Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Computational Visual Cataloguing

Hussein Mohammed

hussein.adnan.mohammed@uni-hamburg.de

The Visual Manuscript Analysis lab (VMA), The Centre for the Study of Manuscript Cultures (CSMC), Universität Hamburg https://orcid.org/0000-0001-5020-3592

Quang-Vinh Dang

The Visual Manuscript Analysis lab (VMA), The Centre for the Study of Manuscript Cultures (CSMC), Universität Hamburg https://orcid.org/0000-0002-6715-7112

Anna Kinder

German Literature Archive (DLA)

Sandra Richter

German Literature Archive (DLA)

Research Article

Keywords: computer vision, computational cataloguing , word detection, historical documents, Notebooks

Posted Date: February 12th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-8849242/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Computational Visual Cataloguing

Hussein Mohammed[1*†], Quang-Vinh Dang[1†], Anna Kinder[2], Sandra Richter[2]

[1*]The Visual Manuscript Analysis lab (VMA), The Centre for the Study of Manuscript Cultures (CSMC), Universität Hamburg, Warburgstr. 26, Hamburg, 20354, Hamburg, Germany.

[2]German Literature Archive (DLA), Schillerhoehe 8-10, Marbach am Neckar, 71672, Baden-Württemberg, Germany.

*Corresponding author(s). E-mail(s): hussein.adnan.mohammed@uni-hamburg.de;
Contributing authors: vinh.dang@uni-hamburg.de; Anna.Kinder@dla-marbach.de;
Sandra.Richter@dla-marbach.de;
†These authors contributed equally to this work.

Abstract

Digitisation of handwritten heritage outpaces cataloguing. Manual workflows remain slow, automatic text recognition struggles with heterogeneous scripts and layouts, and text-only transcriptions omit visual cues such as orientation, colour and writing implements. This limits discovery and reuse in libraries, archives and museums, especially for notebooks with complex, irregular layouts. We propose Computational Visual Cataloguing (CVC), an extensible computer vision framework that performs content-bearing page detection, word localisation irrespective of script or language, and word-level visual-attribute classification (orientation, colour, writing implement). Our framework is designed to learn robustly from densely written manuscript pages and limited human annotations, reflecting the natural challenges of historical handwritten collections. These outputs populate structured, searchable catalogue fields. The contributions of this work are: (1) the formulation of CVC and a word-level visual-attribute schema for handwritten heritage; (2) task-specific methods and publicly released models for page detection, cross-script word localisation and visual-attribute classification; (3) a manually annotated dataset and full computational catalogue for Rainer Maria Rilke's notebooks; and (4) ScriptSight, a tool for exploring and exploiting these catalogues. We evaluate CVC on Rilke's notebooks, a culturally significant corpus with multilingual content, varied orientations and colours, diverse implements and irregular layouts. we report quantitative performance for individual components (including F1-based evaluation and ablation studies) and qualitative assessment of the integrated system, which together demonstrate its applicability to complex pages and its practical utility for exploratory cataloguing.

Keywords: Computational Cataloguing , word detection, historical documents, Notebooks

1 Introduction

The digitisation of handwritten material in libraries, archives and museums has become increasingly rapid and efficient over the last decades. Nevertheless, cataloguing digitised manuscript collections does not keep pace with the accelerating rates of digitisation because it

