Digital Library Perspectives

119

A dynamic system for the automatic classification of complex resources in digital libraries: design and preliminary evaluation

Nicola Barbuti
Dipartimento di Ricerca e Innovazione Umanistica DIRIUM, Universita degli Studi di Bari Aldo Moro, Bari, Italy, and

Tommaso Caldarola
D.A.BI.MUS. S.r.l, Bari, Italy

Received 10 May 2025
Revised 18 August 2025
Accepted 7 October 2025

Abstract

Purpose – Digital image automatic classification has become a critical field in machine learning. With the exponential increase in the availability of digital data and the growing complexity of applications, the need to develop accurate and efficient data automatic classification models has become urgent across multiple sectors, as they contribute to enhancing operational efficiency. This scenario underscores the necessity of exploring and developing new approaches capable of overcoming these challenges, further improving the accuracy and efficiency of classification techniques. This paper aims to present the ongoing research for development and testing an automatic image classification model for digital libraries, based on complex neural networks (CNNs).

Design/methodology/approach – Despite the significant advancements achieved with the advent of deep learning approaches, the challenges of automatic classification of digital resources remain in terms of generalization, model interpretability and reducing dependence on large training data sets. After outlining the state-of-the-art digital resources' automatic classification, the paper describes the model research and design and the pilot of implemented application workflow. Finally, preliminary research results are assessed, considering that experimentation is still ongoing to evaluate the potential integration of AI tools to enhance model performance.

Findings – The research addresses the challenge of developing a model effective for classifying digital resources referring to the huge and various contexts of digital libraries including resources representative of manuscript, early printed and modern books. The process to develop the first pilot of the automatic classification system the researchers designed and developed has been clearly outlined. The workflow and the generation of a specific CNN for classifying digital libraries are detailed by examples, figures and tables that show each step of the process describing the methods, techniques and technologies used.

Research limitations/implications – There are no research limitations/implications.

Practical implications – There are no practical implications.

Social implications – There are no social implications.

Originality/value – The experimentation results provide an encouraging overall picture of the developed models' performance, highlighting their potential for analyzing and classifying the structures of the considered materials. An extensive series of tests were conducted on a diverse data set to assess their effectiveness, accompanied by a rigorous validation procedure on an even larger sample. The pilot model shows remarkable

Digital Library Perspectives
Vol. 42 No. 1, 2026
pp. 119-140
Emerald Publishing Limited
2059-5816
DOI 10.1108/DLP-05-2025-0063

performance in accuracy, achieving an average correct classification rate of 78% for the three analyzed types and over the full validation data set. The learning curve displayed good convergence, suggesting that further optimizations could improve the overall precision, particularly fine-tuning the hyperparameters regulating the training process and refining the machine learning model topology. Such improvements would increase accuracy and reduce uncertainty for classes that showed greater variability.

Keywords Automatic classification, Convolutional neural networks, CNNs, Digital library, Digitization, Digital heritage

Paper type Research paper

1. State-of-the-art

The digitization of cultural heritage represents one of the most complex ongoing challenges of the 21st century, situated at the intersection of technological innovation, cultural mediation and universal accessibility. Due to the acceleration of digital transformation processes, sharply intensified after the SARS-CoV-2 pandemic (Barbuti, 2022; Barbuti and De Bari, 2024) and the widespread availability of large language models (Ciotti, 2023), institutions such as the Vatican Apostolic Library, the British Library and Google [1] have invested billions of euros for digitization projects, generating collections exceeding 100 million pages.

In this rapidly changing scenario, mere analog-to-digital conversion is no longer sufficient (Barbuti and De Bari, 2024). Among the various techniques that have become indispensable in the digitization processes of cultural artifacts, especially of libraries and archives, automatic classification of digital resources has become essential for transforming raw data into structured knowledge, enabling advanced functionalities such as semantic search, stylistic analysis and the digital reconstruction of fragmented works (Neudecker, 2022).

However, although research into automatic classification techniques has been one of the most interesting areas within the digital humanities – and beyond – for the past 20 years (Haralick *et al.*, 1973; Alaei *et al.*, 2009; Krizhevsky *et al.*, 2012; He *et al.*, 2016; Mohammed *et al.*, 2021; Reynolds *et al.*, 2022; Vidal-Gorène *et al.*, 2024), it has found limited application in the generation, metadata creation and management processes of digital resources related to bibliographic artifacts (ancient books, manuscripts, periodicals, etc.). Images of the various parts comprising the structure of a volume, especially ancient one, are rarely online labeled with the "nomenclature" of each part as per ICCU guidelines [2]. This statement is because, to date, the only method to enrich structural information in the metadata managing each image consists of inserting the nomenclature directly of the represented part into the filename.

This option is rarely adopted because it involves a significant amount of manual work. Users must enter the nomenclatures into filenames either before scanning each part, so the information can be retrieved later from the image metadata, or after the scanning cycle is completed, renaming all images using dedicated tools. Both solutions have significant drawbacks, as modifying filenames multiple times during scanning considerably slows down the process and increases the risk of errors that negatively impact subsequent digitization phases; indeed, automatically assigning different nomenclatures to sets of images representing various parts of a volume's structure requires renaming files differently for each book.

The methodological and technical challenges to address for achieving effective automatic classification of digitized library artifacts are multiple and stratified due to the complex and variable structure of the represented artifacts, especially ancient ones and the formal

characteristics of the digital objects' layer. The intrinsic and extrinsic heterogeneity of physical originals requires a dynamic and flexible approach to the digitization and postproduction of digital resources. Manuscripts, for example, show peculiar features, such as supports with irregular textures, inks subject to various types of injuries and marginal annotations whose graphic style can vary even within the same work. Ancient, printed books (16th–19th centuries) introduce additional complexities, such as the transition from woodblock printing to movable type, with variations in typographic resolution and watermark usage. Although industrially produced, modern books still present challenges due to the overlap of graphic elements, diagrams and photographs that confuse algorithms relying on traditional features.

Digitization processes of these artifacts are often conducted in uncontrolled contexts and without following established guidelines and best practices (Barbuti, 2022), using heterogeneous devices (flatbed scanners, DSLR cameras, smartphones) that introduce color distortions, reflections and lighting variations.

Automatic classification of digital image based on traditional approaches has been the preferred methodology for over a decade in several research areas, especially in medicine and hard sciences. Nevertheless, its use on the layers of bibliographic artifacts has been scarce, and results have not always met expectations (Kumbhar, 2012; Sebastian *et al.*, 2012; He *et al.*, 2016). The recent advent of deep learning has fostered the design and experimentation of more promising models, though they mostly remain at the pilot stage (Swain and Ballard, 1991; Ojala *et al.*, 2002; Silla and Freitas, 2011; Hsu *et al.*, 2012; Kumbhar, 2012; Sandler *et al.*, 2018; Shanmugamani, 2018; Schomaker, 2019; Ćosović and Janković, 2020; Aouinti *et al.*, 2022). Convolutional neural networks (CNNs) have demonstrated superior capabilities in extracting hierarchical features from high-resolution images, recognizing patterns unreadable by the human mind, such as microcracks in paper or variations in ink density (Joachims, 1998; Lowe, 2004; Peng *et al.*, 2005; LeCun *et al.*, 2015; Goodfellow *et al.*, 2016; Ciotti, 2023; Milleville *et al.*, 2023; Liu *et al.*, 2024). However, these models require large and balanced data sets for effective algorithm learning, a condition rarely met in the cultural heritage domain, where collections are often fragmented or skewed toward overrepresented classes (e.g. printed books vs manuscripts). The lack of standardization of digitization protocols, despite the existence of consolidated references, increases the problem, creating misalignments between training data and real-world scenarios (Barbuti, 2022).

