SILKNOW

D6.7. SYSTEM DOCUMENTATION

Deliverable number and name:	D6.7 System Documentation
Due date:	30/06/2021
Work Package:	WP6
Deliverable leader:	EURECOM
Authors:	Raphaël Troncy (EURECOM), Thibault Ehrhart (EURECOM), Pasquale Lisena (EURECOM), Thomas Schleider (EURECOM), Mar Gaitán (UVEG), Arabella León (UVEG), Luis Rei (JSI), Dunja Mladenic (JSI), Mareike Dorozynski (LUH), Franz Rottensteiner (LUH), Javier Sevilla (UVEG), Pablo Casanova (UVEG)
Reviewers:	Cristina Portalés (UVEG)
Approved by:	Jorge Sebastián (UVEG)
Dissemination level:	PU
Version:	V1.0

European Commission

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769504

Document revision history			
Version	Date	Contributor	Comments
0.1	04/05/2021	Raphaël Troncy	Table of contents
0.2	26/05/2021	Mar Gaitán	Contribute Chapter 6+7
0.3	01/06/2021	Franz Rottensteiner	Contribute Chapter 4
0.4	24/06/2021	Luis Rei, Franz Rottensteiner, Dunja Mladenic	Contribute Chapter 3
0.5	05/07/2021	Pasquale Lisena, Thibault Ehrhart, Thomas Schleider	Finalization of the videos, contribute to all Chapters
0.6	10/07/2021	Javier Sevilla, Pablo Casanova	Contribute Chapter 5
0.7	20/07/2021	Raphaël Troncy, Dunja Mladenic, Luis Rei	Overall revision
0.8	25/07/2021	Cristina Portalés	Revision
1.0	26/07/2021	Raphaël Troncy	Add the final conclusion

List of acronyms	
API	Application Programming Interface
ARTIC	Art Institute of Chicago
CNN	Convolutional Neural Network
GELU	Gaussian Error Linear Unit
ICT	Information and Communication Technologies
MET	Metropolitan Museum of Art
MFA	Boston Museum of Fine Arts
MTL	Multi-Task Learning
RDF	Resource Description Framework
RISD	Rhode Island School of Design
SPARQL	SPARQL Protocol and RDF Query Language
STL	Single Task Learning
VAM	Victoria and Albert Museum
WP	Workpackage

Table of contents

This deliverable provides the final documentation of the SILKNOW system in the form of a textual report presenting the latest integration efforts and evaluations as well as short video tutorials exemplifying how the ADASilk exploratory search engine can be used to realize different user scenarios tailored to the various targeted audiences of SILKNOW.

1. INTRODUCTION

This deliverable includes both a textual report that describes the latest improvement and evaluation of the final integrated system, and multimedia live examples that come as short videos published on the SILKNOW YouTube video channel[1] together with reproducible scenarios. We describe the final documentation of the SILKNOW integrated system that takes the form of the ADASilk exploratory search engine publicly available at https://ada.silknow.org/. More precisely, we detail:

- The integration of the image and text analysis predictions: these modules aim to predict some missing metadata from the original museum records using machine learning techniques. The deliverable describes 4 different methods with their performance (Section 3).
- The integration of the image-based retrieval that enables an end-user to search museum records by providing an image query as input. The module also enables to obtain recommended content because they are either visually or semantically similar to the currently viewed object (Section 4).
- The integration of the latest versions of the Virtual Loom and Spatio-Temporal Maps (Section 5).

An essential component of the integrated system is the SILKNOW Multilingual Thesaurus. This thesaurus has been continuously improved. It is made available for public browsing at https://skosmos.silknow.org/thesaurus/en/ relying on the SKOSMOS software that we have ourselves further developed and optimized. It is also possible to integrate this resource in a third-party software using an API we provide for this purpose. We describe those two scenarios in this deliverable (Section 6).

Finally, we have revisited the pilot scenarios for the various targeted audiences that we identified at the beginning of the project, and we show, via short videos, how one can realize each of those pilot scenarios using tools developed in SILKNOW (Section 7).

[1] https://www.youtube.com/channel/UCTJJT6jhtJwMRprw808Tw9w/

2. RELATIONSHIP WITH OTHER DELIVERABLES

ADASilk, that stands for "Advanced Data Analysis for Silk Heritage" and is named after Ada Lovelace, is a responsive web application that has been largely described in D6.4 “Design and implementation of the multilingual web-based thesaurus” and in D6.5 “Integrated System”. It enables to explore the SILKNOW knowledge graph that has been first described in D6.3 “Ontology Web Server”.

ADASilk makes use of the SILKNOW thesaurus described in D3.1 “Historical Silk Multilingual Thesaurus”. It also integrates the Virtual Loom described in the D5.4 “Design and implementation of the Virtual Loom” as well as the Spatio-Temporal Maps described in D5.5 “Visualization and Deployable Components”. The integration of these two components in ADASilk was already detailed in D6.5. The evaluation of the developed functionalities and stress testings of the Virtual Loom and the Spatio Temporal Maps components are described in D5.7 “Test Report of Virtual Loom and Deployable Components”.

An image classification module was developed in the context of WP4. Its general principles were originally described in deliverable D4.4 “Design and implementation of a deep learning based image classification system”. The final version of this module is described in D4.6 “Test report of image processing and deep learning module”. The integration of this module into ADASilk is described in this deliverable.

The Text Classification tool was developed in the context of WP3, first reported on in Deliverable D3.3 “Design and implementation of text analytic module”, and later re-evaluated in Deliverable D3.4 “Test report of text analytic module”. We describe in this deliverable the integration of the predictions from this module that are then visualized in ADASilk.

The design and development of the ADASilk web application follows the requirements described in D2.3 “Definition of the Graphical User Interface (GUI)” and enables to support the use cases defined in D2.4 “Pilot scenario definition”.

The system has been evaluated in terms of usability in D7.3 “Usability Testing”. A thorough evaluation of the complete integrated system performance and functionalities is detailed in D6.6 “Functional Evaluation Report”.

3. INTEGRATION OF THE IMAGE AND TEXT ANALYSIS PREDICTION

3.1. Overview

3.1.1. Image Classification

It is the goal of the image classification module to predict information that is missing in the SILKNOW knowledge graph using images as input. It was decided to restrict the module to predict the five semantic variables, namely: *Production Timespan*, *Production Place*, *Production Technique*, *Production Material* and *Subject Depicted Type* (also referred to as *Depiction*), because the other variables were not expected to be related to the visual appearance of a silk fabric. Furthermore, it was decided to restrict the classification to images showing plain fabrics only, i.e., images showing other objects such as furniture or dolls are not considered to be a valid input.

Image classification is based on a Convolutional Neural Network (CNN). First, a pre-trained backbone network is used as a generic feature extraction network; its output is processed by a series of fully connected network layers before a final classification layer delivers a probabilistic class score per variable and class. There are two basic variants of the CNN. The variant based on single-task learning (STL) consists of one branch only and predicts class scores for a single semantic variable only. Thus, in order to predict five variables, five different instances of the CNN have to be trained. On the other hand, the variant based on multi-task learning (MTL) has five classification branches following a common feature extractor, so that only one CNN has to be trained to predict all five variables simultaneously. Furthermore, for both network variants, there are two options for defining a classification task. The standard variant is a multi-class classification with mutually exclusive classes and based on the softmax function for computing the class scores; in this case, only one class label (the one achieving the largest class score) is selected to be the classification result. In addition, the software also can be configured to allow multiple binary classifications for a set of variables that can be configured by the user. In this case, for every possible class of a variable, a binary classification is carried out, predicting whether the image is consistent with that class or not. For instance, in this way the software can be configured to predict that multiple materials were used to produce a specific type of fabric. In all cases, the CNN is trained using training samples exported from the SILKNOW Knowledge Graph. The software is highly configurable in terms of the network structure, various types of hyper-parameters and the loss function to be optimized in training. More details can be found in deliverables D4.4 and D4.6.

In principle, the image classification module should predict class labels for the five considered variables that are consistent with the SILKNOW Thesaurus. However, while being consistent with the Thesaurus, the annotations available in the SILKNOW Knowledge Graph, which are to be used to train the classifier in a supervised way, were found to be too diverse in the sense that there were too many different annotations per variable, so that the number of training samples per class would have been too small. Consequently, two less detailed class structures were defined by domain experts from UVEG as described in deliverable D4.5 and the annotations according to these class structures were integrated into the SILKNOW knowledge graph. The first class structure, integrated into the knowledge graph by an attribute *group* of every record, is slightly more fine-grained than the other one, integrated by an attribute *category* for every record. Each possible label according to the *group* class structure merges several terms of the Thesaurus so that there is a meaningful number of samples that can be attributed to that class; every class label of the *category* structure is the union of a set of *group* classes. Whereas the simplified class structures are consistent with the terms of the Thesaurus in so far as every class can be interpreted as a union of different Thesaurus terms, the classes themselves do not directly correspond to Thesaurus terms in most cases. More details can be found in D4.5 and D4.6.

All records in the SILKNOW knowledge graph that are linked with at least one image and that contain annotations for at least one of the five variables to be predicted can be used to train a CNN classification model. Once such a model is trained, it can be used to predict a class label for an image; in deliverable D4.6 and also in this deliverable, the class label is one of those defined according to the *group* class structure, i.e. the more fine-grained one of the two aggregated class structures. As far as the integration into ADASilk is concerned, the image classification module is not a module that is accessible by the user to predict class labels for arbitrary images he or she uploads. Instead, it is used internally to make the semantic information contained in the SILKNOW knowledge graph more complete. That is, it was used to predict the class labels for records corresponding to plain silk fabrics with associated images but without annotations for some of the five semantic variables mentioned above; the information whether or not an object is a plain silk fabric is contained in the knowledge graph in the variable *category_group*, which has the unique label *fabrics* for such objects. The predictions of the image classification module are integrated into the knowledge graph in the *group* attributes of the corresponding records. This attribute is also accessible to a user in the ADASilk platform, along with the corresponding class score and the information about the way in which the annotation was generated, so that the user knows that the information was produced by an automated process which is not perfect, so that the predictions have to be interpreted with caution. The actual integration of the predictions into the SILKNOW knowledge graph and ADASilk is presented in Section 3.2.

Deliverable D4.6 reports on an extensive set of experiments that were carried out to validate the image classification module. A part of the experiments was related to hyper-parameter tuning, which resulted in the recommendation of a specific variant of the CNN for the integration: The best results were achieved by a multi-task CNN for standard multi-class classification that was trained by minimizing the focal loss; the pre-trained model was also a part of the deliverable D4.6. However, the experiments described in D4.6 were based on the state of the SILKNOW Knowledge Graph on **19/02/2021**. Since then, additional collections have been integrated into the knowledge graph, so that the number of available records with annotations has grown considerably. As a consequence, there is also a larger number of classes with a sufficient number of training samples. Furthermore, problems with the variable *category_group*, used to differentiate records corresponding to silk fabrics from other records, could be solved by EURECOM. Originally, there was an inconsistency in the contents of this variable, so that in D4.6, only objects for which the value of this variable was *fabrics* could be used for training. Due to the modifications, the interpretation of this variable has become consistent: if there are multiple labels for that variable and *fabrics* is one of them, the sample corresponds to a plain fabric and can be used for training.

As a result of these modifications, at the time of preparation of the integration of the image classification results into the knowledge graph, a new dataset was exported from the SILKNOW Knowledge Graph on **21/05/2021**, and it was decided to train and test a new CNN model based on the new dataset. After converting the dataset into the format required by the image classification software and eliminating annotations for infrequent classes (classes for which there are fewer than 150 samples), the new dataset that to be used for training and testing consists of about **48,850 samples**, which is almost twice as many as were available for being used in the experiments reported in deliverable D4.6. The samples are from 12 collections (ARTIC, CER, Garín, IMATEX, Joconde, MET, MFA, Mobilier, RISD, Smithsonian, VAM, Versailles); in D4.6, ARTIC was not considered in the generation of samples. Compared to deliverable D4.6, there are two additional classes for *Production technique* (*resist_dyeing* and *tabby*), one additional class for *Depiction* (*geometrical_shape*), one additional class for *Production timespan* (*15th_c.*) and five additional classes for *Production Place*. More details about the procedure of generating training and test samples can be found in deliverable D4.6; a more detailed description of the dataset is given in Section 3.3.1.

Due to the expansion of both, the class structure and the training dataset, the experiments for hyper-parameter tuning reported in D4.6 were repeated. With one exception, they followed the same protocol as those described in sections 4.3.1 and 4.3.2 of D4.6. The exception is that the standard multi-class classification approach was used in all cases. That is, it was decided not to consider the scenario of multiple binary classification. The latter was implemented in order to be able to use samples having multiple annotations for some classes in training. However, the number of samples that could be included due to this adaptation was very small (about 150) and, thus, the benefit that can be gained from this scenario is very limited. All the other experiments were repeated using five-fold cross validation in order to get a solid statistical evidence. The results followed a similar pattern as those reported in D4.6; they are omitted here in order not to overload this deliverable. Basically, the recommendations made in deliverable D4.6 could be confirmed, the only exception being the hyper-parameter `num_finetune_layers` indicating the number of layers of the ResNet backbone that are to be fine-tuned in the training procedure: in D4.6, it was found advantageous to fine-tune the last five convolutional layers; the results of the experiments carried out for hyper-parameter tuning of the CNN to be used for the integration of the classification module into ADASilk indicated a slight advantage for a variant in which only the last layer of the backbone was fine-tuned. Thus, **the CNN recommended for integration is a multi-task network with five multi-class classification heads that was trained using the focal loss**. The final values for the hyper-parameters that were investigated are shown in Table 1, which also highlights the parameter for which another value was used in the experiments reported in deliverable D4.6; details about the exact definition of these hyper-parameters can also be found in D4.6.

Parameter Name	***Parameter Setting (D4.6)***	***Parameter Setting (D6.7)***
num_finetune_layers	5	1
num_joint_fc_layer	1	
Num_nodes_joint_fc	1500	
learning_rate	0.001	
relevant_variables	['technique', 'place', 'depiction', 'material', 'timespan']	
nameOfLossFunction	focal	
random_crop	[1.0, 1.0]	

`random_rotation90`	`False`
`gaussian_noise`	`0.0`
`flip_left_right`	`False`
`flip_up_down`	`False`
`weight_decay`	`0.001`
`multi_label_variables`	`None`

Table 1. Overview of the control parameters of the functions `crossvalidation_parameter` and `classify_images_parameter`. These parameters are the hyper-parameters used for training the CNN. *Parameter Name*: name of the parameter in the API; *Parameter Setting (D4.6)*: The parameter setting that was found to be optimal in deliverable D4.6, where they are given in Table 40; *Parameter Setting (D6.7)*: The optimal parameter values found by hyper-parameter tuning using the most recent dataset. They are used to train the model for the integration into ADASilk.

The way in which the results of the recommended CNN were integrated into ADASilk is described in Section 3.2. A detailed evaluation of the integrated CNN model based on five-fold cross validation is presented in Section 3.3. Before that, Section 3.1.3 describes how the results of image classification were used in a joint multi-modal classification step involving images, text and available annotations.

3.1.2. Text Analysis

The Text Analysis Module consists of a Convolutional Neural Network (CNN) over cross-lingual pre-trained word embeddings which, given text descriptions of silk objects, predicts the value of a semantic variable. These variables are *Production Timespan*, *Production Place*, *Production Technique*, *Production Material*. They were chosen on the basis of their interest to domain experts, availability of supervised training data, and to match the Image Classification Module which itself had similar goals. Deliverable D3.3 explored a series of alternative model architectures and hyper-parameters which led to the choice of task-specific TextCNN-based architecture as the basis of the text classification tool. Deliverable 3.4 re-evaluated the performance of the algorithm on new data and the updated class structure.

Our CNN-based text classification algorithm consists of an architecture that takes text descriptions found in museum records of silk fabrics and maps the sequence of words into a sequence of cross-lingual pre-trained word embeddings (vectors). The elements of this list are then concatenated together. Different convolutional blocks, with different convolutional kernel sizes (2, 3, 4), consisting of 100 filters each, are then applied to this sequence, generating what can be intuitively thought of as “n-gram” “features” where “n” equals the kernel size. The output of the filters is passed through a GELU non-linearity and applied a max-pooling operation which can be intuitively thought of “extracting” or “selecting” the best features for each block. These are concatenated together into a single vector, regularized by a dropout layer and sent to a softmax classification layer which produces the prediction of the classifier for the given input. Table 2 shows an example of the input data and the predictions for the variable “Production Technique”.