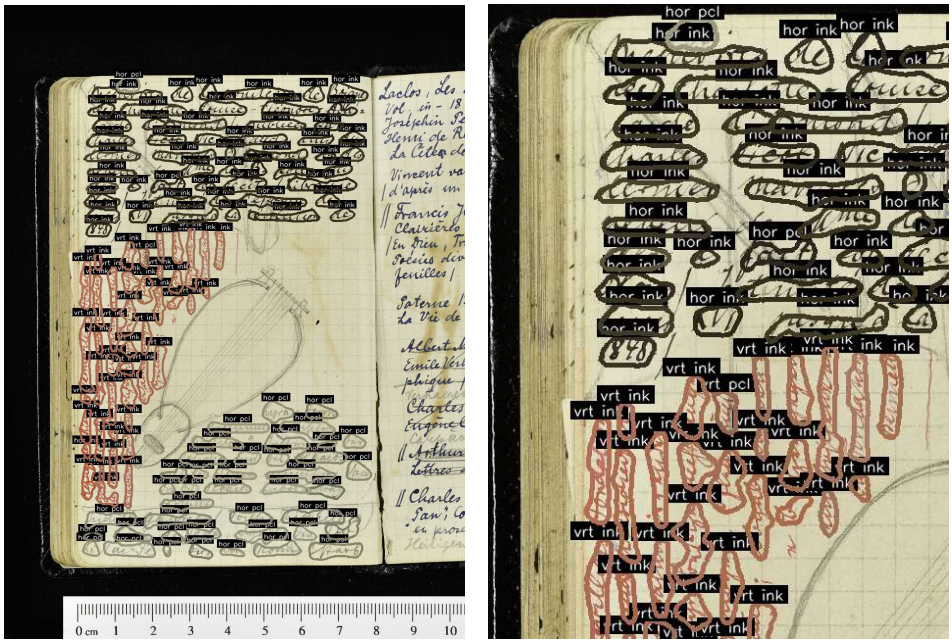


Fig. 1 An example of Rainer Maria Rilke's notebook pages from the Deutsches Literaturarchiv Marbach (DLA), A: Rilke-Archiv Gernsbach. Model predictions are overlaid on the images. The image on the right is a zoomed-in section of the images on the left. Polygon colour represents the predicted word colour, while the predicted orientation and writing implement are shown as labels on a black background. "hor", "vrt", and "pcl" stand for horizontal, vertical, and pencil, respectively.

remains predominantly a manual process, requiring considerable time and resources. Although Handwritten Text Recognition (HTR) aims to automate textual transcription, current HTR systems often exhibit limited performance on historical manuscripts due to handwriting variability, complexity of scripts and layouts, and typical degradation. Even in cases where HTR successfully transcribes textual content, broader computer vision methods remain essential, as they capture additional visual details crucial for understanding the historical context of these artefacts, such as handwriting style, text colour, and writing implements, details often lost when relying solely on textual transcriptions. Consequently, institutions increasingly face the problem that collections remain practically inaccessible for users and thus de facto invisible for potential usage communities.

2 Related Work

We propose an alternative method: automatic cataloguing using a computer-vision-based approach. Computational Visual Cataloguing (CVC) is not intended to replace manual cataloguing; rather, it offers a unique opportunity to open up otherwise inaccessible collections, as it automatically extracts visual information such as the distribution of text on the page, text colour, word orientation, and writing implements used in each text block. Once relevant details are extracted and structured into a searchable format, computational systems can utilise this information to support searching, filtering, and even preliminary cataloguing. Thus, libraries, museums and archives can build on CVC to make large-scale digitised holdings accessible.

In this study, we test CVC on a type of manuscript that has proved particularly challenging for manual cataloguing and navigation: notebooks. As they are not produced in a single instance, notebook pages often contain numerous and highly heterogeneous pieces of content written in diverse scripts and languages, with irregular layouts. This computational cataloguing approach serves to complement existing catalogued collections, or to provide preliminary cataloguing for uncatalogued collections, by automatically analysing significant visual elements of the textual content of digitised notebook pages. The notebooks of Rainer Maria Rilke present a

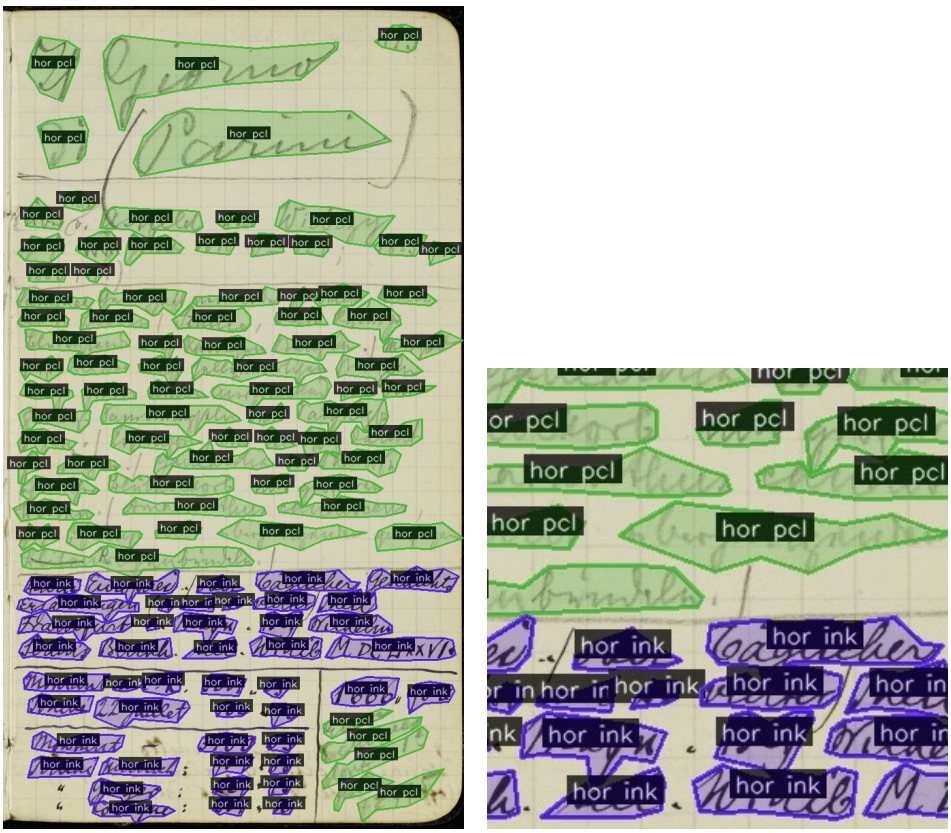
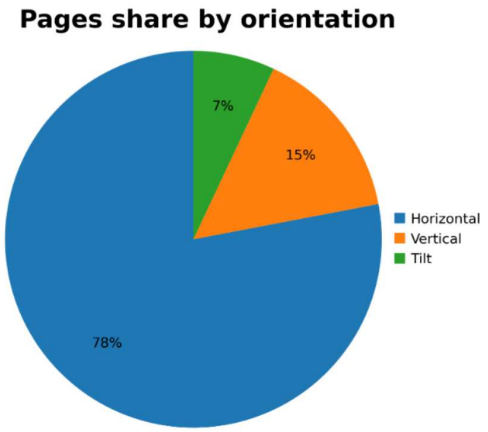