Further limitation is model interpretability. In academic and conservation contexts, it is not enough for an algorithm to correctly classify an image; understanding the basis for the decision is crucial. For instance, an Arabic manuscript might be misclassified due to similar marginal decorations, a mistake that a human expert would avoid by analyzing the handwriting's morphology or the material support (Alaei *et al.*, 2009). Although CNNs are powerful, they operate as "black boxes," limiting end-user trust and hindering integration into critical workflows.

This scenario motivates the need for hybrid systems that combine the power of CNNs with domain-specific knowledge. For example, integrating structured metadata (creation date, geographic provenance, etc.) or using semantic ontologies can improve model robustness. Projects like Europeana [3] and International Image Interoperability Framework [4] are laying the groundwork for such integrations by promoting open standards and interoperability between repositories.

This paper aims to present intermediate results of the study and development of a CNN-based image classification pilot that specifically addresses some of the above depicted challenges, while providing an in-depth evaluation of its performance compared to existing methods.

The ongoing research is situated within a context where algorithmic innovation must engage with multidisciplinary humanistic needs, balancing computational precision,

sustainability and respect for cultural diversity in the digital realm. From this perspective, automatic classification is no longer merely a technical exercise. However, it has become a fundamental requirement to mediate between past and future, as it represents the transition from the tangible matter of books and the nontangible matter of the digital.

2. Automatic classification between traditional approaches and deep learning models

2.1 Traditional approaches

Over the past decade, traditional approaches have dominated the automatic classification of digital objects. These are based on a "handcrafted" paradigm, whose methodological core involves manual feature extraction and selection of machine learning algorithms (Peng *et al.*, 2005; Krizhevsky *et al.*, 2012; Reynolds *et al.*, 2022). Although less powerful and effective than recent CNN-based methods, traditional approaches offer advantages in terms of interpretability and lower computational requirements, making them viable in contexts with limited data or outdated hardware (Shanmugamani, 2018; Schomaker, 2019).

The main phases and key techniques of the traditional approach are outlined below.

2.1.1 Phase 1. Feature extraction. In this phase, relevant features are manually or automatically extracted from images. Traditional features mainly aim to encode distinctive visual aspects through mathematical descriptors and may include information that is considered relevant for classification. Some of the main features useful for the process are:

- *texture:* Algorithms based on co-occurrence matrices [gray-level co-occurrence matrix (GLCM)] quantify textural patterns by analyzing the spatial distribution of pixel intensities (Silla and Freitas, 2011). Haralick's 14 descriptors (Haralick *et al.*, 1973), including contrast, energy and homogeneity, have been widely used to distinguish materials such as parchment and paper (Alaei *et al.*, 2009);
- *color:* Standardized RGB or HSV histograms (Vidal-Gorène *et al.*, 2024) capture color distribution, useful for identifying specific inks or pigments (e.g. azurite in medieval manuscripts);
- *shape and structure:* Descriptors like scale-invariant feature transform (SIFT) (Mohammed *et al.*, 2021) and histogram of oriented gradients (HOG) detect key points invariant to rotation and scale, suitable for recognizing typefaces or decorations;
- *Local patterns:* Techniques such as local binary patterns (LBP) map microstructures by binary comparing neighboring pixels, they are effective for analyzing watermarks or paper textures (Peng *et al.*, 2005).

2.1.2 Phase 2. Feature selection. Sometimes, reducing the dimensionality of extracted features is needed. Feature selection can be done to keep only the most informative attributes and reduce noise in the data. To reduce dimensionality and improve efficiency, algorithms such as principal component analysis or minimum redundancy maximum relevance identify the most discriminative features (Reynolds *et al.*, 2022).

2.1.3 Phase 3. Classifier training. A classification algorithm is trained using the extracted features and class labels from the reference images. This step involves creating a mathematical model, which can distinguish between different classes of images based on the selected characteristics. Statistical models such as support vector machine (SVM), random forest or Naive Bayes are commonly used at this stage (Krizhevsky *et al.*, 2012). In particular, SVMs create optimal separating hyperplanes in feature space. They are effective in binary tasks (e.g. distinguishing manuscripts from printed books), though their performance diminishes in multiclass or nonlinearly separable scenarios.

2.1.4 Phase 4. Validation and optimization. The model is validated using a separate data set. Parameters are adjusted iteratively to optimize performance. The process can be iterated until satisfactory results are achieved.

2.1.5 Phase 5. Testing and evaluation. Finally, the trained model is tested on an independent data set to assess its performance in terms of accuracy, precision, recall and other classification metrics.

2.2 Deep learning models

The *deep learning models* for image classification have evolved rapidly, driven by architectural and methodological innovations that have redefined computational performance limits.

Using pretrained models on large data sets like ImageNet [5], followed by fine-tuning on specific data sets, has proven effective. This approach, known as "knowledge transfer," allows models to learn generic representations from large data sets and apply them to specific issues with fewer training samples.

Recurrent neural networks are preferred to address sequencing matters, such as in the case of image descriptions or timelines analysis.

Initially designed for textual data, vision transformers (ViT) have been adapted to process images divided into patches. Their ability to capture global relationships through attention mechanisms makes them ideal for analyzing marginal decorations or complex layers.

Generative adversarial networks (GANs) and diffusion models are revolutionizing the generation of synthetic data to address the scarcity of real-world examples, as their use enriches limited data sets, and improves final model accuracy (Joachims, 1998; Zhang and Yoshida, 2010). However, using GANs raises ethical issues, such as the risk of unintentionally altering historical elements, requiring cross-validation by humanist experts.

Anyway, CNNs remain the backbone of image classification, with architectures such as EfficientNet and ViT surpassing the performance of classical models such as ResNet and VGG [6]. For example, EfficientNet achieves 84.4% accuracy with 66% fewer parameters than ResNet using balanced scaling of depth, width and resolution to optimize efficiency. In the context of digital library assets, these architectures can be adapted to recognize complex patterns such as parchment texture or microfractures in ink.

Integrating CNNs with attention mechanisms or multimodal data (e.g. historical metadata) is an emerging hard challenge. Hybrid models combining EfficientNet with transformers allow to process images and textual descriptions simultaneously, achieving significantly higher scores than unimodal models. Moreover, using knowledge graphs to incorporate contextual information (e.g. geographic provenance) shows promise in reducing misclassifications caused by hybrid styles.

Nevertheless, transferring knowledge from pretrained models on generic data sets to specialized domains presents unique challenges in heritage contexts. Images of ancient manuscripts or printed artifacts differ significantly from natural ones in brightness, resolution and semantic context, leading to domain shift phenomena. To mitigate this issue, adaptive fine-tuning techniques have been proposed, such as using domain alignment layers to reduce feature distribution divergence (Tan and Le, 2019).

Despite progress, several factors still affect the automatic classification of digital libraries resources, making defining a single universal model challenging.