Object ID	Text Description	Real Value	Predicted Value
http://data.silknow.org/object/cee3686e-b61c-3a54-9627-68910a6f1e63	Dibujo columpio. Color crema y verde. Dos telas cosidas Urdimbre: Trama: pasadas Rapport: 33 cm ancho y 58 cm alto (incompleto)	damask	damask
http://data.silknow.org/object/bcfc9006-b205-395b-a40d-41c1ef34e8f0	Pair of triangular panels, green silk satin, China, Qing dynasty, ca. 1800 Pair of green silk satin panels embroidered with phoenix motif.	embroidery	embroidery

Table 2. Text descriptions and corresponding real and predicted value for “Production Technique”.

3.1.3. Categorical Classification with Gradient Tree Boosting

In Deliverable D3.4 “Test report of text analytic module”, we introduced a categorical classifier using the Gradient Tree Boosting algorithm implemented in the XGBoost library. This classifier uses the categorical values of other semantic variables and the museum source to predict the missing property. In case the property does not have a value, it replaces it with a “NULL” value. Table 3 shows an example input and the predicted values for “Production Material”.

Object ID	Inputs				Production Material	
	Museum	Production Place	Production Technique	Production Date	Real	Predicted
http://data.silknow.org/object/e6ed3faa-0a9f-33dc-8164-1857f9178734	CER	ES	Velvet	eighteenth century (dates CE)	animal_fibre	animal_fibre
http://data.silknow.org/object/bfac9605-0a49-31f9-9c41-96c9c8e3ed75	GARIN	ES	Velvet	NULL	vegetal_fibre	vegetal_fibre

Table 3. Categorical Classifier inputs and corresponding real and predicted value for “Production Material”.

3.2. Integration with ADASilk

We use classes and properties of the Provenance Data Model (Prov-DM), more specifically the PROV ontology (PROV-O) [6], an OWL2 ontology allowing the mapping of the PROV-DM to RDF. It is a W3C recommendation and allow to express the fundamental elements of the predictions based on images and the ones from text descriptions.

The prediction itself can be represented using the `prov:activity` class, respectively with their own type, image or text. This activity takes an image (identified by a permalink from the SiLKNOW media server) or a plain text as input, represented with the property `prov:used`. With the property `prov:atTime`, the exact date of the prediction can be expressed. The property `prov:wasAssociatedWith` connects this activity class to another PROV-O class, `prov:SoftwareAgent`. This class contains solely the property `P70_document` and contains a string describing the image or text analysis experiment that was run for the prediction.

The actual prediction is represented with an `rdf:statement` class, connected to `prov:activity` through a `prov:wasGeneratedBy` property. It expresses the confidence score of the prediction with the property L18 (has confidence score) from our own SILKNOW Ontology, expresses the predicted value in form of a URI with `rdf:object`, the type of predicted property through `rdf:predicate` in form of the appropriate CIDOC-CRM property type. The property `rdf:subject` is lastly connecting this statement to the production class (E12) of the object in the Knowledge Graph (KG).

Lastly, every prediction is inserted at the appropriate place inside the existing KG, too. So, if a material was predicted, it gets inserted with the CIDOC-CRM property `P126_employed` at the production class of the object.

As the text and image analysis models for the predictions are solely trained on group labels, they can only predict group labels. The predicted value gets therefore mapped to a more concrete concept, as the value for a property cannot be directly a group, but only the member of such a group. To give an example: if "Damask" as a group gets predicted, it will be in the form of a so-called facet link:

http://data.silknow.org/vocabulary/facet/damask

During the conversion such a link gets mapped to a specific member that is as close as possible to this group name, so in this case:

http://data.silknow.org/vocabulary/168

which is the URI for the SILKNOW Thesaurus concept "Damask".

All predictions are serially converted using the described data model into the Turtle file format and then uploaded and stored inside their own graph identified by `http://data.silknow.org/predictions`. Therefore, predictions can easily be removed, hidden or displayed separately on ADASilk.

An example of such a prediction in Turtle is:

```
<http://data.silknow.org/statement/4de637f8-ad68-5260-b64f-1c68d19e92b8> a
rdf:Statement ;
```

```
  silk:L18 "0.9173"^^xsd:float ;
  rdf:object <http://data.silknow.org/vocabulary/168> ;
  rdf:predicate ecrm:P32_used_general_technique ;
  rdf:subject <http://data.silknow.org/production/d25ce423-1f03-3544-b859-
ebfff7929cc5> ;
  prov:wasGeneratedBy <http://data.silknow.org/activity/4de637f8-ad68-5260-
b64f-1c68d19e92b8> .

<http://data.silknow.org/activity/4de637f8-ad68-5260-b64f-1c68d19e92b8> a
prov:Activity ;
  prov:atTime "2021-02-10T00:00:00"^^xsd:dateTime ;
  prov:used <https://silknow.org/silknow/media/mobilier/81636_0.jpg> ;
  prov:wasAssociatedWith <http://data.silknow.org/actor/luh-image-
analysis/1> .

<http://data.silknow.org/actor/luh-image-analysis/1> a prov:SoftwareAgent ;
  ecrm:P70_documents "Predictions made using a CNN-based image
classification software. Given an input image, the model, available at
https://zenodo.org/record/4742418, is able to predict values for five
properties, namely production 'timespan', 'production place', 'technique',
'material' and 'depiction'. It has been trained based on a February 2021
snapshot of the Knowledge Graph. The multi-task learning (MTL) variant is
being used in a multi-class classification (mutually exclusive classes)
fashion based on the softmax function for computing the class scores." .

<http://data.silknow.org/production/d25ce423-1f03-3544-b859-ebfff7929cc5>
ecrm:P32_used_general_technique <http://data.silknow.org/vocabulary/168> .
```

This example can also be represented by the graph depicted in Figure 1.

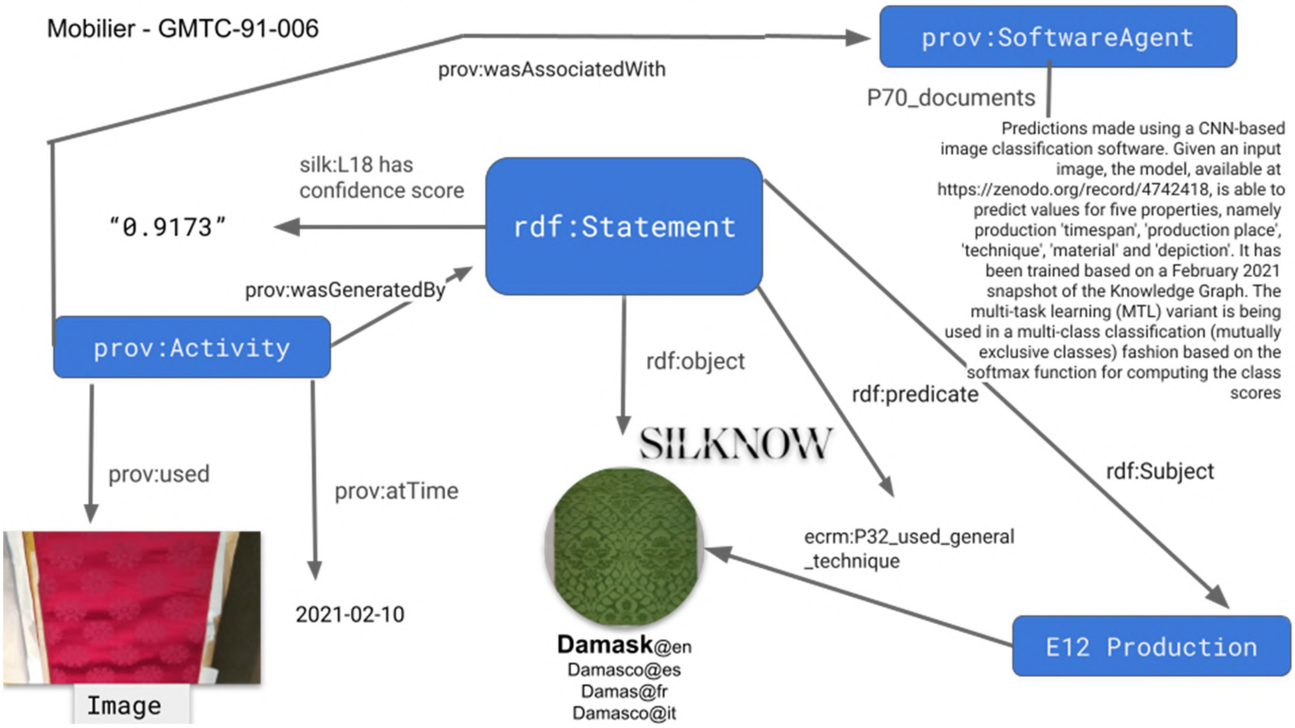


Figure 1. Graph showing the prediction of the production technique (damask) with a high confidence score (0.9173) using the textual analysis software.

In total, 98,379 predictions exist for 19,248 distinct objects and are uploaded into the SILKNOW Knowledge Graph. Figures 2 and 3 show the distribution of the prediction per properties for respectively the image analysis module and the text analysis module.

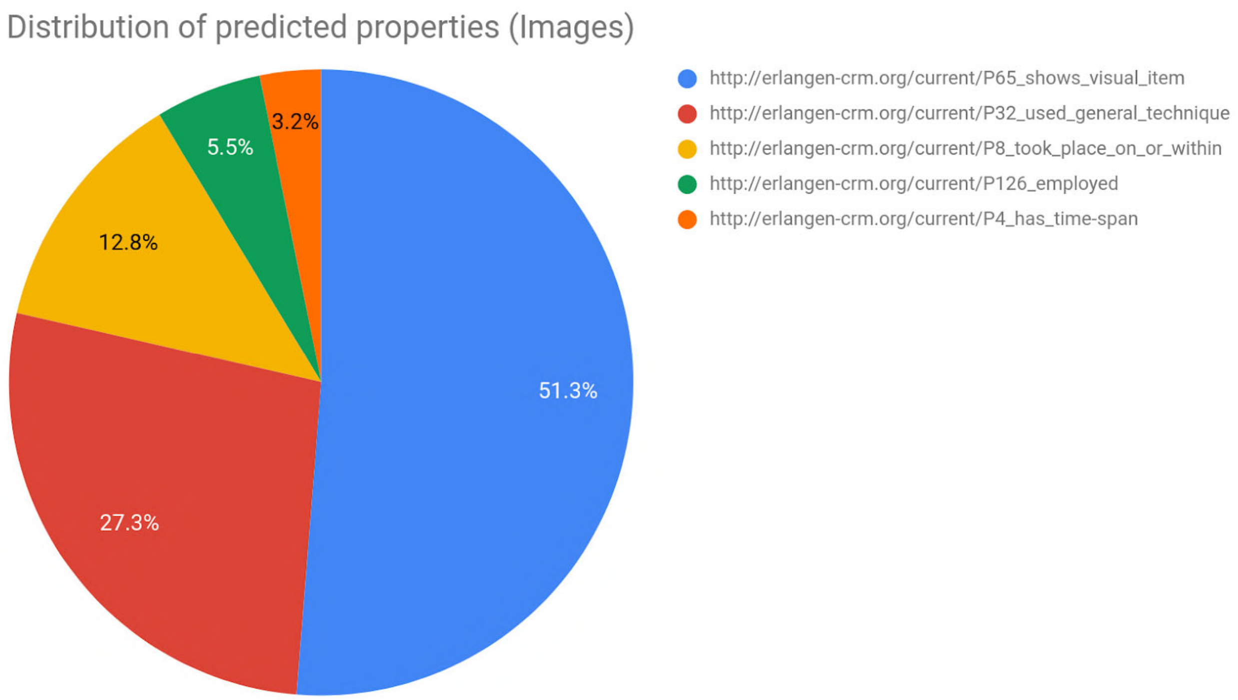


Figure 2. Distribution of the predictions per properties for the image analysis module.

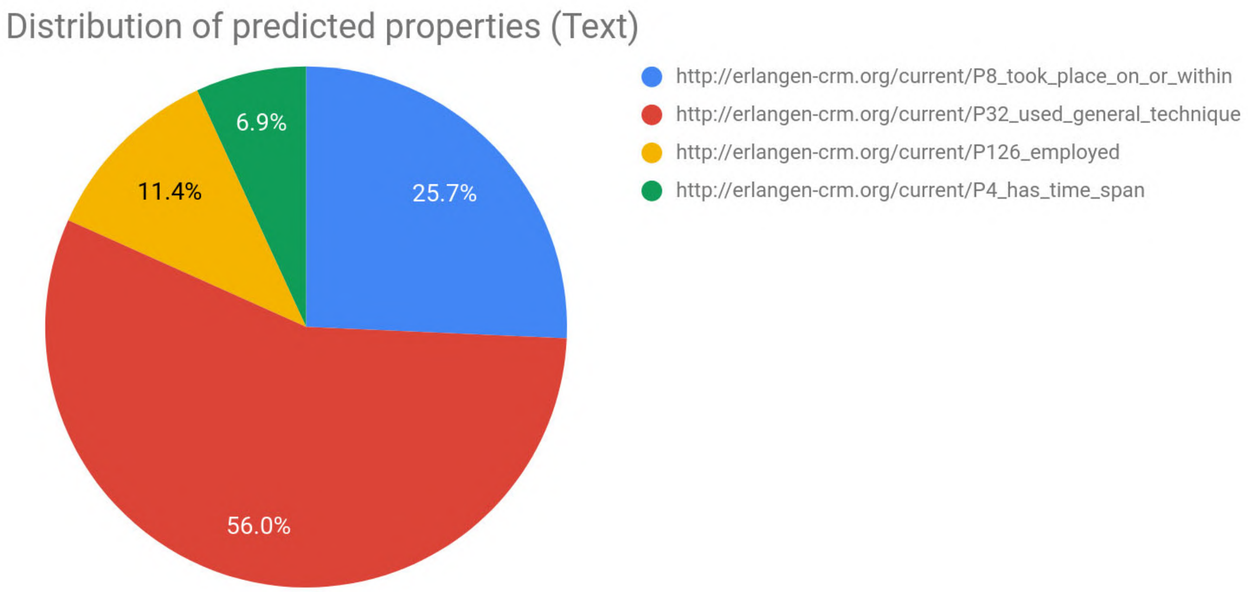


Figure 3. Distribution of the predictions per properties for the text analysis module.

Count	Percentage	Type (of Property)
16788	84%	Visual Item / Depiction
8943	45%	Technique
1805	1%	Material
1038	1%	Time
4177	21%	Place
19876	**100%**	**TOTAL OBJECTS**

Table 4. Table for proportion of properties for predictions based on images.

Count	Percentage	Type (of Property)
8782	79%	Technique
1788	16%	Material
1076	10%	Time
4026	36%	Place
11156	**100%**	**TOTAL OBJECTS**

Table 5. Table for proportion of properties for predictions based on text descriptions.

In order to display the predictions on ADASilk, the SPARQL query has been updated to take into account the new properties, by using subqueries to fetch the information about the predictions. It uses the data from the statements which contain the targeted property (`rdf:predicate`), the predicted value (`rdf:object`), and the prediction score (`http://data.silknow.org/ontology/L18`).

```
{
  SELECT DISTINCT ?production ?material ?materialLabel
?predictedMaterialScore
  WHERE {
    GRAPH <http://data.silknow.org/predictions> {
      ?statement rdf:subject ?production .
      ?statement rdf:predicate ecrm:P126_employed .
      ?statement rdf:object ?material .
      ?statement <http://data.silknow.org/ontology/L18>
?predictedMaterialScore .
    }
    ?material skos:prefLabel ?materialLabel .
  }
}
```

The infobox which contains metadata properties has also been modified in order to display predicted values with a different style. It relies on the availability of a “score” sub-property to consider the value as predicted.

```
material: {
  '@id': '?material',
  label: '?materialLabel',
  score: '?predictedMaterialScore',
}
```

Predictions are shown in blue to be differentiated with non-predicted values. The score is also being displayed as a percentage next to the value. Finally, a tooltip is shown to explain where the value comes from. An example is depicted in Figure 4.