Fig. 2 An example from our DLA-RMR dataset. The manual annotations are visualised on the images as polygons and label text on a black background. Polygon colours are used only to recognise words with different labels. The image on the right is a zoomed-in section of the images on the left. "hor" stands for horizontal, and "pcl" stands for pencil.

particularly interesting use case for several reasons. The notebooks themselves have considerable cultural and historical significance. Moreover, they contain handwriting in different languages, orientations, colours, and writing implements. The layout is highly complex, differing from page to page, even within the same notebook, where handwritten content may overlap and even appear inside sketches and other visual elements.

The proposed system operates as a sequential pipeline, with each component extracting specific visual details and recording them in a structured catalogue. It first detects the main content-bearing pages of digitised notebooks, then locates every word, regardless of script or language. For each word, further visual attributes, such as orientation, colour and writing implement, are determined. A software tool with a simple GUI has also been developed to visually explore images and their CVC-generated features. Visual attributes such as text orientation, writing implements (pencil or ink), and text colour can be filtered concurrently to isolate pages matching specified criteria. Matching pages can be previewed as thumbnails with optional overlays indicating detected text regions, and saved in organised, date-stamped folders for review or further processing.

In this work, we present CVC, an extensible framework for computational visual cataloguing of handwritten heritage that supports a flexible, modular pipeline. We develop and publicly release task-specific methods and models for content-bearing page detection, cross-script word localisation with orientation, colour and implement classification, and publish a manually annotated dataset from Rainer Maria Rilke's notebooks, and the full catalogue for all of his notebooks. We also provide ScriptSight, an interactive tool that enables curators, archivists and researchers to explore these catalogues through concurrent filtering, thumbnail preview and export.



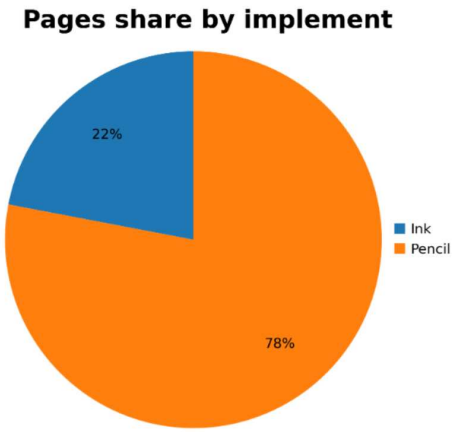
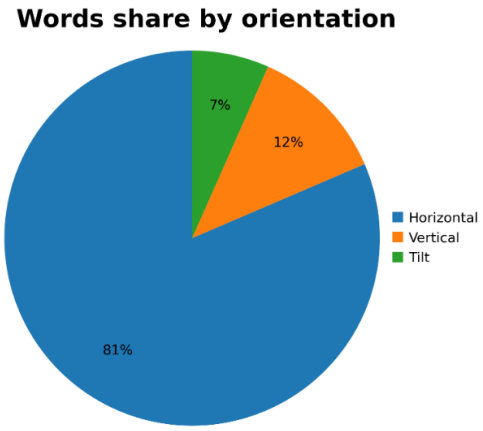


Fig. 3 Distribution of visual attributes in the DLA-RMR dataset at the page level across the 100 curated pages: page orientation (65% horizontal, 25% vertical, 10% tilted) and writing implement (34% ink, 66% pencil).