Among them:

- *annotated data needs:* few-shot learning and self-supervised learning approaches (e.g. SimCLR [7]) are being explored to reduce reliance on large data sets;

- *interpretability:* tools like local interpretable model-agnostic explanations (LIME [8]) have been adapted to clarify CNNs decisions in manuscript contexts, fostering collaboration with art historians; and
- *computational ethics:* model training always requires consensus frameworks and epistemic justice.

Considering the peculiarities of historical materials, such as physical degradation, stylistic variability and scarcely annotated data, it is evident that the application of these technologies to digital libraries assets requires targeted adaptation (Neudecker, 2022; Ciotti, 2023).

2.3 Applications and issues in the digital libraries

Experiments with traditional automatic classification approaches have been rare and have produced mixed results in the domain of digital libraries.

An emblematic case is the "e-Codices" project [9], where SVMs with SIFT features classified medieval manuscripts with an accuracy of 72%, sufficient for filtering works by period but insufficient for detecting subtle calligraphic styles.

The main limitations include:

- *sensitivity to acquisition variations:* images having nonuniform lighting or differing resolutions distort extracted features, requiring laborious preprocessing;
- *inability to handle structural complexity:* marginal decorations, overlapping annotations or physical damages (e.g. burns) introduce noise that confuses manual descriptors; and
- *limited scalability:* manual feature extraction becomes impractical for data sets larger than 10,000 images, requiring a lot of labor time.

Some studies have explored hybrid approaches combining traditional features with shallow neural networks to address these issues. For instance, in the "Digital Resources and Database of Manuscripts, Palaeography and Diplomatic (DigiPal)" project [10], LBP and HOG features were combined with a two-layer network to classify Anglo-Saxon scripts, improving accuracy to 78%. However, such methods still underperform compared to end-to-end CNNs, which exceed 85% accuracy for the same task.

Despite their relative obsolescence, traditional methods have requirements that still make them preferred in some applicability niches due to:

- *interpretability:* in academic contexts, SVM transparency is preferred to validate philological hypotheses;
- *limited resources:* institutions with small budgets use random forest for preliminary filtering activities; and
- *low-quality data:* for extremely noisy or low-resolution images, descriptors like GLCM are sometimes more robust than deep networks.

Thus, traditional approaches remain effective for relatively small data sets or when discriminative characteristics are well-defined and reliably extractable. Nevertheless, some researchers propose to rework traditional features into modern frameworks (Hsu *et al.*, 2012).

Anyway, today the scientific community broadly agrees that the future lies in integrating domain-specific knowledge into CNNs rather than reverting to obsolete paradigms. In recent years, deep learning-based approaches have proved remarkable success in overcoming many limitations of traditional methods, especially in scenarios requiring complex and hierarchical

representations crucial for image recognition (Roald *et al.*, 2024). However, at the state-of-the-art, while offering powerful tools, deep learning requires critical adaptation to the specificities of library holdings. Given the high variability in digitized artifact structures, particularly ancient ones, generating a single automatic classification model valid across all types remains a high-level scientific challenge, requiring a balance between technological innovation, philological rigor and cultural awareness.

3. Methodology

The still ongoing research aimed to address the abovementioned challenges by designing and developing a dynamic system based on CNNs that could operate with high accuracy, computational efficiency and adaptability on heterogeneous library assets. The focus was on three digital libraries categories: manuscripts, early (16th–early 19th centuries) and modern printed books (late 19th–21st centuries).

The adopted approach was grounded in the latest deep learning techniques based on CNNs (Liu *et al.*, 2024), aiming to leverage the distinctive features detectable from digitized images representing ancient and modern library assets.

During the pilot design and training process, TensorFlow [11], an end-to-end machine learning platform and the high-level API library Keras [12] were used alongside libraries for image manipulation and for mathematical operations on arrays.

The experimental methodology involved collecting a representative image data set, careful data preparation, arranging at least 20 examples of nomenclatures for each type of book part for model classification training and learning outcomes validation, and defining a neural network architecture tailored to capture the specificities of the target categories.

3.1 Data preparation

For preparing and training the system, sample image sets representing various structural parts of the three categories of digitized books were identified and selected for recognition and classification. Each part to recognize was associated with a specific class, consistent with the nomenclature outlined by the ICCU [13].

Building a balanced and diversified data set was crucial to reflect typical variations in digital library assets. Digital collections include resources that present both intrinsic differences such as materials (parchment, cotton paper, industrial paper), printing techniques and preservation states, and extrinsic differences arising from acquisition processes (e.g. lighting, color tone, sampling frequency) (Barbuti, 2022). To mitigate these factors, the data set was built using the digital collections generated for the EDIt Digital Library at the University of Bari Aldo Moro [14]. It initially included 60,000 images of ancient, printed volumes scanned with professional scanners at 600 or 400 ppi resolution.

Advanced data augmentation techniques were applied to address the risks due to the limited amount of data:

- *geometric transformations:* rotations (±15°), shear (0.2), zoom (10%) to simulate nonoptimal acquisition angles;
- *photometric changes:* adjustments in brightness (±30%), contrast and saturation to replicate varied lighting conditions; and
- *noise addition:* simulating dust, scratches and stains using Gaussian filters, reflecting the physical degradation of ancient works.

The data set was split, randomly allocating 80% for training and 20% for validation. Preprocessing involved resizing samples resolution from the original format to 224x224

pixels without preserving the aspect ratio, to reduce intraclass variance and improve convergence during training.

3.2 Normalization

In machine learning, *data normalization* is a critical step, as many algorithms are sensitive to variations in feature weight. Normalization facilitates the learning process by allowing the model to assign appropriate weights to each feature during training. Without it, features with larger numerical values could dominate the learning process, leading to biased outcomes and nonoptimal model performance. Therefore, for training purposes data normalization is a crucial preprocessing step to standardize the features so they have a common scale, and to ensure each of them contribute proportionally and effectively to the model's training.

Normalization techniques referenced included:

- *standardization:* subtracting the mean of the feature and dividing by its standard deviation. This ensures that the feature has a zero mean and a standard deviation of one.

  In python: standardized_feature = (feature – mean)/std_dev;
- *min-max normalization:* rescaling features to a specific range, e.g. 0–1.

  In python: normalized_feature = (feature – min_value)/(max_value – min_value);
- *L1 or L2 normalization:* normalizing features ensuring the sum of absolute values (L1) or squares (L2) equals 1.

  In python: L1_normalized_feature = feature/sum(abs(feature)); and

  L2_normalized_feature = feature/sqrt(sum(feature^2)).

"Batch Normalization" was adopted as process to design the model. This technique is used in neural network training to improve stability and accelerate convergence. In practice, Batch Normalization normalizes the output of each hidden layer (or layer input) based on the batch of data being processed. The aim is to make the layer's output more stable during training.

Given a mini batch of size "m" and a specific feature map at a given layer, the mathematical formula for batch normalization can be expressed as follows:

1. *Calculating batch mean and variance:*

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_B)^2$$

where μ_B is the batch mean and $\sigma^2{}_B$ is the batch variance. The mean (μ_B) and variance ($\sigma^2{}_B$) are computed over a mini-batch of m examples. This centers the data around zero and provides a measure of spread, respectively.