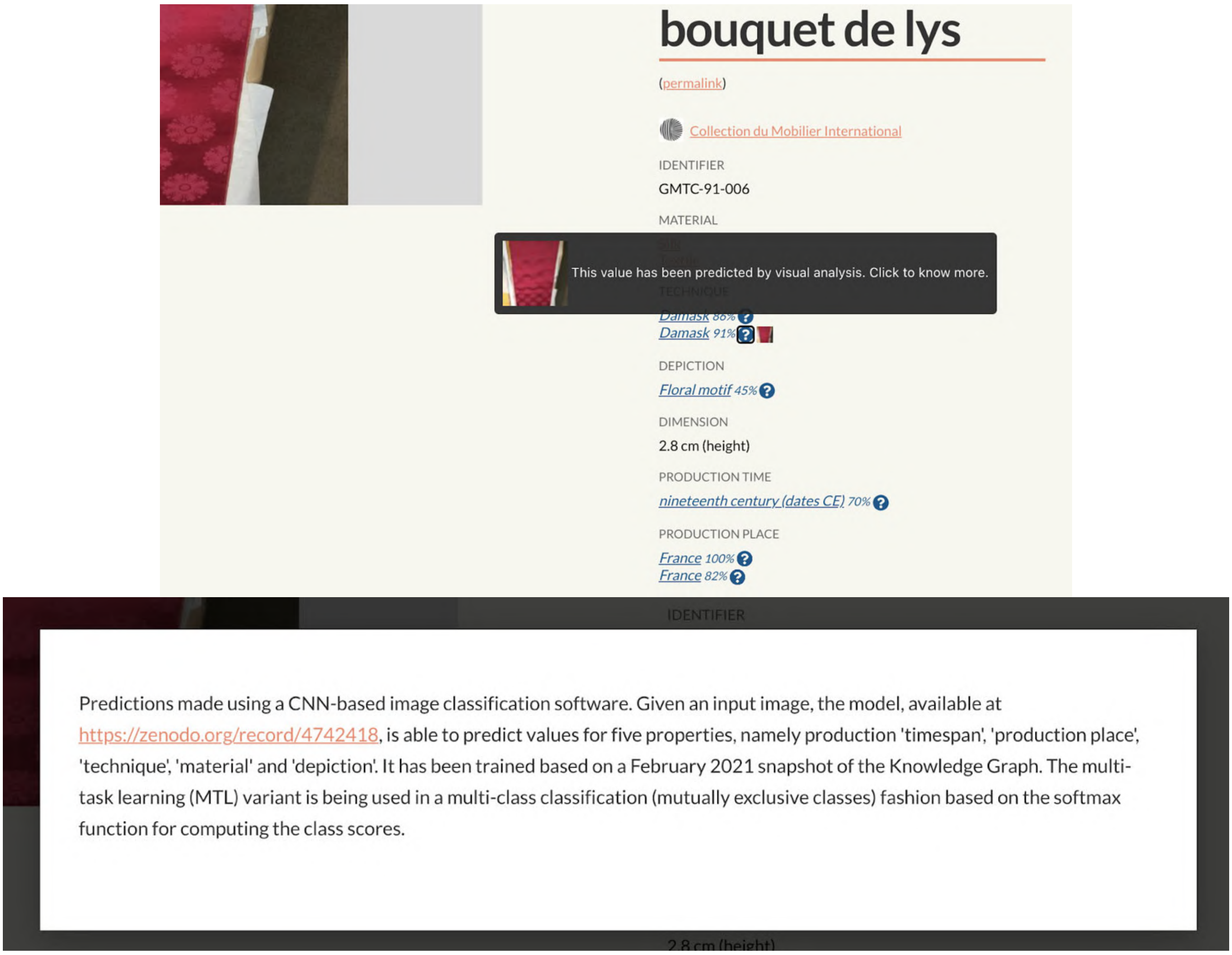


Figure 4. Example of a prediction made by the image analysis module displayed in ADASilk.

3.3. Evaluation

3.3.1. Image Classification

This section describes the evaluation of the CNN model that was used for integration into ADASilk, i.e. the recommended model that was identified in Section 3.1.1 and obtained using the hyper-parameters given in Table 1. Section 3.3.1 gives some statistics of the dataset that was used for evaluation whereas the results are presented in Section 3.3.2.

3.3.1.1. Dataset

The dataset used for training and testing of the CNN to be used for the integration of the image classification module into ADASilk is based on the state of the SILKNOW Knowledge graph on 21/05/2021. It was generated using the function `silknow_image_classification.create_dataset_parameter` with the same parameters settings and using the same procedure as described in D4.6. The only difference in the procedure is that the contents of the variable *category_group/type_a_group* are considered in a slightly different way. Whereas in D4.6, only samples for which this variable had the value *fabric* were considered, due to improvements of the Knowledge Graph, objects having multiple values for that variable can also be considered to correspond to plain silk fabrics if one of the values is *fabric*. Details about the procedure and the the criteria used in this process can be found in D4.6. As a consequence, a dataset consisting of about **48850 images with annotations** that fulfill the requirements of the image classification module was generated. The samples come from **12 different collections** already mentioned in Section 3.1.1. The dataset was randomly split into five subsets to be used in five-fold cross validation. The class structures as well as the class distributions of the five semantic variables to be predicted, namely *Production Material*, *Production Place*, *Production Technique*, *Production Timespan* and *Depiction*, are shown in Tables 6-10. The tables show that the class distributions are still rather unbalanced, especially for the variables *Production Material* (Table 6) and *Depiction* (Table 10). For some variables, the percentage of samples without annotation (indicated by a value of `nan` in the data prepared for classification by the image classification module; cf. deliverable D4.6) is very high. For instance, only 22.2% of the samples have an annotation for the variable *Production Technique* (Table 8); for *Depiction*, this number is only 5.9%.

C	*animal_fibre*	*vegetal_fibre*	*metal_thread*
#	27252	3891	4208

Table 6. Classes (*C*) and number of occurrences (#) of classes for the variable *Production Material*. The number of samples without annotation (`nan`; cf. deliverable D4.6) is 13499 (27.6%).

C	*ES*	*IT*	*JP*	*IR*	*IN*	*CN*	*FR*	*TR*
#	4708	4700	1097	1294	2353	1399	7379	593
C	*GB*	*US*	*GR*	*NL*	*BE*	*DE*	*JM*	*PK*
#	7998	357	479	455	648	592	191	352

Table 7. Classes (*C*) and number of occurrences (#) of classes for the variable *Production Place*. The abbreviations are the country codes given in the SILKNOW knowledge graph. The number of samples without annotation (`nan`) is 14027 (28.7%).

C	*damask*	*embroidery*	*other technique*	*velvet*	*tabby*
#	2768	6861	2526	3051	185

Table 8. Classes (*C*) and number of occurrences (#) of classes for the variable *Production Technique or Procedure*. The number of samples without annotation (`nan`) is 33104 (67.8%).

C	*15th_c.*	*16th_c.*	*17th_c.*	*18th_c.*	*19th_c.*	*20th_c.*
#	685	1829	3378	8423	9975	4012

Table 9. Classes (*C*) and number of occurrences (#) of classes for the variable *Production Timespan*. The classes correspond to centuries. The number of samples without annotation (nan) is 20548 (42.1%).

Class	*flower*	*plant*	*geometrical_shape*
Occurrences	2352	336	202

Table 10. Classes (*C*) and number of occurrences (#) of classes for the variable *Depiction*. The number of samples without annotation (nan) is 45960 (94.1%).

3.3.1.2. Evaluation of the CNN integrated into ADASilk

The multi-task CNN recommended for the integration into ADASilk (cf. Section 3.1.1) was evaluated using the dataset described in Section 3.3.1.1 and five-fold cross validation. That is, the experiment was repeated five times, each time using one of the five subsets produced by the data preparation function for testing and the others for training. Thus, every sample contributed to the evaluation of each variable for which it contained an annotation once. The following quality metrics, already defined in deliverable D4.6, are reported (mean values over all five tests):

- *Overall accuracy*: the percentage of correct predictions. It is presented for individual variables and as an average value over all variables.
- *Recall*: the percentage of samples belonging to a class in the reference that were also assigned to that class by the classifier. It is given for every class and every variable. Average values per variable are also reported.
- *Precision*: the percentage of samples assigned to a class by the classifier that also correspond to that class in the reference. It is given for every class and every variable. Average values per variable are also reported.
- *F1 score*: the harmonic mean of precision and recall. It is a combined metric for every class that is affected both by false positive and by false negative samples of a class. It is given for every class and every variable. The mean F1 score over all classes of a variable is also presented. It is a quality metric that is more susceptible to problems with underrepresented classes than the overall accuracy.

The quality metrics achieved using the classifier used for the integration in ADASilk for the five variables considered are shown in Tables 11-15. The **mean overall accuracy** over all variables is **66.7%**, the **average mean F1 score** over all variables is **49.4%**.

In general, the results are similar to those reported in deliverable D4.6, although the actual quality indices differ; for better comparison, the tables also show the average F1 scores and the overall accuracy reported there. There are considerable differences in the quality indices obtained for the individual variables. A common trend is that the class-specific metrics (precision, recall, F1) heavily depend on the number of samples per class: as it was already the case in deliverable D4.6, underrepresented classes achieve lower class-specific quality metrics. Compared to D4.6, the overall accuracies are better for the variables *Production Material*, *Production Technique* and *ProductionTtimespan*, whereas they are worse for the other two variables. The mean F1 scores are slightly worse for all variables, mostly because the new classes, i.e., those not considered, have only relatively few samples and, thus, are not well differentiated from the others. This can especially be seen in the poor F1 scores of the classes *tabby* for the variable *Production Technique* or *15th_c* for *Production Timespan*. As far as the individual variables are concerned, the following statements can be made:

- ***Subject Depicted Type* (Table 11):** The dominating class (*Flower*) achieves an F1 score of 89%, which is the best value achieved for any class and any variable, but for the two minority classes, one of them (*geometrical_pattern*) not having been considered in deliverable D4.6, the corresponding values are considerably lower (41.3% for both classes). As already pointed out in D4.6, apart from the class imbalance, the fact that this was the variable with the smallest number of annotated samples seems to be problematic and also puts a limit to the applicability of image classification, because there remain only three classes with a sufficiently high number of training samples. Note that the large drop in the mean F1 score compared to D4.6 is due to the consideration of the new class, which only performs on a similar label as the minority class (*plant*) in D4.6.

Class	**Precision [%]**	**Recall [%]**	**F1-Score [%]**
flower	89.9	88.8	89.3
plant	45.1	38.1	41.3
geometrical_shape	35.8	50.0	41.3
Average	**56.9%** (D4.6: 67.5%)	**59.0%** (D4.6: 70.5%)	**57.4%** (D4.6: 68.8%)
Overall Accuracy	**80.2%** (D4.6: 86.1%)		

Table 11. Class-specific quality indices for all classes and overall accuracy of the variable *Subject Depicted Type*. The numbers in parentheses are those reported in deliverable D4.6. Classes printed in bold font were not considered in D4.6.

- ***Production Material* (Table 12):** Here, the F1 scores vary between 29.2% (*metal_thread*) and 82.9% (*animal_fibre*), the latter being the dominant class in the dataset. Compared to D4.6, the F1 values for the dominant class have improved and the F1 values for the underrepresented class have become smaller, which is the reason why the average F1 score has become smaller while the overall accuracy has improved. In this case, the class structure was identical to the one reported in D4.6.

Class	**Precision [%]**	**Recall [%]**	**F1-Score [%]**
Animal_fibre	81.7	84.0	82.9
Metal_Thread	32.1	36.4	34.1
Vegetal_fibre	36.8	24.1	29.2

Average	**50.2%** (D4.6: 50.6%)	**48.2%** (D4.6: 49.9%)	**48.7%** (D4.6: 50.3%)
Overall Accuracy	**71.8%** (D4.6: 68.7%)		

Table 12. Class-specific quality indices for all classes and overall accuracy of the variable *Production Material*. The numbers in parentheses are those reported in deliverable D4.6.

- ***Production Place* (Table 13):** For this variable, the variability of the F1 scores is the largest one among all variables. Whereas the F1 score for *ES, IT, FR* and *GB* is between 55.3% and 64.7%, for *JM, US* and *NL* it is only between 13.4% and 16.8%. Again, *ES, IT, FR* and *GB* are the classes having the largest proportion of training samples. Compared to D4.6, the variability of the F1 scores is reduced; both, the minimum and maximum values are nearer to the average. On average, the metrics are somewhat lower than they were in D4.6. This is partly due to the fact that five additional classes are considered (*BE, DE, JM, PK, RU*), all of which have less than 15% of the training samples of the most dominant classes. Except for *BE*, the F1 scores for the new classes are relatively low. It is not entirely clear why the classifier performs comparably well for BE, achieving an F1 score of 43% of that class. The overall accuracy is about 50%, which means that a correct prediction is achieved in 50% of the cases. Whereas this seems to be relatively low, one has to consider the fact that this is the variable for which the highest number of classes is differentiated (17); a random guess of the class label thus is expected to lead to 100/17 = 5.9% of correct decisions.

- ***Production Technique* (Table 14):** In principle, the CNN works very well in this case, delivering a better OA than the one reported in deliverable D4.6. However, there are problems with the new class *tabby*: whereas for the other classes, the F1 scores are between 57% and 83.6%, for tabby it is only 22.3%. This class has by far the lowest number of training samples (185 compared to 2526-6861 of the other classes; cf. Table 8) and it would seem that this imbalance cannot be compensated by the focal loss.

- ***Production Timespan* (Table 15):** Although the class imbalance is not as pronounced as in, e.g., the case of *Subject Depicted Type*, there is also a clear impact of the number of available training samples on the quality of the results per class. For the dominant classes (*18th_c, 19th_c, 20th_c*), the CNN achieves very similar F1 scores in the order of 65%, but for the other variables they are only between 30% and 40%. The new class is the one achieving the lowest F1 score (32.2%), and it is also the one having the lowest number of training samples. Nevertheless, the overall accuracy of 59.4% is slightly increased compared to deliverable D4.6.

Class	Precision [%]	Recall [%]	F1-Score [%]
ES	67.6	62.0	64.7
IT	53.9	52.7	53.3
JP	27.4	24.3	25.7
IR	33.8	43.3	38.0
IN	54.2	44.8	49.0

CN	31.1	27.2	29.0
FR	56.4	58.2	57.3
TR	20.2	36.1	25.9
GB	56.8	54.0	55.4
US	17.9	14.6	16.1
GR	26.6	29.7	28.1
NL	15.5	18.2	16.8
BE	37.8	50.6	43.3
DE	32.7	30.9	31.8
JM	10.4	18.9	13.4
PK	27.6	35.2	30.9
RU	24.1	27.2	25.6
Average	**35.0%** (D4.6: 39.5%)	**36.9%** (D4.6: 39.6%)	**35.5%** (D4.6: 39.5%)
Overall Accuracy	**50.2%** (D4.6: 53.4%)		

Table 13. Class-specific quality indices for all classes and overall accuracy of the variable *Production Place*. The numbers in parentheses are those reported in deliverable D4.6. Classes printed in bold font were not considered in D4.6.

Class	Precision [%]	Recall [%]	F1-Score [%]
damask	70.3	68.9	69.6
other technique	55.5	58.6	57.0
embroidery	83.9	83.2	83.6
Velvet	70.1	67.8	68.9
Tabby	17.4	30.8	22.3
Average	**55.1%** (D4.6: 69.4%)	**56.0%** (D4.6: 69.8%)	**55.2%** (D4.6: 69.6%)
Overall Accuracy	**71.9%** (D4.6: 70.8%)		

Table 14. Class-specific quality indices for all classes and overall accuracy of the variable *Production Technique*. The numbers in parentheses are those reported in deliverable D4.6. Classes printed in bold font were not considered in D4.6.

Class	Precision [%]	Recall [%]	F1-Score [%]
15th_c.	26.0	42.5	32.3
16th_c.	34.6	32.5	33.5
17th_c.	42.9	37.7	40.1
18th_c.	61.7	64.9	63.2
19th_c.	68.2	65.6	66.9
20th_c.	65.5	65.7	65.6
Average	**49.8%** (D4.6: 52.7%)	**51.5%** (D4.6: 50.6%)	**50.3%** (D4.6: 51.5%)
Overall Accuracy	**59.4%%** (D4.6: 57.6%)		

Table 15. Class-specific error quality indices for all classes and overall accuracy of the variable *Production Timespan*. The numbers in parentheses are those reported in deliverable D4.6. Classes printed in bold font were not considered in D4.6.

The CNN-based image classification module is able to predict the class labels of well represented classes with a F1 score of up to 89%. All classes of all variables are on average classified with a F1 score of 49.4% where in total 66.7% of the evaluated predictions were correct. Despite the availability of more training samples compared to deliverable D4.6, these numbers are slightly worse (e.g. by 0.6% in terms of the number of correct predictions), but this result is achieved under consideration of more classes, which makes the results more relevant for a user. Whereas there is certainly room for improvement, the classification results provide additional information that is correct in most cases (indicated by the OA) and, thus, can be considered to be useful in the context of SILKNOW. However, the quality of the predictions is probably not good enough for samples with unknown annotations for the five semantic variables in question. Nevertheless, the evaluation shows that it does indeed make sense to integrate the predictions in the *group* fields of the knowledge graph and to present them to a user in ADASilk, where it is made clear that this content is the result of automated processing and where the class scores are also presented to act as a measure of uncertainty of the prediction. In this way, additional information about the silk fabrics in the knowledge graph is generated, while the additional information about the way in which it was generated can help the domain experts using the knowledge graph for further analyses to put the corresponding information in perspective.