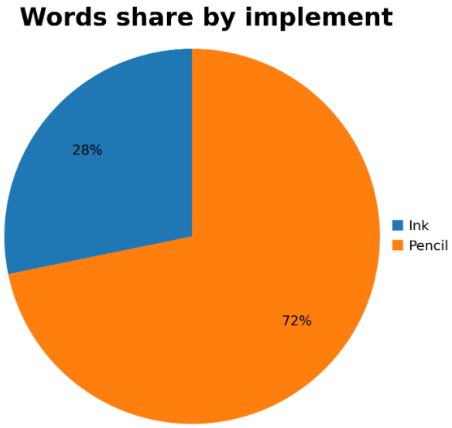


Fig. 4 Distribution of visual attributes in the DLA-RMR dataset at the word level over all annotated words from the same set: (a) word orientation: 84% horizontal, 12% vertical, 4% tilted; (b) writing implement: 29% ink, 71% pencil.

End-to-end platforms for historical manuscripts, most prominently Transkribus and eScriptorium (based on Kraken), have transformed transcription-centred workflows by combining layout segmentation, HTR and search over textual surrogates [1–3]. Beyond HTR stacks, MONK pioneered large-scale visual indexing and word-spotting to support discovery without full transcription, and DIVAServices exposed a web API of document-image methods to facilitate processing pipelines [4–6]. Annotation platforms in the IIIF ecosystem (e.g. Recogito) enable collaborative, standards-based enrichment but generally rely on manual markup rather than automated, word-level visual analysis [7]. Collectively, these systems focus on transcriptions, keyword search or human annotation; they seldom catalogue page-anchored visual attributes (e.g. tight word polygons with orientation, colour and implement) as first-class, queryable metadata. Our Computational Visual Cataloguing (CVC) instead positions computer vision to analyse visual structure in manuscript images, page selection, cross-script word localisation and patch-level material cues, so that edited texts and exploratory analysis can be informed by page-anchored evidence that conventional transcription omits.

For isolating primary pages in bound notebooks, deep segmentation and detection approaches offer strong baselines: dhSegment demonstrates a generic CNN that adapts to page extraction, while PageNet explicitly targets historical page boundaries; earlier page-frame

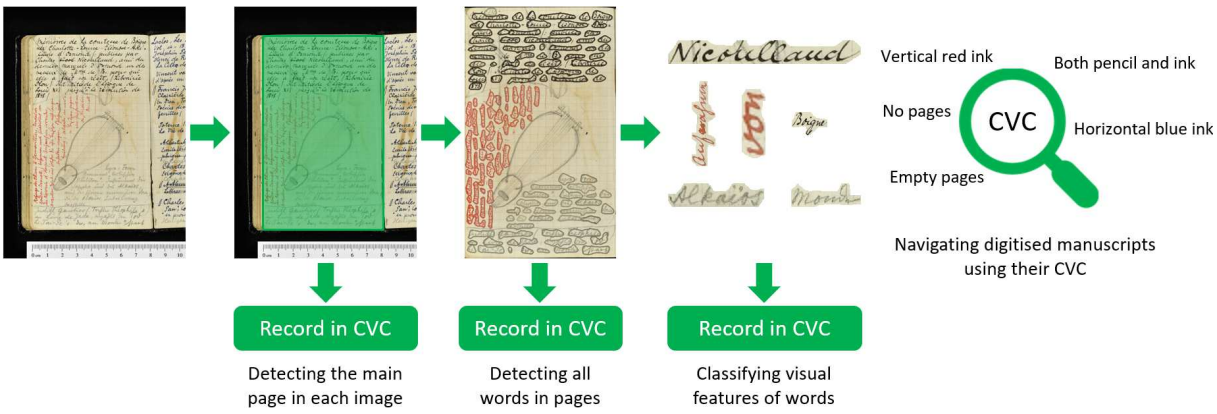


Fig. 5 An illustration of the components of the Computational Visual Cataloguing (CVC) system: the input is a raw image, and the output is a searchable structure.

and double-page methods address spread separation and border suppression in digitisation workflows [8–10]. Our module differs in targeting notebook images with partial opposing pages and low texture: a one-stage detector fine-tuned for edge-dominated paper achieves high recall and low latency, aligning with operational constraints of large-scale digitisation where a single content-bearing page must be cropped reliably.