2. *Normalizing batch outputs:*

$$\widehat{x_i} = \frac{x_i - \mu B}{\sqrt{\sigma_B^2 + \in}}$$

Each input $\hat{x_i}$ is normalized using the batch mean and variance. A small constant ε is added to the denominator to prevent division by zero and ensure numerical stability.

3. *Scaling and shifting normalized values:*

$$y_i = \gamma \hat{x}_i + \beta$$

The normalized values $\hat{x_i}$ are scaled and shifted using two learnable parameters: γ (gamma) and β (beta). These parameters are trained alongside the model and allow the network to restore representational power if needed.

This technique is particularly useful in deep neural networks, and it notably improves learning rates, regularizes training and mitigates the risk of vanishing/exploding gradients in deep networks.

3.3 Pilot development

The pilot consists of complex CNNs, obtained by implementing MobileNetV2 as backbone, a variant of MobileNet (Sebastian *et al.*, 2012). Unlike heavy architectures like ResNet50 or VGG16, MobileNetV2 uses depthwise separable convolutions, which separate spatial and channel operations, reducing parameters by 60% without compromising accuracy. These requirements make it computationally efficient and suitable for execution on devices even with limited resources, such as mobile devices.

Some key requirements of the model are:

- *General structure:* The model has a hierarchical structure with several blocks, each containing deep convolutional layers. The blocks follow a common basic structure, including expansion layers, "depthwise separable convolution" and projection layers.
- *Depthwise separable convolution:* Many convolutions in the model are "depthwise separable"; these parts separate the convolution into two phases: a separate convolution for each channel (depthwise) followed by a 1x1 convolution (pointwise) to mix spatial and channel information.
- *Bottleneck blocks:* Some blocks were designed as bottleneck blocks, reducing dimensionality in the early layers and then expanding it again.
- *Progressive dimensions:* Feature map sizes decrease progressively through blocks, with deeper ones handling smaller feature maps.
- *Last layer:* The last layer (Conv_1) produces a feature map with 1280 channels, followed by a batch normalization layer and a Rectified Linear Unit (ReLU) activation.
- *Input size:* The expected input is a shape tensor (224, 224, 3), requiring RGB images at 224x224 pixel resolution.

A ReLU activation indicates that the last layer of the model, named "out_relu" uses a ReLU activation function widely used in neural networks, especially in convolutional network contexts like the one designed.

The ReLU function defines its output as zero for all negative inputs and linear for positive inputs. Mathematically, it is expressed by the following formula:

$$f(x) = max(0, x)$$

where *x* is the input to the function. So, if the input is positive, the function will return the input itself; if the input is negative, the function will return zero.

Going deeper, you can give more specific details about the sequential parts. Each model is divided into two sequential parts. The first part contains a more complex model with a "Functional" structure, while the second part consists of two dense layers.

Below is a graphic export showing the complexity of the layers of the first sequential part component. The graph represents only 10% of the entire structure of the first sequential part of the model for early printed books (Figure 1).

In a neural network context, a "sequential part" refers to a specific type of model architecture: for example, in Keras it is called a "Sequential Model." In a sequential model, layers are added one after another sequentially. Each layer has only one input and one output, thus creating a linear sequence of layers. This is a very common architecture, particularly suitable for building feedforward neural networks.

Specifically, in the models used for classifying manuscripts, early printed books and modern books, which are very similar to each other, the first sequential part consists of a very complex layer, including 16 blocks constituted by a certain structure. Below is the description of a block within a CNN. This block is part of a MobileNetV2 neural network architecture:

- *block_1_expand (Conv2D):* This layer is a two-dimensional convolution operation (Conv2D) named "block_1_expand" with an output shape (None, 112, 112, 48). The convolution has 384 parameters and receives input from a layer called "expanded_conv_project_BN[0][0]".
- *block_1_expand_BN (BatchNormalization):* This layer performs Batch Normalization on the output of the previous convolution layer ("block_1_expand[0][0]").
- *block_1_expand_relu (ReLU):* this layer applies the ReLU activation (Rectified Linear Unit) to the output of the previous batch normalization layer ("block_1_expand_BN[0][0]").
- *block_1_pad (ZeroPadding2D):* This layer adds zero padding to the output of the ReLU layer ("block_1_expand_relu[0][0]").
- *block_1_depthwise (DepthwiseConv2D):* This is a depthwise convolution layer named "block_1_depthwise" with an output shape (None, 56, 56, 48). It has 432 parameters and receives input from the "block_1_pad[0][0]" layer.
- *block_1_depthwise_BN (BatchNormalization):* This layer performs batch normalization on the output of the depthwise convolution layer ("block_1_depthwise[0][0]").
- *block_1_depthwise_relu (ReLU):* Applies ReLU activation to the output of the previous batch normalization layer ("block_1_depthwise_BN[0][0]").
- *block_1_project (Conv2D):* This layer performs an additional 2D convolution named "block_1_project" with an output shape (None, 56, 56, 8). It has 384 parameters and receives input from the "block_1_depthwise_relu[0][0]" layer.
- *block_1_project_BN (BatchNormalization):* Performs batch normalization on the output of the previous convolution layer ("block_1_project[0][0]").

The second sequential part has two "dense" layers. Dense layers, referred to as "Dense" in Keras and other similar deep learning frameworks, are one of the most common types of layers in artificial neural networks. They are also called fully connected layers.

A dense layer is characterized by three main components:

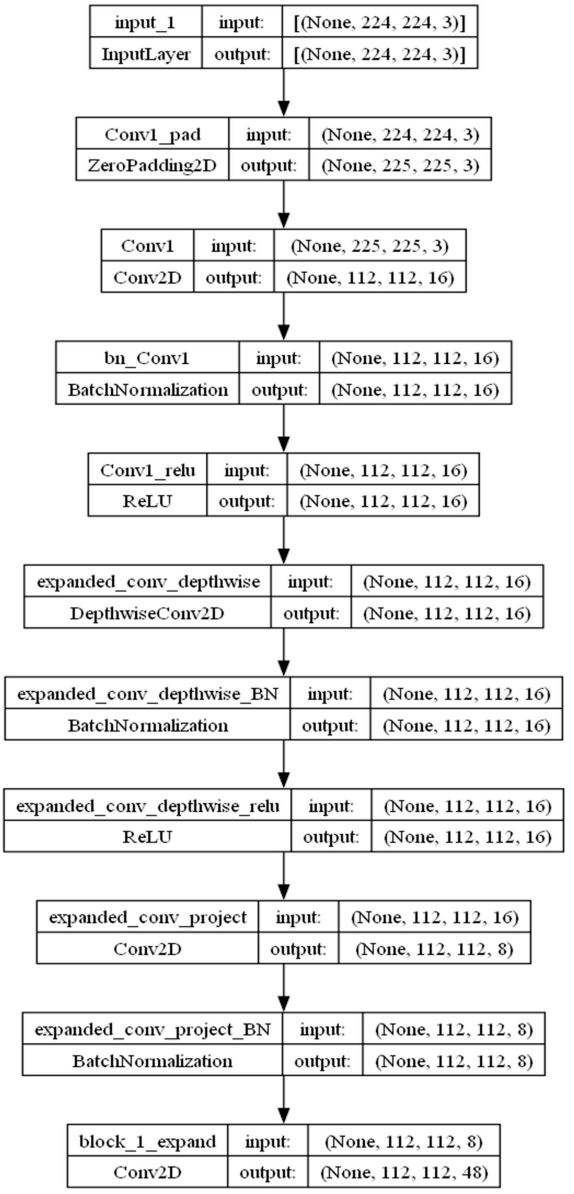

Figure 1. Layers of the component of the first sequential part
Source: Figure by authors

- *Fully connected connections:* Each neuron in a dense layer is connected to every neuron in the next layer. This means there is a direct connection between every pair of neurons.
- *Weights:* Each connection between neurons has an associated weight. These weights are learned during the neural network training process.
- *Activation function:* Each neuron has an associated activation function that determines the neuron's output given its weighted inputs. The activation function introduces nonlinearity into the network.