As already pointed out in deliverable D4.6, future research work beyond the SILKNOW project should focus on additional methods for compensating imbalanced class distributions of the training samples.

3.3.2. Text Analysis

In this section, we report on the specifics of producing the final predictions for the missing values in the Knowledge Graph based on train and test data exported on **21/05/2021.** Only labels with a minimum of 50 occurrences were included. In Table 16 we show the accuracy summary of the evaluation of the classifier on a subset of the labeled data compared with previous deliverables. Note that the records, the splitting into train, development, and test sets are different, and the number of labels is different than reported in D3.4 and thus, it is expected that the results differ. The results for D3.3 were also different in those aspects and in the class structure of the labels, as explained in D3.4, the labels were grouped by domain experts before that deliverable.

Deliverable	Place	Time	Technique	Material
D3.3	82%	92%	93%	87%
D3.4	92%	72%	94%	80%
D6.7 (current)	93%	83%	93%	82%

Table 16. Comparison of accuracy as percentage with previous deliverables, per task.

For each semantic variable we show the per label counts in the training data and the number of predictions for the missing values in the Knowledge Graph (unlabeled data). This is shown in Tables 17-20.

Label	Present in Training Data	Predicted for Test Data
animal_fibre	7562	4880
vegetable_fibre	1562	591
metal_thread	1262	30
Total	10386	5501

Table 17. Production Material labels and respective counts present and predicted in data exported from the Knowledge Graph.

Label	Present in Training Data	Predicted for Test Data
Damask	1300	23603
embroidery	1274	6051
other_technique	910	5120
Velvet	491	1389
Tabby	50	1271
Total	4025	37434

Table 18. Production Technique labels and respective counts present and predicted in data exported from the Knowledge Graph.

Label	Present in Training Data	Predicted for Test Data
nineteenth century (dates CE)	6144	3164
eighteenth century (dates CE)	3534	501
twentieth century (dates CE)	2463	76
seventeenth century (dates CE)	1014	46
sixteenth century (dates CE)	633	49
fifteenth century (dates CE)	266	1
Total	14054	3837

Table 19. Production Timespan labels and respective counts present and predicted in data exported from the Knowledge Graph.

Label	Present in Training Data	Predicted for Test Data
ES	3756	4034
GB	3330	553
FR	2239	5211
IT	1314	786
IN	1081	63
CN	908	182
IR	521	32
JP	283	47
TR	278	22
BE	173	40
PK	171	0
DE	150	12
GR	128	1
RU	111	36
NL	93	0
US	62	11
Total	14598	11030

Table 20. Production Place labels and respective counts present and predicted in data exported from the Knowledge Graph.

The text classification tool is available at https://github.com/silknow/text-classification. The repository includes code to replicate these predictions under the directory experiments/predictions. In particular, the script run_exp.sh[2] includes each step, in sequential order and can be used in conjunction with the data exported from the Knowledge Graph to replicate the results presented in this deliverable. The models trained for this experiment are also available online[3] and can be used with the tool on different data formatted as a tab-separated-values (TSV) with at least an "obj" identifier and a "text" column or using the included REST server. These and other ways to run the text classification tool were described in D3.3 and documented together with the tool.

3.3.3. Categorical Classification with Gradient Tree Boosting

We used this classifier to predict the missing values in the Knowledge Graph. For each semantic variable we show the per label counts in the training data and the number of predictions in the test set based on data exported from the Knowledge Graph on **21/05/2021**. Only fabrics were used, with other types of objects discarded from the training data. Only labels with more than 50 occurrences were included. Table 21 shows the accuracy summary results of evaluating this classifier on a labeled subset of the data compared to D3.4 where it was introduced. The data is slightly different resulting from added records, different number of labels, and different splitting of the data into train, development, and test subsets.

Deliverable	Place	Time	Technique	Material
D3.4	45%	66%	70%	77%
D6.7 (current)	58%	66%	72%	82%

Table 21. Comparison of accuracy as percentage with previous deliverable, per task.

For each semantic variable we show the per label counts in the training data and the number of predictions for the missing values in the Knowledge Graph (unlabeled data). This is shown in Tables 22-25.

Label	Present in Training Data	Predicted for Test Data
animal_fibre	15263	5962
vegetable_fibre	1705	30
metal_thread	1785	279
Total	18753	6271

Table 22. Production Material labels and respective counts present and predicted in data exported from the Knowledge Graph.

[2] Commented script for generating predictions: https://github.com/silknow/text-classification/blob/master/experiments/predictions/run_exp.sh
[3] SILKNOW Text Classifier Models: https://zenodo.org/record/5070696

Label	Present in Training Data	Predicted for Test Data
Damask	1637	2120
embroidery	2442	31559
other_technique	1086	5837
Velvet	2097	1099
Tabby	133	0
Total	7395	40615

Table 23. Production Technique labels and respective counts present and predicted in data exported from the Knowledge Graph.

Label	Present in Training Data	Predicted for Test Data
nineteenth century (dates CE)	3464	368
eighteenth century (dates CE)	3619	3168
twentieth century (dates CE)	2406	359
seventeenth century (dates CE)	812	0
sixteenth century (dates CE)	670	0
fifteenth century (dates CE)	286	0
Total	11257	3895

Table 24. Production Timespan labels and respective counts present and predicted in data exported from the Knowledge Graph.

Label	Present in Training Data	Predicted for Test Data
ES	2411	4846
GB	1465	205
FR	4116	4273
IT	2982	3841
IN	442	3404
CN	458	0
IR	543	420

JP	455	0
TR	257	0
BE	77	0
PK	53	0
DE	400	0
GR	264	0
RU	130	0
NL	181	0
US	200	0
CH	82	0
EG	65	0
Total	14581	169891

Table 25. Production Place labels and respective counts present and predicted in data exported from the Knowledge Graph.

The code for this classifier and for replicating these predictions is also available on github at https://github.com/silknow/text-classification/tree/master/experiments/predictions_cat.

4. INTEGRATION OF THE IMAGE RETRIEVAL COMPONENT

4.1. Overview

The SILKNOW image retrieval module was developed in the context of WP4. Its general principles were originally described in deliverable D4.5. The integration into ADASilk is based on the final version of the module, which is described in deliverable D4.6. It is the goal of the image retrieval to search the SILKNOW knowledge graph using images as input. A user presents an image, referred to as query image, to the module, and the image retrieval module delivers the object URIs of the records in the SILKNOW knowledge graph that are expected to be most similar to the object shown in the query image. The number *k* of objects retrieved is a parameter that can be set by the user. It is expected that the search results will contain some records that are meaningful so that the user can get some additional information about the context of the object shown in the query image.

Image retrieval is based on a Convolutional Neural Network which takes a digital image as its input and converts it into a feature vector (referred to as *descriptor*) that is supposed to capture the main characteristics of the image. Images being similar according to some definition are expected to have similar descriptors and, thus, the Euclidean distance of two descriptors can be used as a measure for the similarity of the corresponding images: the smaller the descriptor distance, the more similar a pair of images is expected to be. Having trained the CNN to produce similar feature vectors for similar images, it is used to determine a descriptor for every image available in the SILKNOW knowledge graph. These descriptors are stored in a kd-tree, which serves as a spatial index for fast nearest neighbor search. In that index, each descriptor is also linked to the corresponding record in the SILKNOW knowledge graph. At test time, a query image is presented to the CNN to predict a descriptor. Afterwards, image retrieval itself consists of finding the *k* descriptors that are most similar to the query descriptors (i.e. the k descriptors having the smallest Euclidean distance from the query descriptor) in the kd-tree containing the descriptors of all images in the SILKNOW knowledge graph. This also identifies the *k* records expected to be most similar to the query image. Details about the overall workflow and the structure of the CNN can be found in deliverable D4.5.

The biggest challenge in the development of the image retrieval module was the generation of appropriate training samples. A training sample consists of a pair of images with known similarity status (*similar* vs. *dissimilar*). However, similarity is a subjective property of an image pair, so a manual annotation of image pairs, besides being a tedious and time-consuming task, is expected to lead to very heterogeneous results if multiple persons with different backgrounds and preferences are involved. Thus, it was decided to avoid manual annotation and derive the training data from the SILKNOW knowledge graph automatically according to some objective rules for what constitutes similarity of silk fabrics. As the training of the CNN requires the minimization of a loss function, each rule for defining similarity was used to formulate one such loss function term. In the final version of the image retrieval module, described in deliverable D4.6, there are two basic definitions of similarity, for both of which two loss function terms were introduced:

1) **Similarity based on semantic properties:** This is a concept of similarity that entirely disregards the visual appearance of an object. Rather than that, it focuses on the semantic

properties of a record in the SILKNOW knowledge graph. Informally speaking, the more semantic properties of two objects are identical, the more similar the corresponding objects are expected to be. If trained on the basis of this concept of similarity, the image retrieval module is expected to find records in a database that have similar semantic properties (e.g., the same material or the same production place) as the query image. This concept of similarity is considered by two loss function terms that are related to two different aspects of semantic similarity:

a) **Semantic Similarity Loss E_t**: This loss considers triplets of images consisting of an anchor image, a positive sample and a negative sample, where the positive sample is an image that is more similar to the anchor image than the negative one. Similarity is measured based on the level of agreement of the annotations of samples for the five variables *production timespan*, *production place*, *production material*, *production technique* and *subject depicted type* as contained in the SILKNOW knowledge graph. This is a gradual definition of similarity rather than a binary one, and it also considers the uncertainty introduced by incomplete samples, i.e. by samples for which the information about some of the variables is missing in the knowledge graph. The corresponding triplet loss E_t tries to pull the descriptors of the anchor and the positive sample closer to each other while pushing the descriptors of the anchor and the negative sample away from each other.

b) **Domain Expert Loss E_r:** Analysing the results of the first version of the image retrieval module, which was only based on the semantic similarity loss, the cultural heritage experts came up with some rules to identify pairs of similar records. These rules are quite specific and, thus, they only identify a relatively small set of similar pairs, but a pair being consistent with one of these rules is certain to be similar, so that the information contained in that pair is considered to be stronger than the soft definition of similarity that forms the basis of the semantic similarity loss E_t. The domain expert loss E_r simply aims at minimizing the Euclidean distance of the descriptors of two images considered to be similar according to one of the rules formulated by the cultural heritage experts.

2) **Similarity based on visual appearance:** This is a concept of similarity that entirely disregards the semantic properties of an object, and it is less obvious how to integrate it into the training process of a CNN if no manual annotation is to be carried out. Following discussions with cultural heritage experts in the context of the evaluation of the image retrieval module in the context of task T7.1, it was decided to use colour as the main visual cue, because it can also be described numerically by mathematical concepts, so that the generation of training samples considering visual aspects can be automated. Visual appearance is also considered by two loss function terms that are related to two different aspects of visual appearance:

a) **Colour Similarity Loss E_c**: This loss is based on the similarity of the distribution of colours in the images that are to be compared. It considers a 2D histogram of colour hue and saturation after transferring the RGB (red green blue) vectors of an image into the HSI (hue saturation intensity) colour space. The degree of similarity of two images is measured by the cross correlation coefficient of these 2D histograms: the larger the correlation the more similar two images are supposed to be. The colour similarity loss E_c aims at achieving a small descriptor distance for images having a high correlation of their colour histograms and vice versa.

b) **Self-similarity Loss E_s:** This loss is actually a kind of regularization loss, encoding that the descriptors produced for two images of the same object should be very close to each other. It is implemented as a variant of data augmentation, i.e., for each image, a synthetic image is derived by applying some geometric and / or radiometric transformation to the original, and the original and the synthetic image are considered to be similar. Like the domain expert loss, the self-similarity loss E_s simply aims at minimizing the Euclidean distance of the descriptors of the augmented image and the original one.

The total loss that is to be minimized in training is a linear combination of the four loss function terms just described. By varying the weights of the individual loss terms, different scenarios of what constitutes similarity can be defined by an application. In the evaluation of the image retrieval module performed in the context of WP4 and reported in deliverable D4.6, five such scenarios were compared (Scenarios A-E). The results of the evaluation depended on the evaluation criteria. Consequently, in D4.6, two of the evaluated scenarios were identified as candidates for the integration of the image retrieval module into ADASilk:

1) **Scenario B: Visual Similarity**. This scenario only considered the colour and self-similarity terms in the training procedure, both of them with equal weight. It was found to produce the highest percentage of meaningful results according to an evaluation by the cultural heritage domain experts from UNIPA. It has to be noted that the ability of the module to produce results of a similar colour was one of the criteria used in that evaluation.

2) **Scenario E: Semantic and Visual Similarity.** This scenario considered all of the four implemented similarity terms with equal weight. It was found to deliver the best results if the evaluation criterion was based on semantic aspects, i.e., if the success of image retrieval was measured by the module's capability of retrieving images having semantic properties similar to those of the query image.

This is consistent with deliverable D2.4, which emphasized the user requirement to provide the image retrieval function based on two different definitions of similarity, one focusing on semantics and the other one being based on visual aspects of similarity. Consequently, these two scenarios were integrated into ADASilk. The way in which this was done is described in section 4.2.

4.2. Integration with ADASilk

For both recommended scenarios, CNNs trained using data exported from the SILKNOW knowledge graph on **19/02/2021**, which corresponds to a relatively advanced state of the knowledge graph, were made available for integration into ADASilk; cf. Section 4.3.1 and D4.6 for a description of the dataset and the training procedure. The kd-trees serving as spatial indices for image retrieval were not provided because additional collections were still being integrated into the SILKNOW knowledge graph at the time of writing of deliverable D4.6. Consequently, they had to be generated in the context of the work leading to this deliverable.

For the integration, the API of the image retrieval software from D4.6 had to be slightly changed. The reason for this change is that the function `silknow_image_retrieval.get_kNN_parameter` as documented in that deliverable would load the CNN parameters and the kd-tree from the hard disk, which made the response time of image retrieval too slow. Consequently, two new functions for loading the CNN parameters and the kd-tree into memory were implemented:

- `silknow_image_retrieval.preload_cnn_model`: this function takes the path to the folder containing the trained CNN model (`model_dir`) resulting from the function `silknow_image_retrieval.train_model_parameter()`as an input, loads the model architecture as well as the trained model parameters into memory and returns the image retrieval CNN in the form of the parameter `loaded_model`. A detailed description about the input and output parameters can be found in the documentation of the `silknow_image_retrieval` package.
- `silknow_image_retrieval.preload_kd_tree`: this function takes the path to the folder containing the kd-tree (`tree_dir`) resulting from the function `silknow_image_retrieval.build_kDTree_parameter()`as an input, loads the kd-tree into memory and returns the loaded tree as well as further information about the images whose descriptors constitute the kd-tree in the form of the parameters `tree`, `labels_tree`, `data_dict_train`, `relevant_variables`, `label2class_list`. A detailed description about the input and output parameters can be found in the documentation of the `silknow_image_retrieval` package.

Furthermore, an additional variant of the function for image retrieval requiring the CNN and the kd-tree to have already been loaded is provided:

- `silknow_image_retrieval.get_kNN_from_preloaded_cnn_and_tree`: this function takes the preloaded CNN model (`model`) and the preloaded kd-tree (`tree`, `labels_tree`, `data_dict_train`, `relevant_variables`, `label2class_list`) resulting from the functions described above as inputs as well as further input parameters (`master_file_retrieval`, `master_dir_retrieval`, `pred_gt_dir`, `num_neighbours`, `bool_labeled_input`, `multi_label_variables`) defining the image retrieval setup. The latter set of parameters was already required by the former image retrieval function `silknow_image_retrieval.get_kNN_parameter()`, where a detailed description of those parameters can either be found in deliverable D4.6 or in the documentation of the `silknow_image_retrieval` package.

Consequently, the CNN and the kd-tree only need to be loaded once. After that, whenever the user uploads an image and pushes the button to retrieve its k most similar images from the knowledge graph, the new image retrieval function is invoked. It uses the pre-loaded CNN to compute the descriptor of the image and then it retrieves the k-nearest neighbors based on the descriptors in the pre-loaded kd-tree. Whereas loading the CNN and the kd-tree might take up to 1 minute, the image retrieval itself is very fast and delivers results with less than a few seconds.

The version of the SILKNOW image retrieval software that was integrated into ADASilk, i.e. the version of D4.6 expanded by the new functions, is available on github at https://github.com/silknow/image-retrieval

The steps required to do produce a kd-tree are:

- export records from the Knowledge Graph;
- convert them into the format required by the image retrieval software (for which the latter offers a function);
- build the kd-tree using another function of the image retrieval software.