For word localisation under scarce labels and dense handwriting, document- and scene-text literature increasingly models polygonal instances and cross-script robustness. Scene-text detectors such as CRAFT, PSENet, PAN and DB learn character-aware or segmentation-style maps that yield sharp contours and separate close instances; ABCNet further parameterises curved text with Bézier curves [11–15]. Historical-document efforts like docExtractor [16] exploit synthetic pretraining for element extraction with minimal ground truth, while HierText [17] and ICDAR challenges [18, 19] popularised polygonal, orientation-sensitive evaluation in multi-lingual settings; MapText extended polygonal word detection to heritage maps [20]. Our detector follows this trajectory but is tailored to notebooks: it outputs tight, stroke-aligned word polygons with overlap suppression to enable clean crops for downstream (colour, orientation and implement) cues, and increases label-efficiency via an auxiliary segmentation branch. To inject vision–language priors without heavy annotation, we employ CLIP prompting through Context Optimization (CoOp) with learnable prompts and Conditional Context Optimization (CoCoOp) with image-conditioned prompts; in this text-conditioned prompting (TCM) style, the text encoder is frozen while prompts adapt per page, a strategy shown to improve base-to-novel generalisation with few labels [21, 22].

Colour, orientation and writing-implement cues remain comparatively underexplored as word-level, page-anchored catalogue fields in historical manuscripts. For colour, classic local-thresholding and foreground-selection methods in document enhancement (e.g. Sauvola, Wolf–Jolion and Bernsen) underpin robust extraction of ink pixels from degradations [23–25]. Recent non-invasive studies indicate that RGB analyses can characterise inks in cultural-heritage contexts, but typically at page or region granularity rather than per word [26]. By contrast, most implement and ink identification research in forensics relies on chemical or spectral measurements rather than catalogue-oriented RGB scans [27, 28]. To the best of our knowledge, there is no published work that (i) assigns word-level colour names and exact RGB codes at scale for historical notebooks, or (ii) performs word-level orientation and implement classification directly from RGB digitisation. Our choices, training-free foreground selection with perceptual colour naming and a lightweight CLIP-based patch classifier for orientation and implement, therefore fill this gap while staying compatible with polygonal, multi-oriented detection pipelines [11, 14].

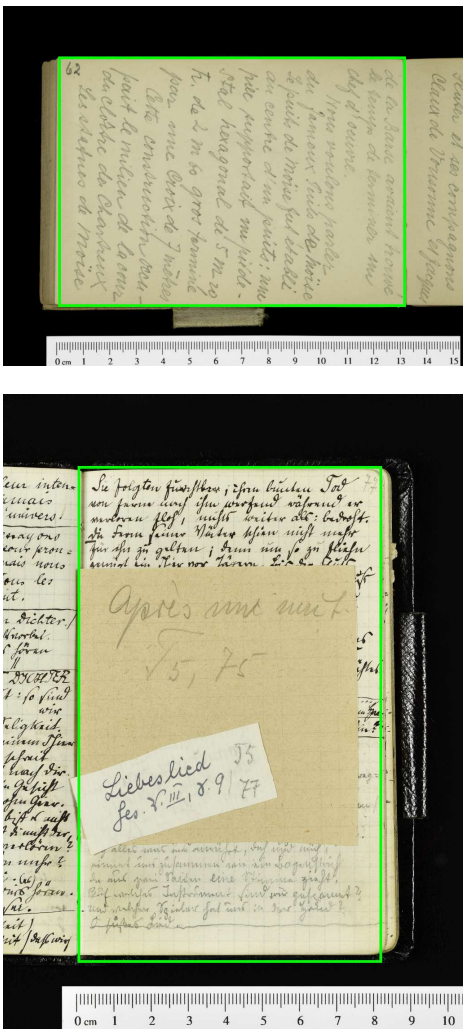


Fig. 6 Examples of Rainer Maria Rilke's notebook pages from the Deutsches Literaturarchiv Marbach (DLA), A: Rilke-Archiv Gernsbach. Page detection is presented as a green bounding-box.