Dense layers are commonly used in the output layers of a neural network for classification tasks. In addition, they can also be present in the intermediate layers of the network.

Recall that, in neural networks, a neuron is the basic unit that receives one or more inputs, performs a weighted sum of these inputs, applies an activation function to the result and produces an output.

3.4 Training

The training phase involved passing data through the model using a "forward propagation" process. This allows calculating the loss between the model's predictions and the desired values. "Backpropagation" is used to calculate the gradients of loss relative to the model's weights, and finally, the model's weights are updated using an optimizer to reduce the loss.

Training is divided into epochs, where an epoch represents a complete pass through the entire training data set. In this specific case, ten "epochs" were processed using the training data and validation data.

Below a log is shown when training a neural network using TensorFlow and Keras (Figure 2).

The lines represent the following data:

	A
1	Epoch 1/10
2	12s 164ms/step - loss: 1.8127 - accuracy: 0.4573 - val_loss: 0.9502 - val_accuracy: 0.7082
3	Epoch 2/10
4	9s 150ms/step - loss: 1.0249 - accuracy: 0.6368 - val_loss: 0.8574 - val_accuracy: 0.7339
5	Epoch 3/10
6	9s 146ms/step - loss: 0.7904 - accuracy: 0.7147 - val_loss: 0.7265 - val_accuracy: 0.7296
7	Epoch 4/10
8	9s 146ms/step - loss: 0.6143 - accuracy: 0.7639 - val_loss: 0.7797 - val_accuracy: 0.6953
9	Epoch 5/10
10	9s 160ms/step - loss: 0.5589 - accuracy: 0.7735 - val_loss: 0.7101 - val_accuracy: 0.7210
11	Epoch 6/10
12	9s 146ms/step - loss: 0.4307 - accuracy: 0.8312 - val_loss: 0.6260 - val_accuracy: 0.7039
13	Epoch 7/10
14	9s 148ms/step - loss: 0.4247 - accuracy: 0.8280 - val_loss: 0.6795 - val_accuracy: 0.7768
15	Epoch 8/10
16	9s 149ms/step - loss: 0.3511 - accuracy: 0.8483 - val_loss: 0.6079 - val_accuracy: 0.7597
17	Epoch 9/10
18	9s 153ms/step - loss: 0.3399 - accuracy: 0.8579 - val_loss: 0.8017 - val_accuracy: 0.6824
19	Epoch 10/10
20	9s 147ms/step - loss: 0.3447 - accuracy: 0.8408 - val_loss: 0.7175 - val_accuracy: 0.7124

Figure 2. Training log
Source: Figure by authors

- *epoch 1/10:* indicates that the model is going through the first training epoch;
- *12s:* represents the time taken to complete the current epoch;
- *164ms/step:* indicates the average time taken to process each step (batch) during the epoch. A step is a single update of the model's weights, performed on a batch of data. A low value is generally desired, as it indicates faster training;
- *loss:* is the value of the loss function on the training data set. The loss function measures how far the model deviates from the ground truth with respect to its predictions during training. The goal is to reduce this value;
- *accuracy:* represents the model's accuracy on the training data set, measured as the percentage of correct predictions. In this case, the accuracy is 45.73%;

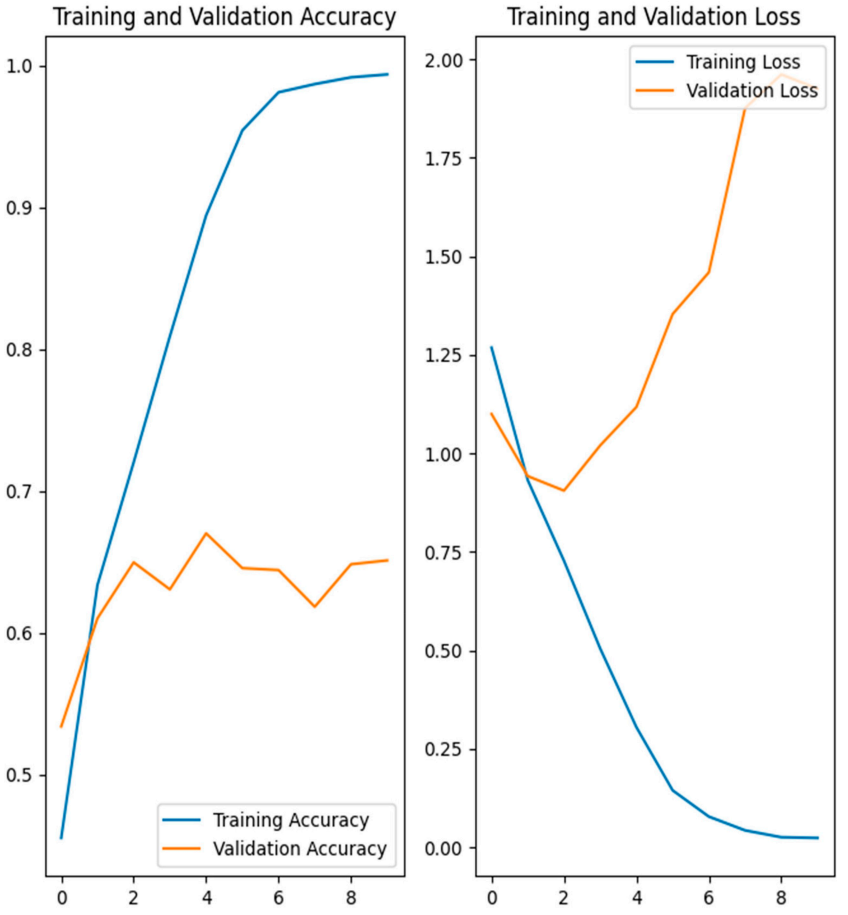

Figure 3. Non optimal accuracy
Source: Figure by authors

- *val_loss:* is the value of the loss function on the validation data set. The validation data set is a separate set of data used to evaluate the model's performance on data unseen during training; and
- *val_accuracy:* represents the model's accuracy on the validation data set. In this case, the accuracy is 70.82%.

This representation is usually generated during each "epoch" and provides useful information about the training progress, including time spent, loss and accuracy metrics on training and validation data.