The GUI differentiates these methods by the wording of the corresponding control items ("visually similar images" - Scenario B - vs. "objects with similar properties" - Scenario E); the detailed wording has been decided upon in collaboration between UVEG and EURECOM. The number k of nearest neighbours retrieved by a query is a parameter that can be tuned by the user (default value: 10)

There are two use cases:

- retrieve similar images for images of records already available in the knowledge graph;
- retrieve images for an image uploaded by the user.

The image retrieval module is deployed as a microservice which is used to bridge the gap between the frontend interface of ADASilk and the functionalities of the API. It is developed in Python and exposes a basic HTTP server using Flask. The application loads the models used for visual and semantic retrieval, by calling `` `preload_cnn_model(model_path)` `` with `` `model_path` `` being the path to the model file. One route is available, namely POST /retrieve, which takes an image as an input. When the route is called, the server proceeds with the following sanity checks:

- Making sure that the file size is not over 5 MB.
- Verifying that the file is a valid image, by using the native Python module `imghdr` which returns the format of an image (e.g., "jpeg", "png").

The image is then temporarily saved on the disk, and the service calls

`` `get_kNN_from_preloaded_cnn_and_tree` ``

from the image retrieval API with the path to the image file. The API returns predictions as JSON. Each prediction contains the URI of the predicted object, which is passed to the search API of ADASilk. The source code of the microservice is available at https://github.com/silknow/image-retrieval-server

On the frontend of ADASilk, there are two ways of interacting with the image retrieval module:

- Through the homepage, by manually uploading a photo.
- On the details page of an object, by clicking on the “View similar” button.

In both cases, ADASilk calls the `/retrieve` route with either the user’s photo or the photo of an existing object, then fetches the details of each object returned by the predictions, before displaying them to the user. In the case of viewing similar results based on an existing object, its photo and name will be displayed on the search results page. Alongside the photo, the user can also decide between two models: “Visually similar images”, or “Objects with Similar Properties”, and switch between them at any time on the search page.

Let's imagine that a user is willing to search for similar objects than the one illustrated by the record identified by https://collection.cooperhewitt.org/objects/18569441/ in the Cooper Hewitt museum.

Figure 4. Textile (Italy), 1750–75; silk and metallic thread on silk; Overall: 48.3 x 40.6 cm (19 x 16 in.); Gift of Mrs. Robert Woods Bliss, Dumbarton Oaks Collection; 1943-46-17.

ADASilk will return the 20 objects depicted in Figure 5 when choosing "Visually Similar Images" and the 18 objects depicted in Figure 6 when choosing "Objects with Similar Properties".

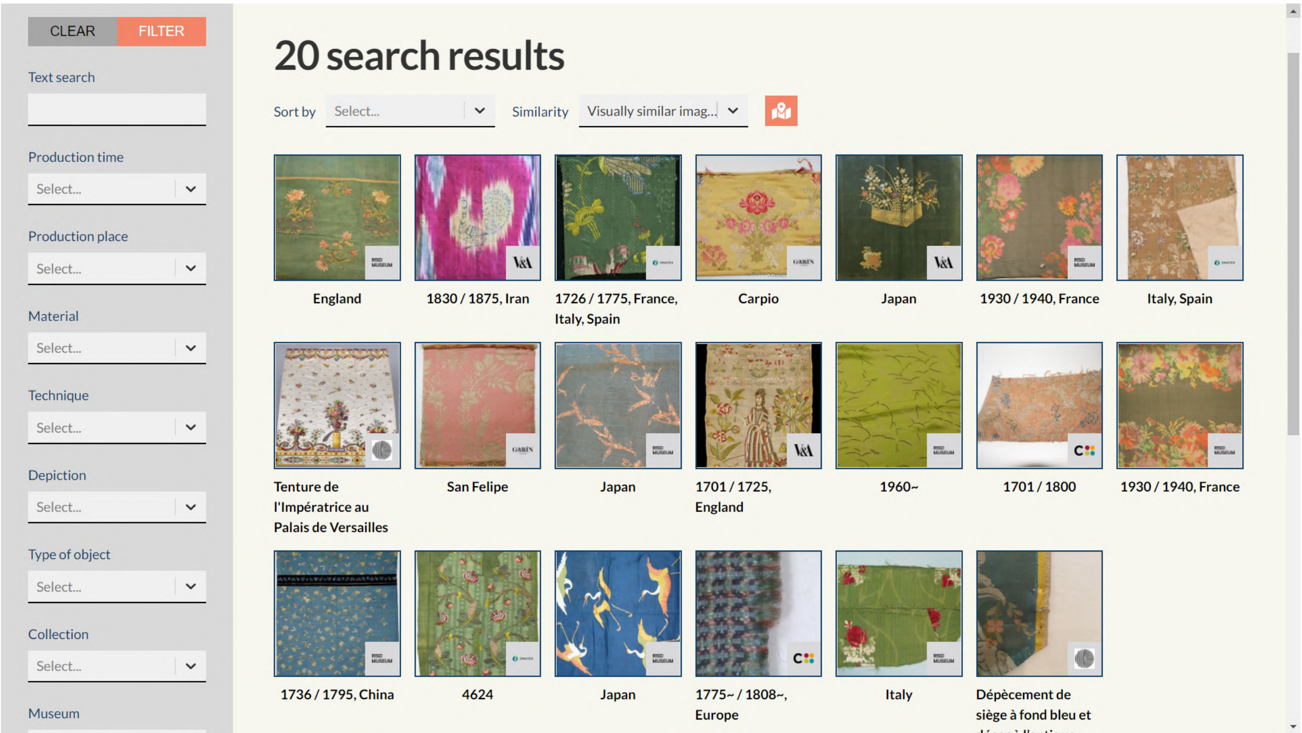


Figure 5. Results from the Image-Based Retrieval module when selecting "visually similar".

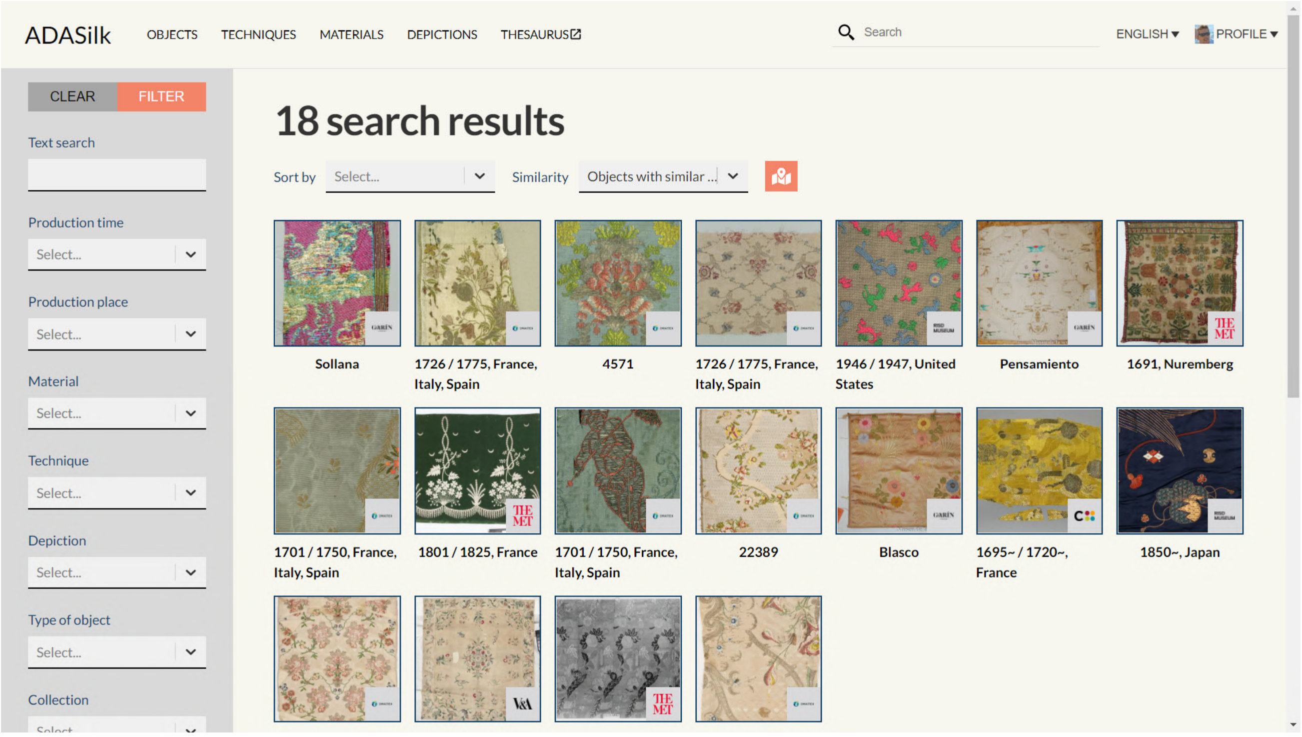


Figure 6. Results from the Image-Based Retrieval module when selecting "objects with similar properties".

When browsing the SILKNOW knowledge graph, the user can also see related or recommended objects. For example, if the user is exploring the object from the Art Institute of Chicago identified by https://ada.silknow.org/object/00865dd1-6be2-3e21-93a7-e7328e9a57a7, he can see recommended objects that are either visually similar (Figure 7) or semantically similar (Figure 8).

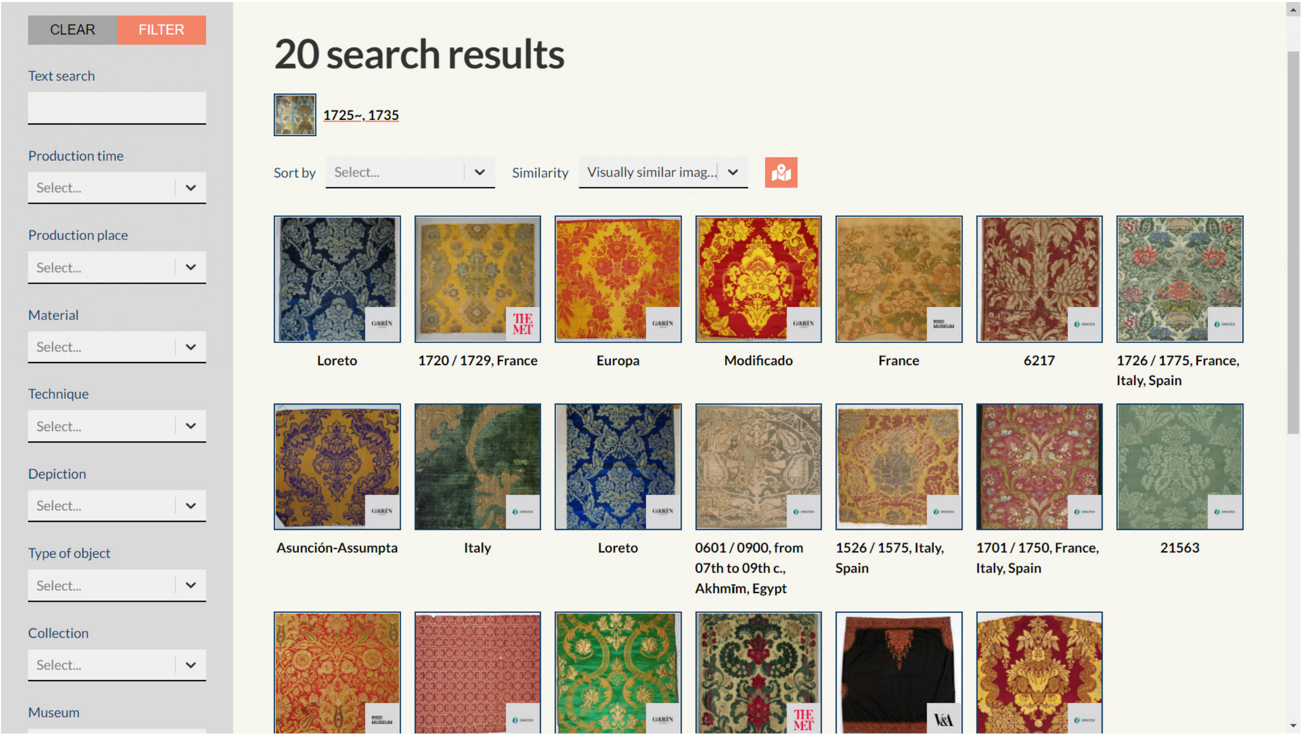


Figure 7. Objects that are visually similar with respect to https://ada.silknow.org/object/00865dd1-6be2-3e21-93a7-e7328e9a57a7

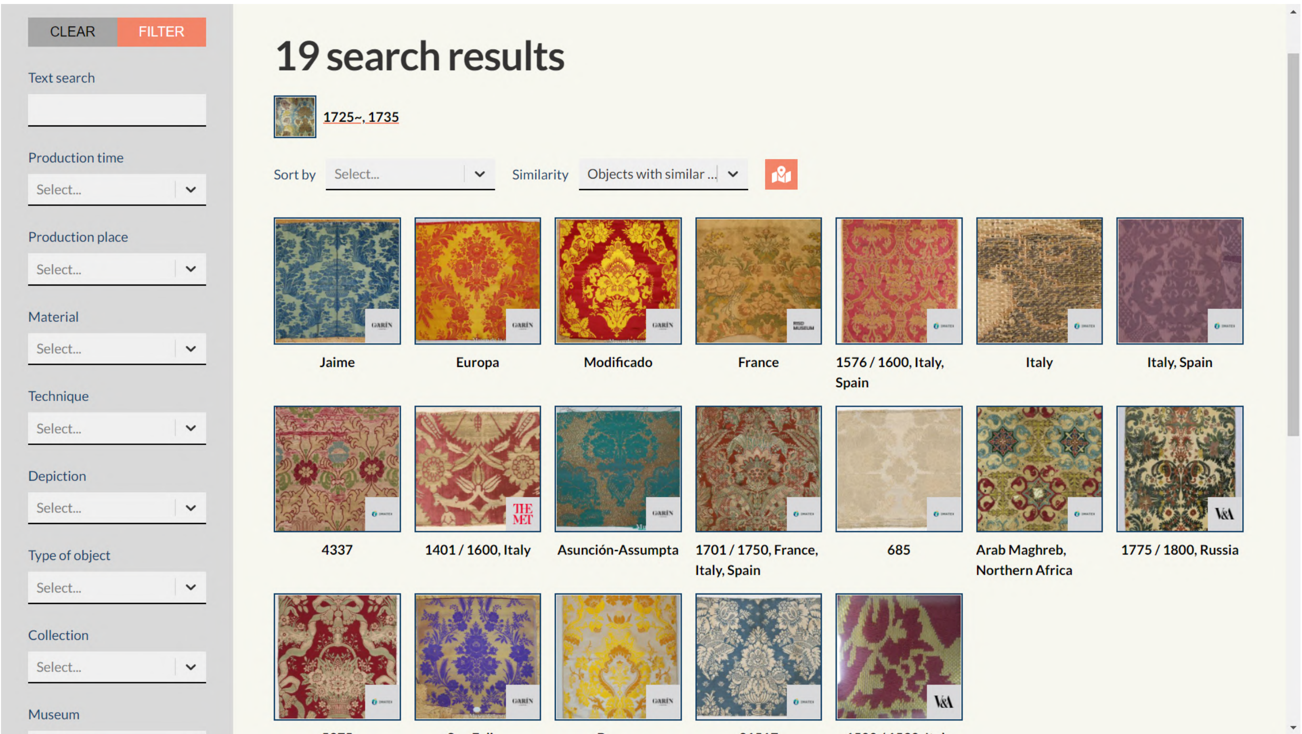


Figure 8. Objects that are semantically similar with respect to https://ada.silknow.org/object/00865dd1-6be2-3e21-93a7-e7328e9a57a7

4.3. Evaluation

The two CNNs that were integrated into ADASilk were exactly those recommended in deliverable D4.6, and the parameters of the two CNN variants were exactly those determined in the context of D4.6. As that deliverable contains a detailed evaluation of the image retrieval module, which includes an evaluation of exactly the same models that were integrated into ADASilk, no additional evaluation of that module was performed in the context of WP6. In this section, the main findings of the evaluation reported in D4.6 are summarized; for more details, please refer to section 5 of D4.6.

