.

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

Hentschel, Ch., Wiradarma, T. P., Sack, H., 2016. Fine tuning cnns with scarce training dataadapting imagenet to art epoch classification. *IEEE International Conference on Image Processing*, 3693–3697.

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

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 25 (NIPS'12)*, 1, 1097–1105.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., Jackel, L. D., 1989. Backpropagation applied to handwritten ZIP code recognition. *Neural Computation*, 1, 541–551.

Leiva-Murillo, J. M., Gómez-Chova, L., Camps-Valls, G., 2013. Multitask remote sensing classification. *IEEE Transactions on Geoscience and Remote Sensing*, 51(1), 151–161.

Li, S., Liu, Z.-Q., Chan, A. B, 2014. Heterogeneous multi-task learning for human pose estimation with deep convolutional neural network. *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 482–489.

Long, M., Cao, Z., Wang, J., Yu, P. S., 2017. Learning multiple tasks with deep relationship networks. *Advances in Neural Information Processing Systems 30 (NIPS'17)*, 1594–1603.

Luo, Yong, Tao, Dacheng, Geng, Bo, Xu, Chao, Maybank, Stephen J, 2013. Manifold regularized multitask learning for semi-supervised multilabel image classification. *IEEE Transactions on Image Processing*, 22(2), 523–536.

MfAB, 2018. Museum of Fine Arts Boston. `https://www.mfa.org/collections` (accessed 14 February 2018).

Misra, I., Shrivastava, A., Gupta, A., Hebert, M., 2016. Cross-stitch networks for multi-task learning. *IEEE Conference on Computer Vision and Pattern Recognition*, 3994–4003.

Nair, V., Hinton, G. E., 2010. Rectified linear units improve restricted boltzmann machines. *Proceedings of the 27th international conference on machine learning (ICML-10)*, 807–814.

Saleh, B., Elgammal, A., 2016. Large-scale classification of fine-art paintings: Learning the right metric on the right feature. *International Journal for Digital Art History*, 2, 70–93.

Sharif Razavian, A., 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 Workshops*, 806–813.

Sur, D., Blaine, E., 2017. Cross-depiction transfer learning for art classification. Technical Report CS 231A and CS 231N, Stanford University, USA.

Tan, W. R., Chan, C. S., Aguirre, H. E, Tanaka, K., 2016. Ceci n'est pas une pipe: A deep convolutional network for fine-art paintings classification. *IEEE International Conference on Image Processing*, 3703–3707.

Torresani, L., Szummer, M., Fitzgibbon, A., 2010. Efficient object category recognition using classemes. *European Conference on Computer Vision*, 1, 776–789.

Xiao, Z., Liu, X., Wu, J., Geng, L., Sun, Y., Zhang, F., Tong, J., 2018. Knitted fabric structure recognition based on deep learning. *Journal of the Textile Institute*, 109(9), 1–7.

https://doi.org/10.5194/isprs-annals-IV-2-W6-47-2019