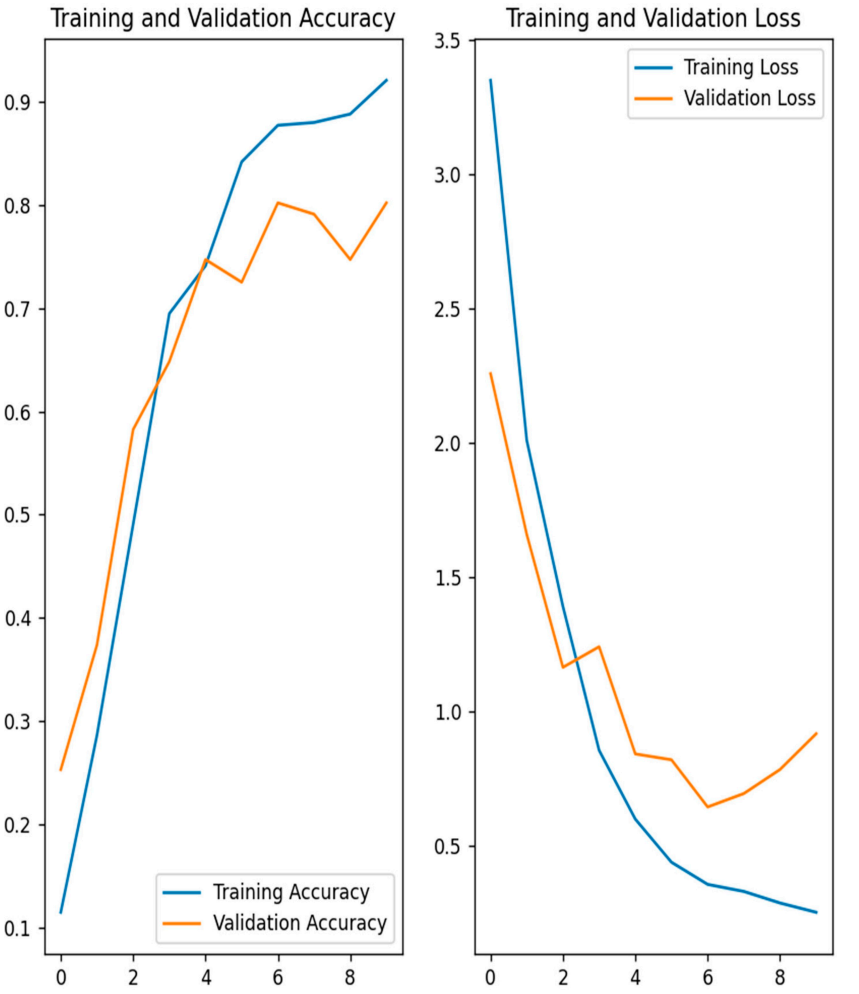

Figure 4. Accuracy on manuscripts
Source: Figure by authors

During training, the model used the data set to adjust weights based on the defined loss function. The validation set was used to monitor the model's performance on data not used in training and to evaluate any risk.

The training results with related information have been stored for later use to plot the learning curve and evaluate the model's performance.

Another interesting output to analyze at the end of training is the accuracy and loss value. The following graph shows a classic example of nonoptimal accuracy, meaning the model only achieved about 60% accuracy on the validation set (Figure 3).

As we can see, the training accuracy increases linearly over time, while the validation accuracy stalls at about 60% in the training process. Furthermore, the difference in accuracy between training and validation is evident.

When there is a limited number of training examples, the model sometimes learns from noise or unwanted details from the training examples to an extent that negatively affects

Figure 5. Accuracy of ancient and modern printed books
Source: Figure by authors

Figure 6. Error recognition
Source: Figures courtesy of University of Bari Aldo Moro

performance on new samples. This phenomenon is known as *overfitting*. It means the model has difficulty generalizing to a new data set.

There are several countermeasures to address overfitting in the training process, including data augmentation and techniques like dropout.

Following the application tests, the model's effectiveness was evaluated through a set of performance metrics, including image classification accuracy, ability to generalize to new data and the interpretability of each model's decisions.

For the three models, the accuracy trend was nearly similar. Below we represent the trend for manuscript book types (Figure 4).

Instead, these are the accuracy parameters for early printed and modern books (Figure 5):

4. Results and further perspectives

The experimentation results provide an encouraging overall picture of the developed pilot models' performance, highlighting their potential for analyzing and classifying the structures

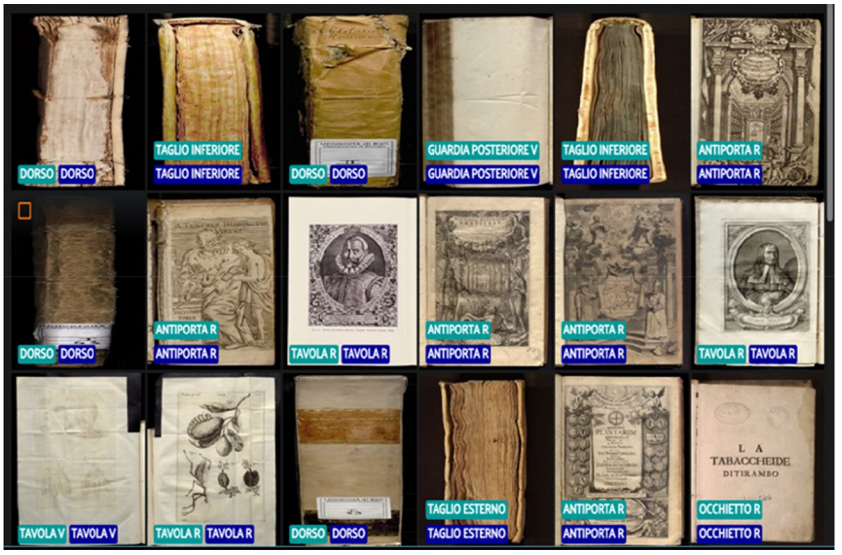

Figure 7. Example of unique correct predictions
Source: Figures courtesy of University of Bari Aldo Moro

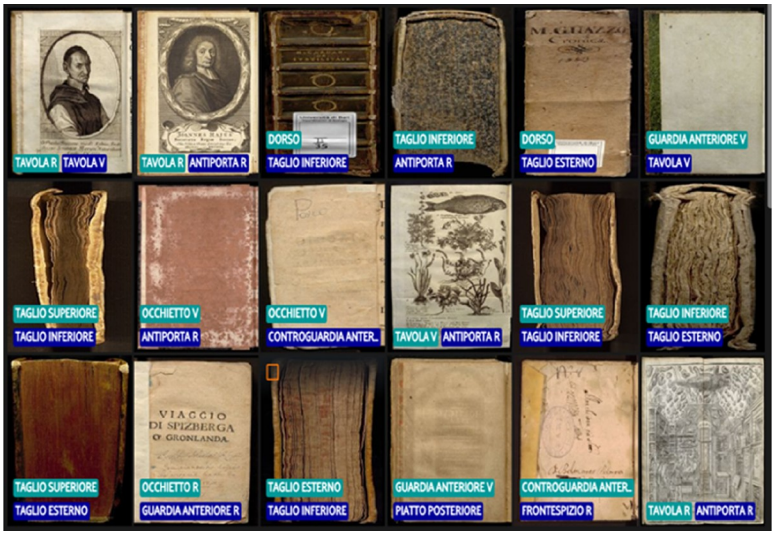

Figure 8. Example of incorrect predictions
Source: Figures courtesy of University of Bari Aldo Moro