4.3.1. Data used for evaluation

The data used for the evaluation (and, consequently, also for training the CNN models that were integrated into ADASilk) are based on the SILKNOW knowledge graph in its state on 15/12/2020. The exported data only considered samples corresponding to plain silk fabrics according to the value of the variable *category*_group in the knowledge graph. As described in D4.6, the data preparation function of the SILKNOW image retrieval module was used to convert the raw exported data into a format that could be used for training and testing the module. In this context, samples with annotations for at least one of the five semantic variables identified in section 4.1 were considered; classes with fewer than 150 samples were discarded. This data preparation process resulted in two sets of training samples: the *labelled subset* consisting of 25,825 images with annotations, which was extracted from the knowledge graph export, and the *rules subset* consisting of 1,087 images which were affected by one of the cultural heritage domain experts' rules for similarity (cf. Section 4.1). These subsets were not disjoined; only 849 of the images in the *rules subset* were not contained in the labelled dataset. The number of classes per variable varied between 2 for *subject depicted type* and 12 (*production place*); for *production timespan* and *production technique*, 5 and 4 classes were differentiated, respectively; for *production material*, the number of classes was 3, but in this case, multiple annotations were allowed for each sample because a silk fabric could be made of multiple materials. The class distribution was very imbalanced for most variables.

4.3.2. Evaluation strategy

The evaluation of the module focused on two different aspects:

1) **Empirical evaluation based on semantic aspects**: This part of the evaluation focuses on the ability of the module to retrieve images that are semantically similar to the query image. For that purpose, the *k* retrieved records are used for a k nearest neighbor classification: for every variable, the annotations of the retrieved images are analyzed; each image casts a vote for its class, and the class achieving the maximum number of votes is considered to be the predicted class label of the query image. Based on a comparison of the predicted class labels to reference annotations exported from the knowledge graph, a confusion matrix can be derived, on the basis of which additional quality indices can be derived. In D4.6, the focus was on the overall accuracy and the mean F1 scores (cf. Section 3.3.1.2). These two quality measures are determined for each variable; average values over all variables are also reported. For variable *production material*, for which multiple annotations were allowed, two variants of these quality indices were determined, which

differed in the way in which samples assigned to no class by the classifier were considered; please refer to D4.6 for details.

For the semantic evaluation, the dataset described in section 4.3.1 was split into five disjoint subsets that formed the basis of five-fold cross validation. That is, each experiment was repeated five times, each time using five subsets for training and the fifth subset for evaluation; in each of these test runs, another subset served as the test set, so that each sample was considered for testing once. The reported quality measures are averages of the five test runs.

2) **Evaluation by cultural heritage experts**: For this part of the evaluation, the dataset described in section 4.3.1 was split into a test set consisting of 100 query images and a training set consisting of all the samples not corresponding to any of the query images. The training samples were used to train the CNN for image retrieval. Afterwards, the image retrieval results for the 100 test images were presented to cultural heritage experts from UNIPA. The experts classified each retrieved image according to whether they considered it to be a meaningful result or not. The criteria for taking this decision in the evaluation process was based on color, pattern (related to the depicted subject) and general appearance; a retrieved image was considered to be meaningful if it matched the query image in at least two of these criteria according to the cultural heritage experts. Based on this assessment of the retrieved images, the reported quality metrics are the *top-k-score*, i.e. the percentage of query images for which at least one meaningful result is among the k retrieved images, and the percentage P_m of samples for which there are at least m meaningful images among the k_{max} =10 nearest neighbors.

The evaluation procedure started with a hyper-parameter search. For that purpose, the CNN for image retrieval was trained using different values of the hyper-parameters using the semantic similarity loss E_t and the semantic evaluation procedure; the hyper-parameter values identified to give the best results were used in all subsequent experiments and, thus, also for training the models that were integrated into ADASilk. Based on the selected hyper-parameters, five different scenarios were defined, each of them using another set of weights for the combination of the four loss functions mentioned in section 4.1 and, thus, corresponding to a different definition of what constitutes “similarity” of a pair of images. A comparison of the results thus achieved identified the two scenarios B and E, also described in section 4.1, to be recommended for integration into ADASilk. In section 4.3.3, the results of the evaluation of these two scenarios and, thus, of the integrated variants of the image retrieval module are summarized.

4.3.3. Evaluation results

Table 26 presents the overall accuracies and mean F1 scores for all variables for the two scenarios integrated into ADASilk. The table shows that the differences in the quality of the results according to an evaluation focusing on semantic aspects does not differ too much between the two scenarios, even though in deliverable D4.6, they were found to achieve the best and the worst performance, respectively. The difference in overall accuracy is about 2% on average, with a maximum of about 4% for variable *production technique*; for variable *subject depicted type*, the overall accuracy of scenario B is even slightly better than the one of scenario E, though only by a small margin (0.5%). The average mean F1 scores show a similar behavior, except that the differences are slightly more pronounced. On average, the difference of these scores between scenarios B and E is about 3.5%. Here, the results for scenario E are better for all variables, the maximum difference being 4.5% for variable *production technique*. The relatively low values of the F1 scores compared to the overall accuracies indicate that there are problems with individual classes; the analysis of the class-wise performance metrics for scenario E in deliverable D4.6 indicated that these problems mostly occur with underrepresented classes, i.e. classes for which only relatively few training samples are available. The variable for which the best results are achieved in both scenarios is *subject depicted type*, whereas *production place* seems to be the most problematic one. It would seem that it is no coincidence that these variables are those with the smallest and the largest number of differentiated classes, respectively. The results shown in Table 26 show the image retrieval module's capacity to retrieve images having semantic properties similar to those of the query image in many cases. As already pointed out in D4.6, even if the class predicted by the k nearest neighbor classification, i.e. the majority class among the retrieved images, is incorrect, there may still be some images among the retrieved ones which correspond to the correct class. A detailed investigation of this aspect has to be left to future work.

Variable		*Production Material*	*Production Place*	*Production Technique*	*Production Timespan*	*Subject Depicted Type*	**Average**
Scenario B	*OA*	77.1 / 72.6	40.6	57.3	52.5	89.6	63.4 / 62.5
	F1	25.1 / 26.7	22.8	52.8	41.2	62.0	40.8 / 41.1
Scenario E	*OA*	78.2 / 73.4	44.1	61.6	53.8	89.1	65.4 / 64.4
	F1	29.6 / 29.7	26.7	57.3	42.9	65.6	44.4 / 44.4

Table 26. Results of the semantic evaluation of the two scenarios (B and E) of the image retrieval module that were integrated into ADASilk for the five semantic variables considered. These numbers are extracted from tables 57 and 58 of deliverable D4.6. *OA*: Overall accuracy [%]; *F1*: mean F1 score [%]. The last column gives average values over all variables. In case of the variable *Production Material*, the first value refers to the classification results based on the binary classification procedure described in Section 3.1.6 of D4.6; the second value refers to the results including the most probable class of samples assigned to the background for all classes.

Table 27 shows the results of the evaluation by the cultural heritage experts from UNIPA for the two scenarios integrated into ADASilk. Scenario B, which exclusively considers visual similarity aspects in training, i.e. self-similarity and color similarity, clearly outperforms Scenario E and was found to achieve the best evaluation metrics compared to all other scenarios tested in D4.6 by a large margin. Scenario E, while performing considerably worse than Scenario B, was still identified to deliver the second-best results in deliverable D4.6. For Scenario B, the *top-1-score* amounts to 30%, which is 10% better than the one for Scenario E, and the *top-10-score* amounts to 83%, which is 29% higher than the one of scenario E. These numbers mean that in 30% of the cases, the image considered to be the most similar one by the image retrieval module was also considered to be a good match by the cultural heritage expert, while in 83% of the cases there was at least one good match among the 10 retrieved images. In 17% of the cases, the cultural heritage experts found that none of the retrieved images was meaningful according to their criteria. The quality of the 10 most similar retrieved images is indicated by the percentages P_m presented in Table 27. As already indicated by the *top-10-score*, for 83% of the test images, at least *m=1* meaningful result is contained in the retrieved images according to the cultural heritage experts; in the majority of cases (P_m > 50%) there are at least two meaningful images, in about one quarter of the cases there are at least four, and for 2% of the test images there are even *m=8*. However, there was no query image for which 9 or even 10 retrieved images were meaningful. More details about the evaluation can be found in deliverable D6.7.

The evaluation results are not perfect for both scenarios integrated into ADASilk. However, the evaluation shows that image retrieval is capable of delivering meaningful results for both investigated scenarios; Scenario B is to be preferred if the user's focus is on visual aspects, whereas Scenario E delivers better results if semantic properties of fabrics are considered to be more relevant. We believe that the results of the SILKNOW image retrieval module can be a good starting point for a user who wants to explore the SILKNOW knowledge graph using an image as the basis for a search.

k / m		*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
Scenario B	*top-k-score*	30	45	57	65	70	74	76	80	82	83
	P_m	83	69	45	23	12	7	4	2	0	0
Scenario E	*top-k-score*	20	30	34	40	43	45	47	51	54	54
	P_m	54	31	17	7	4	4	0	0	0	0

Table 27. Results of the evaluation by cultural heritage experts of the two scenarios (B and E) of the image retrieval module that were integrated into ADASilk. The quality indices are the *top-k-Scores* [%] for $k \in [1,10]$ and P_m values [%] for $m \in [1,10]$. These numbers are extracted from tables 64 and 65 of deliverable D4.6.

5. INTEGRATION OF THE SPATIO-TEMPORAL MAPS AND VIRTUAL LOOM COMPONENTS

5.1. STMaps

5.1.1. Overview

STMaps is a visual tool implemented in Unity (Unity 2020.2.8.f1 is used to develop the last version of the tool). The use of Unity allows developing a cross-platform application with state-of-the-art graphics. The different releases of the tools are generated like a WebGL plugin. The WebGL technology allows the integration of a software module within a web application, the communication between the plugin and the web application is performed by invoking Javascript methods.

STMaps allows the spatio temporal visualization of knowledge graph data. This software uses and expands on the Visualization ontology (VISO) [7] work in order to define how the knowledge graph data is going to be visualized. The functionalities, configuration and the design of the communication protocol between ADASilk and STMaps are detailed in deliverable D5.5.

The main functionalities of STMaps are:

- Visualization in a 2D/3D navigational environment where the spatio temporal data of the Knowledge Graph is displayed. This is performed by showing a map, where the user can navigate on it, by zooming and moving to every part of the map.
- Filtering the data according to the different properties of the data shown on the map.
- Visualization of the relationships between the different objects being displayed.
- Getting additional information about a displayed object.
- Visualize and analyse the variation of the data over the temporal dimension, using different techniques.

Figure 9 shows one of the possibilities to visualize the variation of data according to its temporal dimension.

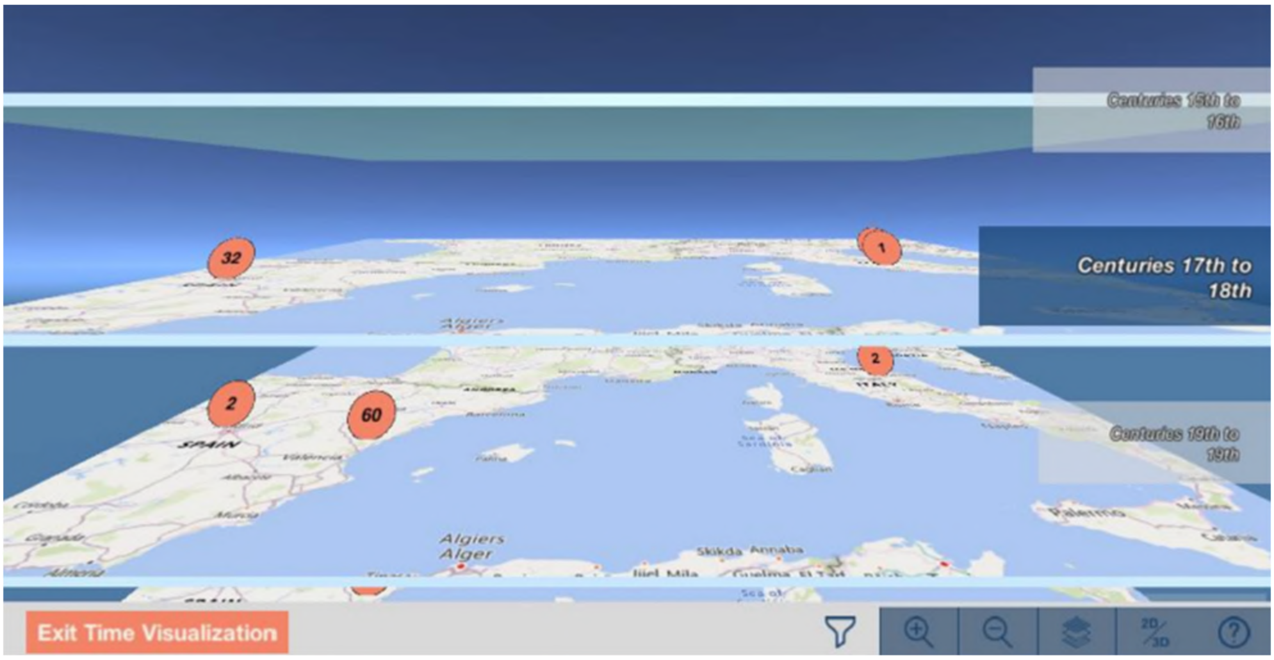


Figure 9. Visualization of data variation per century using a layered time visualization technique.

5.1.2. Integration

The STMaps software module is integrated into the ADASilk web application like a WebGL plugin.

ADASilk provides lists of objects (silk fabrics or other objects) that match the filters selected by the users. These search results are typically displayed as a list of images with pagination as in most common search engines. To show these results inside the Spatio-Temporal visualization component, the user can click on the “Show on the map” button.

The integration of STMaps with ADASilk is detailed in D6.5. ADASilk sends a paginated JSON blob object to STMaps with information about the objects generated as a result in the last query of ADASilk.

The structure of the JSON Blob is:

```
{
"id":"http://data.silknow.org/object/aa4788c0-f59f-3503-9621-
86f64ce43584",
   "identifier":"6611",
   "production":{
      "id":"",
      "location":{
         "country":"Spain",
         "id":"https://sws.geonames.org/2510769/",
         "label":"Spain",
         "lat":"40",
```

```
            "long":"-4"
         },
         "material":[
            "Silk",
            "Metal thread"
         ],
         "technique":[
            "Brocading weft",
            "Damask",
            "Patterned fabric"
         ],
         "time":"eighteenth century (dates CE)"
      }
   }
```

Figure 10 shows in a schematic form with the communication between ADASilk and STMaps. The result set corresponding to a set of filters selected on the ADASilk exploratory search engine is passed to the STMaps module. More precisely, some metadata that are useful for the STMaps module such as the identifier of the object, its label, the geo coordinates of the production location, the production time as well as the material and technique used, are wrapped into a JSON enveloped and sent to the component that loads the data and enables additional interactions.

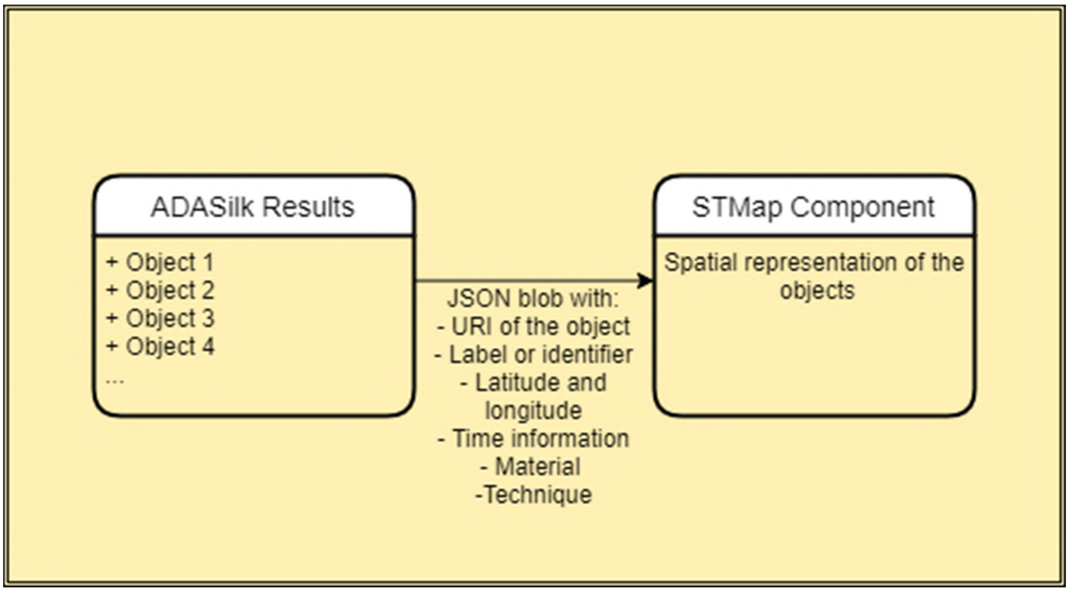


Figure 10. The communication process between ADASilk and the STMaps.