3 Dataset

The German Literature Archive (Deutsches Literaturarchiv Marbach, DLA) [29] is a non-university research and archival institution and ranks among the most significant and extensive literary archives in Europe. Its collections preserve a wealth of source material central to understanding literary and intellectual history from 1750 to the present. The holdings include around 1,600 literary estates and collections of writers and scholars, more than 50 archives of literary publishers, and over 450,000 visual and material objects. The Manuscripts Department at the DLA holds numerous notebooks, which are among the typical materials found in writers' and scholars' estates [30]. These usually small-format, bound booklets bear a wide range of handwritten and drawn traces of use—everyday notes, diary-like entries, observations and travel notes, as well as drafts of letters and literary texts, together with various inserted items, some contributed by third parties, such as photographs, postcards, letters, dried plants, and more. In their diversity and coexistence of different textual forms, notebooks differ from other handwritten documents and constitute an interface between everyday observations, readings, and thoughts on the one hand, and poetic drafts and creative projects on the other. By documenting processes of work, notebooks form an integral part of an author's poetic practice and should be regarded as belonging to the respective body of work [31].

3.1 The Notebooks of Rainer Maria Rilke

Together with the private estate of the author Rainer Maria Rilke (1875–1926), acquired in 2022 and known as the Rilke Archive Gernsbach (RAG), a number of notebooks entered the collections of the DLA. According to current findings, the total of 68 artefacts classified as notebooks includes the 58 items previously known as Taschenbücher ("pocketbooks"), which are the subject of the present study [32]. The RAG has been fully catalogued, is searchable online via the DLA's OPAC Kallías, and can be consulted on campus [33]; for conservation reasons, the notebooks are primarily available in digital form [34].

Rilke used pencil and ink for his writing, and his handwriting varied according to the writing circumstances [35] and his stage of life. The attribution of many notebooks to the times and places of their creation is possible by identifying references in the entries, though exact dates or references to the day of the week are often missing and frequent changes of location, particularly in the Swiss notebooks from 1919 to 1921, make a precise overview necessary. The frequency of entries varies considerably; some years are sparsely recorded, and in some cases there are almost empty notebooks as well. Most of the notebooks are written in German and French, with the latter



Fig. 7 Augmented manuscript pages with word-level polygon annotations (green). The image–polygon pairs remain aligned after geometric and photometric transformations, helping increase data diversity while preserving handwriting structure.

given considerable space during the Parisian and Swiss years. Other languages, among them Danish, Ancient Greek, Italian, Russian, and Spanish, also appear, such that it is possible, to a certain degree, to speak of multilingual and multiscriptural holdings. In addition, they contain numerous drafts of letters and poems. Alongside the written elements, there are several sketches in the notebooks. Rilke seems to have sketched them rather hastily, and they appear mainly to serve his memory. The sketches form distinct units of content and are not necessarily combined with writing. Most of the time, they capture visual impressions and stand alone. They also include about 95 dried plants.

3.2 The Manually-Annotated DLA-RMR Dataset

The images of the DLA-RMR dataset [36] derive from The notebook pages of Rainer Maria Rilke. The corpus comprises two aligned sets of 100 PNG images each: a *cropped_images* collection containing only the primary page extracted from the corresponding full scans, and a *full_images* collection representing the complete notebook scans in which that primary page appears alongside additional material. Word annotations are provided for the cropped pages, whereas page annotations are supplied for the full scans.

Word annotations are split into train and test sets. Each word instance is described by a polygon together with auxiliary metadata, including the word's orientation (horizontal, vertical, or tilted) and the writing implement (ink or pencil). Page annotations are given in YOLO format within the *page_annotations* directory, with a single class (page) specified in *classes.txt*. For each image in *full_images*, the corresponding annotation file encodes the bounding box of the primary page. See Fig. 2 for an example of annotated page.

In addition, the distributions of these visual attributes are summarized in Fig. 3 (two pie charts: page share by implement and by orientation) and Fig. 4 (two pie charts: word share by implement and by orientation). The set of 100 images was deliberately curated to span a broad range of orientations and writing implements rather than to reflect their overall prevalence across the notebooks. Consequently, these proportions should not be interpreted as representative of the full archive. Notably, many pages mix attributes, e.g. ink and pencil, and multiple orientations. In other words, page annotations aggregate heterogeneous within-page word distributions.

4 Computational Visual Cataloguing

The proposed system consists of a sequential pipeline composed of multiple interdependent components, each designed to extract specific visual details, record them in catalogue file, and pass the result to the subsequent component. The catalogue itself has a predefined structure designed to accommodate incompleteness, allow extension, and facilitate the updating of existing records with more precise details. First, it

detects primary pages within images of a digitised notebook. By 'primary page', we mean the main content-bearing page(s). In a second stage, CVC detects every word within the identified pages, irrespective of script or language, including the identification of empty pages. A tight polygon is generated around each detected word, with coordinates arranged in a precise sequence. Once the exact coordinates of each word's location on the page are established, CVC identifies various pieces of visual information, such as orientation, colour and writing implement. See the general illustration in 5.