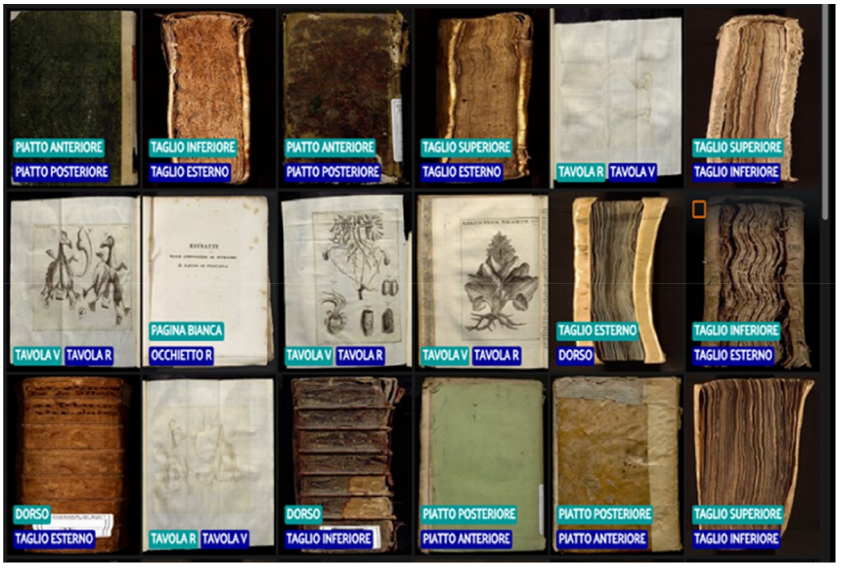

Figure 9. Incorrect annotation given as training
Source: Figures courtesy of University of Bari Aldo Moro

of the considered materials. An extensive series of tests was conducted on a diverse data set to assess their effectiveness, accompanied by a rigorous validation procedure on an even larger sample.

This pilot model shows remarkable performance in accuracy, achieving an average correct classification rate of 78% for the three analyzed types and over the full validation data set. The learning curve displayed good convergence, suggesting that further optimizations could improve the overall precision, particularly fine-tuning the hyperparameters regulating the training process and refining the machine learning model topology. Such improvements would increase accuracy and reduce uncertainty for classes that showed greater variability.

Several significant examples of training outputs are provided below (Figures 6–9), useful to visualize the model's effectiveness and identify areas for improvement:

Performance analysis highlights a balanced distribution among the different classes, showing the models' solid ability to operate effectively across a wide range of categories (Table 1).

Table 1. Performance analysis

Data set	Dimensions	Classes	Average score (%)
Manuscript	2k samples	19	75
Ancient	3k samples	25	85
Modern	2k samples	7	82

Source(s): Table by authors

These results confirm the reliability of the adopted approach and underscore its potential for concrete applications in real-world contexts.

However, some classes show room for improvement, particularly:

- *Manuscript:* cover; paper; counterguard; title page (or frontispiece);
- *Early printed:* cover; counterguard; paratext; and
- *Modern:* page; illustration plate.

Currently, experimentation is ongoing to optimize the model and further increase classification accuracy. The ambitious goal is to reach 100% precision for each of the three developed models. To achieve this goal, the research team is evaluating AI feature applications for model training and to use larger image data set of about 1,500,000 digital resources.

Notes

[1.] www.vaticanlibrary.va/it/il-patrimonio/il-progetto-di-digitalizzazione.html (last viewed 30 January 2025); https://www.bl.uk/more/digitisation/ (last viewed 30 January 2025); www.bl.uk/more/digitisation/ (last viewed 30 January 2025).

[2.] www.internetculturale.it/getFile.php?id= (last viewed 30 January 2025).

[3.] www.europeana.eu/it?__cf_chl_tk=5FmI7gOExM5C5_aw4aurFB27llcfgVw021VXZC4DT0s-1739053737-1.0.1.1-kX6ljCOajhOiEWuVDXZsrHz5tHTRzbvqrH6i86fc_7w (last viewed 30 January 2025).

[4.] https://iiif.io/ (last viewed 30 January 2025).

[5.] www.image-net.org/ (last viewed 31 January 2025).

[6.] www.robots.ox.ac.uk/~vgg/ (last viewed 30 January 2025).

[7.] https://github.com/google-research/simclr (last viewed 31 January 2025).

[8.] https://github.com/marcotcr/lime (last viewed 31 January 2025).

[9.] www.e-codices.unifr.ch/en (last viewed 31 January 2025).

[10.] www.digipal.eu/ (last viewed 31 January 2025).

[11.] www.tensorflow.org/?hlt (last viewed 31 January 2025).

[12.] https://keras.io/ (last viewed 31 January 2025).

[13.] www.internetculturale.it/getFile.php?id=44402 (last viewed 28 January 2025).

[14.] https://bibliotecadigitale.uniba.it/home (last viewed 29 January 2025).

References

Alaei, A., Nagabhushan, P. and Pal, U. (2009), "Fine classification of unconstrained handwritten Persian/Arabic numerals by removing confusion amongst similar classes", International Conference on Document Analysis and Recognition, ICDAR 2009, *Barcelona*, 26-29 July, pp. 601-605, doi: 10.1109/ICDAR.2009.181, available at: https://ieeexplore.ieee.org/document/5277578 [accessed 30 January 2025].

Aouinti, F., Eyharabide, V., Fresquet, X. and Billiet, F. (2022), "Illumination detection in IIIF medieval manuscripts using deep learning", *Digital Medievalist*, Vol. 15 No. 1, pp. 1-18, doi: 10.16995/dm.8073.

Barbuti, N. (2022), *La Digitalizzazione Dei Beni Documentali: metodi, Tecniche, Buone Prassi*, Editrice Bibliografica, Milano.

Barbuti, N. and De Bari, M. (2024), "Libri e biblioteche tra museabilità e musealizzazione digitale: sogno o realtà?", in Di Silvestro, A. and Spampinato, D. (Eds), *Proceedings Del XIII Convegno Annuale AIUCD Me.Te. Digitali. Mediterraneo in Rete Tra Testi e Contesti*, pp. 84-88, doi: 10.6092/unibo/amsacta/7927, available at: https://amsacta.unibo.it/id/eprint/7927/1/AIUCD2024-proceedings.pdf

Ciotti, F. (2023), "Minerva e il pappagallo: IA generativa e modelli linguistici nel laboratorio dell'umanista digitale", *Testo e Senso*, Vol. 26, pp. 289-315, doi: 10.58015/2036-2293/671, available at: https://testoesenso.it/index.php/testoesenso/article/view/671/587

Ćosović, M. and Janković, R. (2020), "CNN classification of the cultural heritage images", 19th International Symposium INFOTEH-JAHORINA (INFOTEH), *IEEE*, doi: 10.1109/INFOTEH48170.2020.9066300, available at: https://ieeexplore.ieee.org/document/9066300 [accessed 30 January 2025].

Goodfellow, I., Bengio, Y. and Courville, A. (2016), "Deep learning", The MIT Press, Cambridge, available at: www.deeplearningbook.org/

Haralick, R.M., Shanmugam, K. and Dinstein, I. (1973), "Textural features for image classification", *IEEE Transactions on Systems, Man, and Cybernetics*, SVol. SMC-3 No. 6, pp. 610-621, available at: https://haralick.org/book_chapters/TexturalFeatures.pdf [accessed 30 January 2025].