5.1.3. Evaluation

The stress testing of the STMaps tool was detailed in deliverable D5.7, which also describes a functionality evaluation. In that deliverable, we describe how the STMaps tool was tested with 4 different datasets, from 300 to 30,000 objects, using different hardware configurations, in order to evaluate the performance of the tool. The parameters measured in that stress test are:

- Memory usage: The memory required for loading the component.

- Frame-rate: This parameter allows to determine the user experience level. If the frame-rate is very low, it is very difficult to manage the navigation and interaction with the STMaps component.
- Time required for initial process. The time required for launching the component and visualize a map with data, allowing the user to navigate.

The evaluation of STMaps tool concludes that the component could be executed in a medium computer without problems, but also identify processes that could be improved in order to get a better user experience:

- The generation of markers.
- The generation of structures for line-based visualization of relationships.
- The JSON deserialization process.

These processes have been already improved and integrated in the STMaps v1.2, which can be found at the GitHub repository, in https://github.com/silknow/spatio-temporal-map/releases/tag/1.2.

5.2. Virtual Loom

5.2.1. Overview

Virtual Loom is an application that deals with the 3D virtual representation of historical silk fabrics at the yarn level. Silk fabrics have specific characteristics, as they are nearly flat objects and very fragile. The documentation of their visual appearance has been traditionally done by means of imaging devices (e.g., RGB cameras, digital microscopes, etc.). However, within these devices, only the surface of the objects is documented, so the complex internal structure composed of a variety of yarns and their interlaces, remains undiscovered. To deal with this, in Virtual Loom we produce 3D models of silk fabrics at the yarn level, with the minimum information of an image as input data. The implementation of Virtual Loom has been detailed in D5.4, its relationship with 3D printing is described in D5.5, and some case studies are presented in [5].

5.2.2. Integration

The Virtual Loom component is integrated into the ADASilk web application as a WebGL component. This component is rendered in the same web application and can communicate with ADASilk via Javascript calls.

For those objects in ADASilk that have an image representation, the web application provides an interface button to open that object into the Virtual Loom. When this button is pressed, a JSON blob is sent from the exploratory search engine to the webgl component. The JSON blob has the following structure:


```
{
  "language": "EN",
  "imgUri": "http://silknow.org/silknow/media/met-
museum/223472_0.jpg",
```

```
  "dimension": {
    "x": 12,
    "y": 8
  },
  "technique": ["Velvet"],
  "weaving": "Plain",
  "backgroundColor": {
    "r": 0.7075471878051758,
    "g": 0.2302865833044052,
    "b": 0.2302865833044052,
    "a": 0
  },
  "materials": ["Silk"]
}
```


This information is then loaded into the Virtual Loom component that loads the image to produce the 3D representation of the textile. The integration of Virtual Loom with ADASilk is detailed in D6.5.

5.2.3. Evaluation

The stress testing of the Virtual Loom component was detailed in D5.7, which also describes a functionality evaluation. In that deliverable, we present how the Virtual Loom tool has been tested for 2 different platforms (Windows standalone and WebGL) using 5 different hardware configurations, in order to evaluate the performance of the tool.

The tests were performed with multiple scenarios describing different techniques and textiles resolutions The parameters measured in this stress test are:

- Mesh Memory usage. The memory required by the 3D meshes to represent the textile.
- Mesh Memory usage. The memory required by all the textures in the component.
- Weaving Time: This parameter measures the time used to generate the geometry of the 3D model and assign materials without taking into account the drawing time.
- Number of vertices: This parameter is used to determine the resolution of the 3D representation of the textile.
- Frame-rate. This value indicates how smoothly the system moves the displayed 3D object. We consider that values above 30 FPS are required for a good user experience.

The stress testing of the Virtual Loom has been satisfactory in most cases. We analyzed different techniques on different PCs and we can say that Virtual Loom is usable in almost all recent PCs. We have only had performance problems on a low-profile PC and high-quality 3D models.

6. MAKING USE OF THE MULTILINGUAL THESAURUS

6.1. Searching for terms

The multilingual SILKNOW thesaurus enables users to search through more than 600 specialized silk terms in 4 languages. It has been designed especially for researchers and museum professionals. However, as it also defines non-specialized terms, it can be used to discover silk heritage. In the following lines, we present a scenario in which a researcher who is not a silk specialist uses the thesaurus. It is worth mentioning that anyone can replicate this scenario.

A museum specialist has to catalogue a piece of fabric (Figure 11). As she is not a specialist in this field, she decides to look up the term that appears in 19th century documentation in Italian. She needs information not only to understand what it is, but also how to catalogue it.

Figure 11. Piece of fabric that the museum specialist needs to catalogue.

She searches for that term in Italian that is not her native language. To do so, she can either use a free text search field or click on all concepts sorted alphabetically. She types "brocatto": three options appear (Figure 12, a). Next, she goes to alphabetical order and decides to click on the first one, broccato (tecnica) (Figure 12, b).

b

Figure 12. (a) Alphabetical sorting of the terms defined in the thesaurus; (b) Search box with auto completion: the user is typing 'broccato' in Italian and the system is suggesting 3 possible terms that match this search.

Thanks to the illustrating image, she notices that it was what she was looking for. Then she selects "Spanish" and then she can read the definition. However, the definition specifies that it is a generic concept, so she notices that there are related terms, and she thinks that espolín is the term that she was looking for as it is something that she has heard before. After reading, she selects the concept that the definition refers to, brochar. Once she can confirm that it is the fabric she was looking for, she clicks in the hierarchy in order to understand how she can better catalogue it. Then she looks at the bibliography to better investigate, as she understands English and French, she changes the language to discover further literature on this fabric.

Finally, she realizes that it is linked to Wikidata. She clicks and discovers that there is more literature and it is linked to Europeana, which will allow her to find other collections similar to her own.

This scenario shows how a search session in the SILKNOW's thesaurus can allow a researcher, who is not a silk specialist, to be more accurate in her cataloguing because museums are not always specialists in what they preserve.

6.2. Integration the thesaurus in third-party applications

In addition to the deployment of an optimized version of the SKOSMOS software that enables to search and browse the SILKNOW thesaurus content, we further developed the SILKNOW RESTful API to provide programmatic access to the thesaurus.

This API enabling to integrate the thesaurus into a third-party application is documented and deployed at https://grlc.eurecom.fr/api-git/silknow/api/. More precisely, we are providing 3 possible routes:

- `GET /concept-search` takes as input a string, and returns a list of concepts that matches the string. It is in effect a search API for concepts defined in the thesaurus. The string matching is performed on the preferred label as well as all alternative labels (i.e. synonyms) of concepts, in all languages.
- `GET /concept-detail` takes as input the URL identifying a concept in the thesaurus and return all information known about this concept including its labels and synonyms in the 4 languages (English, Spanish, French, Italian), its narrower, broader and related concepts, its relationship with external sources such as the Getty AAT and Wikidata, its definition as well bibliographical sources.
- `GET /concept-children` takes again as input the URL identifying a concept in the thesaurus and return a list of identifiers (URL) of the narrower concepts.

The source code of this API is open sourced at https://github.com/silknow/api.

7. REVISITING THE PILOT SCENARIOS FOR THE VARIOUS TARGET AUDIENCES

In D2.3 (pages 10-11), we described three user personas, namely Anna F. (a textile museum creator), John B. (a museum visitor) and Louisa G. (the CEO of a silk textile company) in order to guide the design of the overall user interface of ADASilk. In D2.4 (pages 9-29), we described numerous user scenarios for each of the identified targeted audiences.

For this deliverable (i.e., D6.7) we have selected one scenario for each target audience (cultural heritage and leisure; research and education sector; media; creative industries; and tourism) and we have produced short videos showing how those scenarios can be realized using the various tools developed by the project and integrated in ADASilk. This constitutes the final multimedia documentation of the system, a total of five videos which are available at the project's Youtube Channel.

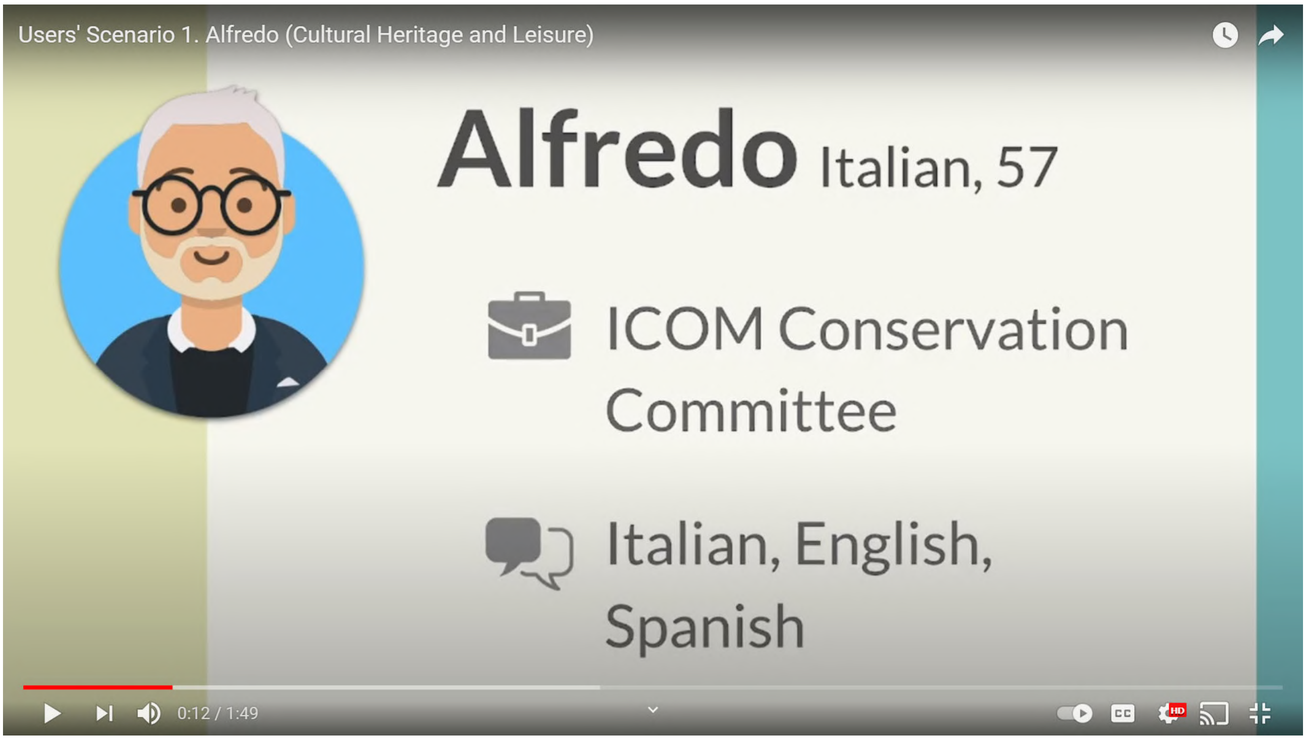
In the following subsections, we provide a summary of the different scenarios, as described in deliverable D2.3, and add a screenshot and the URL for the related videos, produced as part of deliverable D6.7.

7.1. Scenario 1: Cultural Heritage and Leisure

Cultural Heritage and Leisure	
Alfredo, 57, native Italian speaker, works for the ICOM Conservation Committee, has a good level of English and a basic level of Spanish. He uses the Internet primarily for work.	
Personal key facts	57 years old, museum conservation, international organization
Scenario	
Search / Research interests	Alfredo must report on the state of conservation of some silk textiles in a museum. Some of the pieces had very rare decorative patterns and therefore it is very difficult to establish the conditions of the weaves and the original colour tones.
Search / Research interests related with ADASilk	Alfredo needs to compare the images in his possession with those of similar textile in various museums and EU collections.
Search / Research interests related with the Virtual Loom	Alfredo needs to have a high-resolution image of the motif as he needs to understand the weaves being used.
Search / Research interests related with the Thesaurus	He also needs to distinguish the depictions that appear to properly establish the chronology and, thus, to better catalogue them and conserve them.
Use of ADASilk	Alfredo carries out a few searches on ADASilk, comparing the images in his possession with those retrieved by the system. Thanks to the technical metadata and to the high-resolution

	versions of the images, he is able to catalogue the decorative patterns of textiles he had examined and can make an accurate report on the preserving conditions. He clicks on “depictions”, and chooses “floral motif”. In the results page, he filters by “palm”; technique = “damask” and production time = “18th century”. He finally selects the museum record https://ada.silknow.org/object/085680cc-aa84-394d-9e62-a9f18cd5dad0
Use of Virtual Loom	He launches the Virtual Loom from the image illustrating this record in ADASilk and weaves it in damask style.
Use of the Thesaurus	At the same time, Alfredo goes to the thesaurus, and searches for *palm* as this is the motif he has identified after the comparison among several images. He uses the thesaurus in English as he is using ADASilk in that language and the museum is English speaking and so his catalogue. Then, he clicks in Italian to better understand the definition. He confirms that this type of pattern is widely used in velvets, which is the fabric he is studying.
Expectation and sources	Alfredo would like to find in a short time the decorative patterns that interest him for his report.

The video illustrating this scenario is available at https://www.youtube.com/watch?v=IlFKbci_dwA



7.2. Scenario 2: Research and Education Sector

Research and Education Sector	
Vincenzo is 38 years old. He is an Italian native speaker, and he is an English teacher in a vocational school. Because of his job, he has a very good level of English. He uses Internet daily for research related to his work and to keep himself informed through popular online newspapers.	
Personal key facts	38 years old, vocational school teacher, he uses Internet daily for didactic research purposes.
Scenario	
Search / Research interests	Vincenzo is a teacher in a secondary and vocational school. He teaches English in some fashion courses. He needs to tap into a technical vocabulary and some material to organize class assignments.
Search / Research Interests related with ADASilk	Vincenzo needs to show images of the words he will teach to his students to make his course more attractive.
Search / Research interests related with the Virtual Loom	To make it even more attractive to his students, he needs to show how digital technologies can help students to enhance their creativity.
Search / Research interests related with the Thesaurus	Vincenzo uses the material available via ADASilk, such as the multilingual thesaurus, to learn about the history of silk and uses what this material to create resources for his fashion students. Later, he will create homework for his students. He expects the vocabulary to be easily understandable and attractive, so he focuses on objects, especially costumes.
Use of ADASilk	He goes to objects and then he filters by “costume attire”. He selects the most attractive one such as https://ada.silknow.org/object/644c03c9-c7d4-3f9e-8e1f-876d4ceeb399
Use of Virtual Loom	He uploads an image and plays with it.
Use of the Thesaurus	As he wants costumes in English, he switch to this language. Then, he types in the search textbox “attire” and then he goes to the hierarchy to select those terms more appropriate to his classes. He shows the term in both English and Italian.
Expectation and sources	Vincenzo's reference points are ADASilk: he searches material for educational purposes for the subject he teaches, in order to allow his students a more complete specialized vocabulary. Moreover, Vincenzo looks for specialized sites to suggest to his students, so he expects that ADASilk he is using is always updated with new materials and good links.