Fig. 8 Sample word patches used for orientation and writing implement classification. Each patch is tightly cropped from a detected word polygon.

To mitigate data scarcity, the system leverages publicly available pretrained models, most notably CLIP [37], which has been trained on hundreds of millions of text–image pairs. CLIP inherently bridges vision and language, forming an inborn text-friendly model [38] that learns a shared semantic space between image and text. Its neurons can directly respond to textual structures [39], making it especially suitable for analysing documents. By exploiting this intrinsic cross-modal alignment, the CVC pipeline generalises effectively under limited supervision and accurately extracts word-level attributes such as location, orientation, and type of writing implement.

4.1 Detecting Primary Pages

We address the problem of reliably isolating the primary notebook page in images that may contain a partial opposing page and other background elements. A one-stage YOLOv11 detector [40] is fine-tuned from large-scale pre-training on only few annotated images with a single "page" class, with images resized to 1024 px during training. YOLOv11 is chosen for its favourable accuracy–latency trade-off and strong recall on edge-dominated, low-texture targets typical of paper. Transfer learning leverages generic contour and shape features while limiting annotation burden; annotating the primary page with a bounding box in only a small set of images takes an insignificant amount of effort and time. Optimisation uses AdamW with a conservative base learning rate to stabilise fine-tuning of a high-capacity backbone; decoupled weight decay provides principled regularisation on a modest dataset. Augmentations are kept lightweight and primarily photometric, with mild geometric jitter, to promote invariance to illumination and viewpoint while avoiding unrealistic document deformations.

At inference, the detector may return multiple page hypotheses. No further post-processing was required, as all instances of primary pages were detected, and the confidence score was sufficient to select the page from source images. On the validation split, the model attains high detection quality ($precision = 0.985, recall = 1.0, mAP@50 = 0.990, mAP@50–95 = 0.983$), indicating robust localisation with minimal misses. See Fig. 6 for detection examples.

4.2 Detecting Words

This component addresses two practical challenges in historical manuscript: scarce annotations and densely packed handwriting with very tight inter-word spacing. This contrasts with conventional OCR systems, which typically process text in relatively clean backgrounds and use coarse rectangular bounding boxes that do not capture fine-grained spatial detail. Inspired by the approach in Turning a CLIP Model into a Scene Text Detector (TCM) [39], which performs well on scene text, we adapt and develop it further to enhance its effectiveness for the specific challenges of historical manuscripts. To address the issue of limited labelled data, we employ replayable, paired image–polygon augmentations specifically tailored to the structure of manuscripts. Following a TCM-style setup, we keep the CLIP text encoder frozen, while learnable prompts are initialised with the default phrase "text pixels in a historical manuscript," and an image-conditioned language-prompt generator adapts the textual cues for each manuscript page. This design improves generalisation under limited supervision.

For dense layouts, the detector returns tight polygonal word instances that follow handwriting



Fig. 9 Feature visualizations of selected CLIP neurons showing strong activation for specific textual concepts, as reported in [38]. These examples illustrate CLIP's inherent ability to associate visual patterns with linguistic structure, even without explicit OCR supervision.

strokes and suppress overlaps, enabling reliable word-patch cropping and downstream colour, orientation, and writing implement recognition. During training, we add a lightweight auxiliary pixel-level word-segmentation branch whose supervision sharpens local boundaries and stabilises optimisation under label scarcity. Because full masks can merge neighbouring words into one region, the detection head is trained with shrunken masks that preserve small gaps between adjacent words; after detection we slightly expand each region to recover the full word extent while keeping instances separate.

4.2.1 Data and Augmentation

We train on our DLA-RMR dataset, which is modest in size. To improve generalisation and reduce overfitting while preserving the geometric structure of the documents, we apply replayable, paired image–polygon augmentations. Each original page yields multiple augmented variants through transformations that maintain the alignment between images and their word-level polygon annotations as Fig. 7. These polygons are then simplified and clipped to image boundaries, with malformed or excessively small shapes discarded to ensure annotation quality.

4.2.2 CLIP-Based Word Detection in Historical Manuscripts

We adopt a segmentation-style word detector built on Differentiable Binarisation [14]. The network predicts a probability map for text pixels and a threshold map that together yield a sharpened binary mask. From this mask, we extract contours, slightly expand them to recover full word extents, and simplify the boundaries to obtain compact polygons. A small-width constraint suppresses spurious fragments, and a final overlap-suppression step removes nested and duplicated instances. This design produces tight, stroke-aligned word polygons that minimise overlaps on dense pages, crucial for downstream colour, orientation, and writing implement classification and for reliable word-patch cropping.