He, K., Zhang, X., Ren, S. and Sun, J. (2016), "Deep residual learning for image recognition", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), doi: 10.1109/CVPR.2016.90, available at: https://ieeexplore.ieee.org/document/7780459

Hsu, D., Kakade, S.M. and Zhang, T. (2012), "A spectral algorithm for learning hidden Markov models", *Journal of Computer and System Sciences*, Vol. 78 No. 5, pp. 1460-1480, arXiv:0811.4413v6, doi: 10.48550/arXiv.0811.4413, available at: https://arxiv.org/abs/0811.4413 [accessed 30 January 2025].

Joachims, T. (1998), "Text categorization with support vector machines: learning with many relevant features", *ECML*, Vol. '98, pp. 137-142, doi: 10.1007/BFb0026683, available at: https://link.springer.com/chapter/10.1007/BFb0026683

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012), "ImageNet classification with deep convolutional neural networks", in Pereira, F., Burges, C.J., Bottou, L. and Weinberger, K.Q. (Eds), *Advances in Neural Information Processing Systems*, Vol. n. 25, available at: https://papers.nips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

Kumbhar, R. (2012), "Automatic book classification, reclassification and non-classificatory approaches to knowledge organization", in Kumbhar, R. (Ed.), *Library Classification Trends in the 21st Century*, Elsevier, Woodhead, available at: www.sciencedirect.com/science/article/abs/pii/B9781843346609500060

LeCun, Y., Bengio, Y. and Hinton, G. (2015), "Deep learning", *Nature*, Vol. 521 No. 7553, pp. 436-444, doi: 10.1038/nature14539.

Liu, J., Ma, X., Wang, L. and Pei, L.4. (2024), "How can generative artificial intelligence techniques facilitate intelligent research into ancient books?", *Journal on Computing and Cultural Heritage*, Vol. 17 No. 4, pp. 1-20, doi: 10.1145/3690391.

Lowe, D.G. (2004), "Distinctive image features from Scale-Invariant keypoints", *International Journal of Computer Vision*, Vol. 60 No. 2, pp. 91-110, doi: 10.1023/B:VISI.0000029664.99615.94, available at: https://link.springer.com/article/10.1023/B:VISI.0000029664.99615.94 [accessed 30 January 2025].

Milleville, K., Chandrasekar, K.K.T. and Verstockt, S. (2023), "Automatic extraction of specimens from multi-specimen herbaria", *Journal on Computing and Cultural Heritage*

(JOCCH), Vol. 16 No. 1, pp. 1-15, doi: 10.1145/3575862, available at: https://dl.acm.org/doi/10.1145/3575862

Mohammed, H.A., Märgner, V. and Ciotti, G. (2021), "Learning-free pattern detection for manuscript research: an efficient approach toward making manuscript images searchable", *International Journal on Document Analysis and Recognition (IJDAR)*, Vol. 24 No. 3, pp. 167-179, doi: 10.1007/s10032-021-00371-7, available at: https://link.springer.com/article/10.1007/s10032-021-00371-7.

Neudecker, C. (2022), "Cultural heritage as data: digital curation and artificial intelligence in libraries", in Paschke, A., Rehm, G., Neudecker, C. and Pintscher, L. (Eds), *Proceedings of the Third Conference on Digital Curation Technologies (Qurator 2022)*, Berlin, 19th-23rd, available at: https://ceur-ws.org/Vol-3234/paper2.pdf

Ojala, T., Pietikainen, M. and Maenpaa, T. (2002), "Multiresolution Gray-Scale and rotation invariant texture classification with local binary patterns", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24 No. 7, pp. 971-987, doi: 10.1109/TPAMI.2002.1017623.

Peng, H.C., Long, F. and Ding, C. (2005), "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27 No. 8, pp. 1226-1238, doi: 10.1109/TPAMI.2005.159, available at: www.computer.org/csdl/journal/tp/2005/08/i1226/13rRUy3xY96 [accessed 30 January 2025].

Reynolds, T., Dhali, M.A. and Schomaker, L. (2022), "Image-based material analysis of ancient historical documents", *arXiv Preprint arXiv:2203.01042*, doi: 10.48550/arXiv.2203.01042, available at: https://arxiv.org/abs/2203.01042

Roald, M., Birkenes Breder, M. and Gunnarsønn Bagøien, J. (2024), "Visual navigation of digital libraries: retrieval and classification of images in the national library of Norway's digitised book collection", doi: 10.48550/arXiv.2410.14969, [accessed 15 August 2025].

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. and Chen, L.-C. (2018), ""MobileNetV2: Inverted residuals and linear bottlenecks", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4510-4520, available at: https://openaccess.thecvf.com/content_cvpr_2018/papers/Sandler_MobileNetV2_Inverted_Residuals_CVPR_2018_paper.pdf

Schomaker, L. (2019), "A Large-Scale field test on word-image classification in large historical document collections using a traditional and two deep-learning methods", *arXiv Preprint arXiv:1904.08421*, doi: 10.48550/arXiv.1904.08421, available at: https://arxiv.org/abs/1904.08421

Sebastian, V.,B., Unnikrishnan, A. and Balakrishnan, K. (2012), "Grey level co-occurrence matrices: generalisation and some new features", *International Journal of Computer Science, Engineering and Information Technology (IJCSEIT)*, Vol. 2 No. 2, doi: 10.5121/ijcseit.2012.22, available at: https://arxiv.org/pdf/1205.4831

Shanmugamani, R. (2018), *Deep Learning for Computer Vision: Expert Techniques to Train Advanced Neural Networks Using TensorFlow and Keras*, Packt Publishing.

Silla, C.N. and Freitas, A.A. (2011), "A survey of hierarchical classification across different application domains", *Data Mining and Knowledge Discovery*, Vol. 22 Nos 1-2, pp. 31-72, doi: 10.1007/s10618-010-0175-9.

Swain, M.J. and Ballard, D.H. (1991), "Color indexing", *International Journal of Computer Vision*, Vol. 7 No. 1, pp. 11-32, doi: 10.1007/BF00130487, available at: https://link.springer.com/article/10.1007/BF00130487#citeas [accessed 30 January 2025].

Tan, M. and Le, Q.V. (2019), "EfficientNet: rethinking model scaling for convolutional neural networks", *Proceedings of the 36th International Conference on Machine Learning (ICML), Long Beach, CA*, pp. 6105-6114, available at: https://arxiv.org/pdf/1905.11946

Vidal-Gorène, C., Camps, J.B. and Clérice, T. (2024), "Synthetic lines from historical manuscripts: an experiment using GAN and style transfer", in Foresti, G.L., Fusiello, A. and Hancock, E. (Eds), *Image Analysis and Processing - ICIAP 2023 Workshops. Lecture Notes in Computer Science*, Vol. 14366 Springer, Cham, Chapter 40, doi: 10.1007/978-3-031-51026-7_40., available at: https://link.springer.com/chapter/10.1007/978-3-031-51026-7_40

Zhang, W. and Yoshida, T. (2010), "A comparative study of TF-IDF, LSI and Multi-Words for text classification", *Expert Systems with Applications*, Vol. 38 No. 3, pp. 2758-2765, doi: 10.1016/j.eswa.2010.08.066.

Further reading

Yang, Y. and Liu, X. (1999), "A Re-Examination of text categorization methods", *Sigir '99*, pp. 42-49, doi: 10.1145/312624.312647.

Corresponding author

Nicola Barbuti can be contacted at: nicola.barbuti@uniba.it