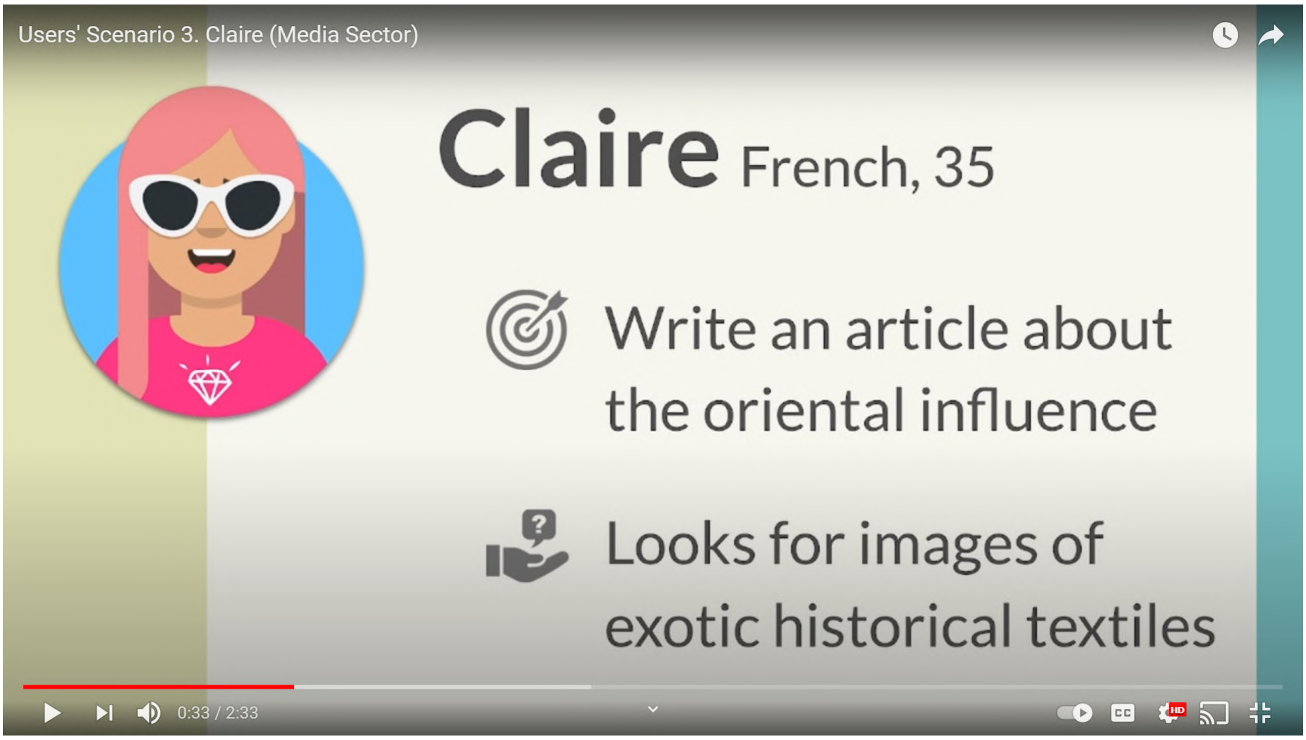
The video illustrating this scenario is available at https://www.youtube.com/watch?v=mLal-TOfUsc

7.3. Scenario 3: Media

Media Sector	
Claire is 35 years old. She is originally French. She has moved to Tokyo to work for 2 years as a fashion journalist. She is particularly interested in street fashion. She is a regular contributor for an international fashion magazine and in a number of social media outlets.	
Personal key facts	35 years old, fashion journalist, international press, social media
Scenario	
Search / Research interests	Claire has been asked to write an article for the magazine she works for. The subject of the article focuses on street fashion and must highlight how much traditional oriental fashion has influenced taste internationally throughout history.
Search / Research interests related with the AdaSilk	Claire looks for images of historical textiles inspired by the test of exoticism, in order to compare them to contemporary fashion.
Search / Research interests related with the Virtual Loom	She wants to accompany her article with a woven fabric.

Search / Research interests related with the Thesaurus	As Claire needs to write an article, she needs to know the specific vocabulary of those styles. In order to do so, she needs to be as accurate as possible so she looks in the thesaurus.
Use of the AdaSilk	Claire changes the language to French. Next, she looks for "chinoiserie" using the free text search facility. She clicks on the STMaps module as she will use it to highlight the international relations and exchanges of silk motifs. She explores in particular the object https://ada.silknow.org/object/4b96d304-a01d-3fc2-9921-d3add96322c6
Use of the Virtual Loom	She uploads the image she has selected to the Virtual Loom as she wants to show how an oriental motif can be woven with a European technique.
Use of the Thesaurus	Claire uses the thesaurus in French and goes to the alphabetical search looking for styles. She chooses "style oriental" and she notices that there are 2 related terms that are related to her search, "japonisme" and "chinoiserie". She clicks on both, and changes the language to English as her article is in that language.
Expectation and sources	Claire expects to find shareable images on social media that not only show the decorative patterns, but also entire silk items. In her article, she will underline how the diffusion and exchange of images on the Web has broken geographical barriers, allowing fashion companies to inspire new fashion collections in historical textiles and images.

The video illustrating this scenario is available at https://www.youtube.com/watch?v=DX1GnH-ioYY.

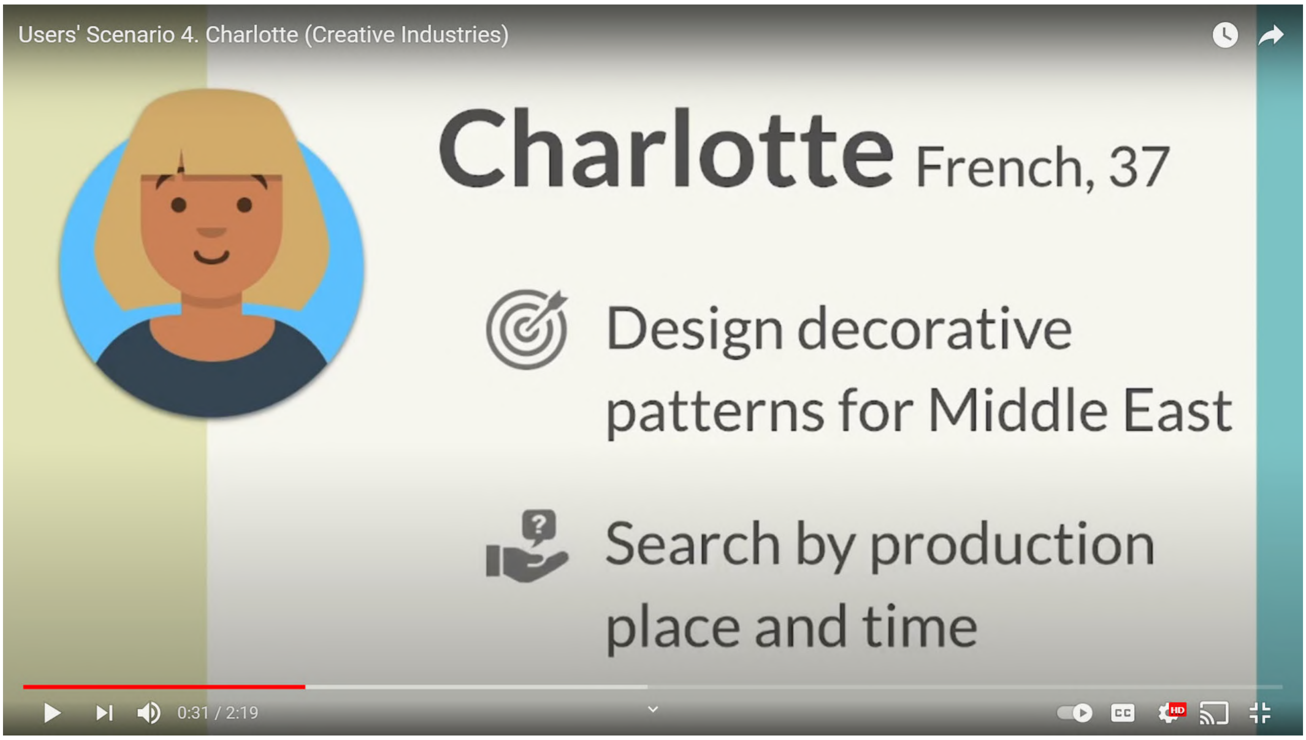


7.4. Scenario 4: Creative Industries

Creative industries	
Charlotte is 37 years old. She speaks French, English and Italian. She works as a textile designer for a silk company in Como.	
Personal key facts	37 years old, textile designer, she speaks French and English.
Scenario	
General research interests	Charlotte must design for her company some decorative patterns for silk textiles intended for the Middle East market. Fabrics must be inspired by the European tradition. Under the guidance of Franz, the CEO of her company, she logged in to ADASilk to get inspiration for the textiles to be designed.
Search / Research interests related with ADASilk	As Charlotte needs to get inspired, she needs a repository where she can filter by production place and production time.
Search / Research interests related with the Virtual Loom	Charlotte needs to look at the fabric at yarn level so she can get an idea of how complicated her patterns will be woven.
Search / Research interests related with the Thesaurus	As the fabrics will be sold in the Eastern Market, she needs to sell them with proper names, but she usually calls fabrics and depictions by the names given in the company and not the correct ones, so she needs a place to look for the correct terms.

Use of ADASilk	She goes to ADASilk and selects “France” as the “production place”; She then selects “18th century” for the “production time” as it is the most famous century for French fabrics. She browses through the result set to find inspiration and she finally selects https://ada.silknow.org/object/3eb5ea35-f66a-3402-aea1-d69d991702ae
Use of Virtual Loom	Charlotte wants to get inspired by fabrics widely used in Europe. Thanks to ADASilk she found a fabric with a pattern that she loved with the name of the designer. She selects it and activates the Virtual Loom to see how some motifs appear depending on the technique.
Use of the Thesaurus	The designer who made the fabric she liked was Jean Revel. She uses the thesaurus to know more about him, as it is called Style Jean Revel. As she speaks French, Italian and English, she looks at the references so she can go further in her research thanks to the bibliographic citation. https://skosmos.silknow.org/thesaurus/en/page/688
Expectation and sources	Charlotte wants to select only textiles with decorative patterns that were widely used in Europe. She expects to find the decorative patterns as vector drawings and to be able also to download 3D renders and give her an idea of the finished result.

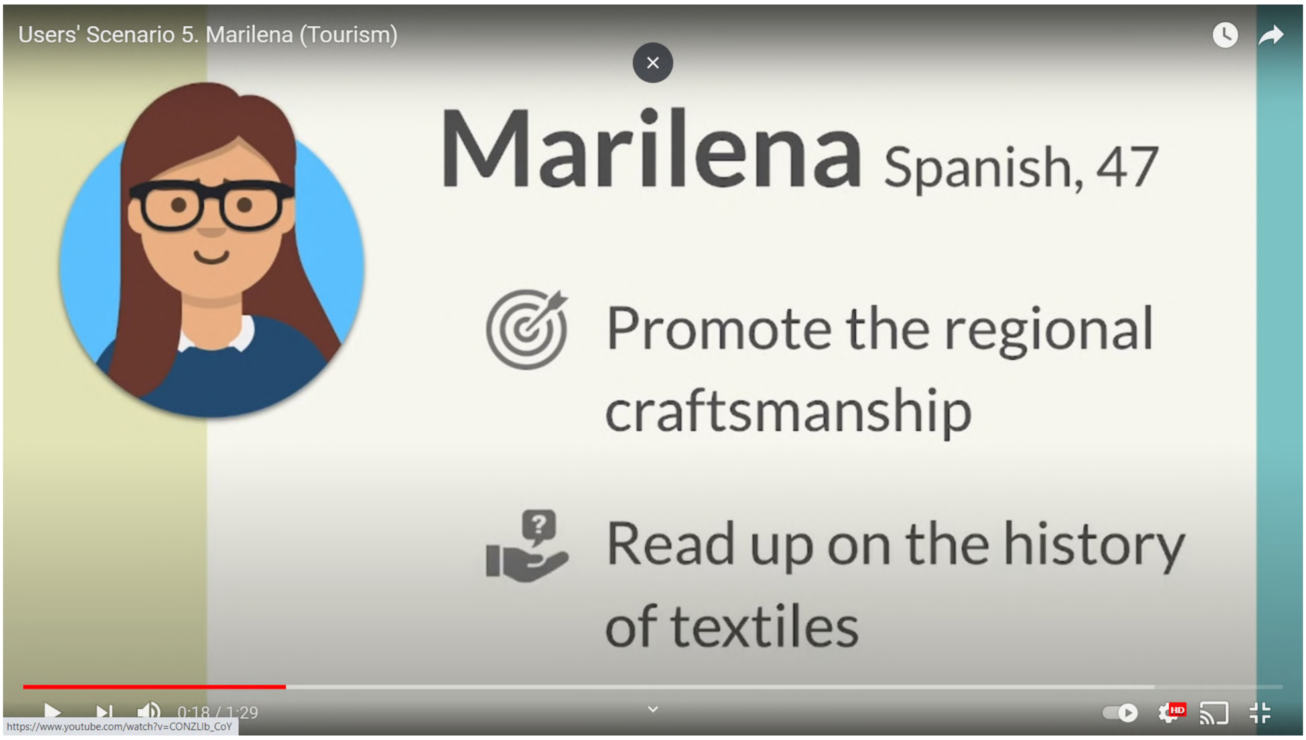
The video illustrating this scenario is available at https://www.youtube.com/watch?v=bMa6to5_F4M.



7.5. Scenario 5: Tourism

Tourism	
Marinela, Spanish, 47, president of a regional craft association, good level of English, uses social media to promote the activities of the association.	
Personal key facts	47 years old, president of a regional craft association, speaks English and Spanish.
Scenario	
General research interests	Marinela's association has to promote the textile craftsmanship of her region and to do it she needs to read up the history of textiles.
Use	Thanks to the historical data and technical data available on ADASilk, Marinela is able to give value to local craftsmanship, which is important to the production of silk textiles. She organizes events to enhance the tools of weavers and focuses on those artisans that still produce traditional clothes for local festivals, not losing historical heritage.
Search / Research interests related with ADASilk	Marinela needs to find patterns and techniques similar to those she weaves as she wants to highlight the craftsmanship of her region.
Search / Research interests related with Virtual Loom	As she wants to highlight the craftsmanship of her region and organizes events to enhance weavers, she wants to give free souvenirs that are related to her region.
Search / Research interests related with Thesaurus	Marinela wants to know how espolin is called in other languages.
Use of ADASilk	Marinela looks for "espolin" using the free text search functionality and filters "Valencia" by production place and finally selects https://ada.silknow.org/object/4a87ca38-785a-3084-98b6-5b64b9d97a3a
Use of the Virtual Loom	Marinela uploads an image of her own production and weaves it as a damask and she prints it on her 3D printer to give as a souvenir and especially to give it in the tourism office of her region to attract people.
Use of the Thesaurus	Marinela looks for "espolin" in the thesaurus and changes the language to English.
Expectation and sources	Marinela expects to find data on the history of places where silk textile production has developed the most. She will also try to build partnerships for the development of local textiles, traditions, using the platform's contacts.

The video illustrating this scenario is available at https://www.youtube.com/watch?v=IRiW4aAWnus.


Users' Scenario 5. Marilena (Tourism)
Marilena Spanish, 47
Promote the regional craftsmanship
Read up on the history of textiles
0:18 / 1:29

8. CONCLUSION AND FUTURE WORK

In this deliverable, we have provided the final documentation of the SILKNOW integrated system which takes the form of an exploratory search engine named ADASilk.

ADASilk is developed as open source under the Apache 2 license at https://github.com/silknow/adasilk as a skinned engine obtained via a configuration file of a more general exploratory search engine for any knowledge graph developed at https://github.com/D2KLab/explorer. ADASilk is available as a Docker image which facilitates its deployment in one click. It is permanently available at https://ada.silknow.org/. ADASilk integrates a number of key results from SILKNOW namely:

- The Image-Based Retrieval component developed at https://github.com/silknow/image-retrieval and https://github.com/silknow/image-retrieval-server;
- The predictions for additional metadata about the production of the objects generated by a text classification component (https://github.com/silknow/text-classification) and an image classification component (https://github.com/silknow/image-classification);
- The Virtual Loom developed at https://github.com/silknow/virtual-loom;
- The Spatio-Temporal Maps component developed at https://github.com/silknow/spatio-temporal-map.

This deliverable is multimedia rich thanks to the production of short video tutorials that exemplify how to perform specific scenarios that were previously defined for each targeted audiences of SILKNOW and using original personas. These videos are available at https://www.youtube.com/channel/UCTJJT6jhtJwMRprw808Tw9w/videos.

REFERENCES

1. W3C WCAG guidelines. https://www.w3.org/WAI/standards-guidelines/wcag/
2. W3C WAI-ARIA Initiative https://www.w3.org/TR/wai-aria-1.2/.
3. Amin, A., Hildebrand, M., van Ossenbruggen, J., & Hardman, L. (2010). Designing a thesaurus-based comparison search interface for linked cultural heritage sources. *In 15th international conference on Intelligent user interfaces*, (*IUI)*, 249-258.
4. Lisena P., Meroño-Peñuela A., Kuhn T. & Troncy R. (2019). Easy Web API Development with SPARQL Transformer. In *18th International Semantic Web Conference (ISWC)*, Auckland, New Zealand, October 26-30, 2019.
5. Gaitán, M., Portalés, C., Sevilla, J., Alba, E.: Applying Axial Symmetries to Historical Silk Fabrics: SILKNOW's Virtual Loom. Symmetry. 12, 742 (2020). https://doi.org/10.3390/sym12050742
6. PROV Ontology, https://www.w3.org/TR/prov-dm/
7. Visualization Ontology (VISO), http://purl.org/viso/
8. ADASilk: https://ada.silknow.org/
9. The Silk Heritage Thesaurus: https://skosmos.silknow.org/thesaurus/en/
10. Image Based Retrieval module: https://github.com/silknow/image-retrieval

Vniversitat de València

Università degli Studi di Palermo

Leibniz Universität Hannover

MONKEYFAB

Institut "Jožef Stefan" Ljubljana, Slovenija

Instituto Cervantes

cnrs

EURECOM

GARÍN 1820