Injecting vision–language priors.We inject vision–language priors by integrating CLIP in a TCM-style manner: the CLIP text encoder remains frozen, while learnable prompt tokens and an image-conditioned language-prompt generator adapt the textual cues dynamically for each page. A visual-prompt generator based on cross-attention propagates these cues into the visual stream, yielding text-aware locality features that help delineate word regions under label scarcity. This plug-in design integrates seamlessly with standard detectors, requiring no additional pretraining.

Loss Function. Alongside the DB-style detection head, we introduce an additional pixel-level text supervision derived from CLIP-based instance–language matching. This auxiliary segmentation supervision is designed to enhance local text awareness and to stabilise training under limited word-level annotations. The overall training objective is defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{det}}^{\text{DBNet}} + \alpha\, \mathcal{L}_{\text{aux}}^{\text{seg}},$$

where $\mathcal{L}_{\text{det}}^{\text{DBNet}}$ denotes the standard Differentiable Binarization loss employed in DBNet [14], and α is a scalar weight controlling the contribution of the auxiliary segmentation supervision.

The auxiliary segmentation loss supervises a pixel-level text probability map P, produced via CLIP-based instance–language matching, using the same binary ground-truth mask G as the detection branch:

$$\mathcal{L}_{\text{aux}}^{\text{seg}} = \mathcal{L}_{\text{BCE}}(P, G) + \mathcal{L}_{\text{Dice}}(P, G).$$

$\mathcal{L}_{\text{BCE}}$ is the binary cross-entropy loss applied at the pixel level, while $\mathcal{L}_{\text{Dice}}$ is the Dice loss. The Dice term mitigates the severe foreground background imbalance typical of handwritten manuscripts, where text pixels occupy only a small fraction of the page, and encourages coherent and compact text regions. Together, these two terms yield sharper text boundaries and more stable optimisation, particularly in the presence of dense handwriting and limited annotations.

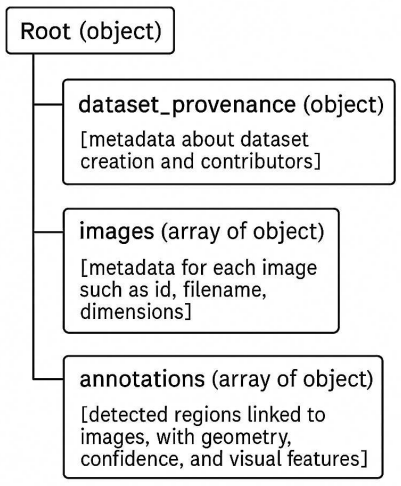


Fig. 10 An illustration of the searchable structure generated by CVC.

Output representation. At inference, the model returns polygonal word instances with associated confidence values. We then apply a simple, deterministic refinement: detections with low confidence are removed; very small candidates are discarded; and instances that substantially overlap with a larger neighbour are suppressed. This process yields non-overlapping, polygonal word instances that adhere closely to handwriting strokes, preserving fine manuscript detail while preventing redundancy and ensuring clean crops for subsequent CVC modules.

4.3 Recognising Word Colour

To keep the system label-efficient, this component is training-free. For each detected word polygon, we crop a tight, isolated word patch and assign two attributes: an exact RGB code and a canonical colour label. This yields a quantitative, searchable colour field in the CVC without adding annotation burden.

Text-pixel selection. After a light preprocessing step to reduce illumination nonuniformity and enhance local contrast, we construct a binarized text map by ensembling the Sauvola, Wolf–Jolion, and Bernsen local thresholders across multiple window sizes and fusing their outputs into a single mask [23, 24, 41]. A brief morphological cleanup followed by connected-component filtering, constrained by estimated stroke width and aspect-ratio checks, removes small specks and paper grain. The foreground set for each word then consists of pixels that lie within its polygon and are marked as text by this map.

Colour RGB code (training-free). From the text-foreground pixels inside each word polygon, we take the per-channel median of the RGB values and record this triplet as the word's colour code. This simple, training-free statistic is outlier-resistant and remains stable under modest illumination drift and residual background bleed.

Colour naming (perceptual matching). We convert each word's RGB code to CIE Lab and assign the nearest colour in a standard web palette (CSS4) using Euclidean distance in Lab space. This yields a human-readable label (e.g., red, navy, darkslategray) paired with the exact numeric code.

This module adds a word-level, quantitative colour attribute using classical, training-free vision. Each word receives an RGB code and a standardised colour label, enabling search, comparison, and future calibration. The design is modular, serving as a reliable baseline and remaining easy to replace when colour annotations or learned models become available.

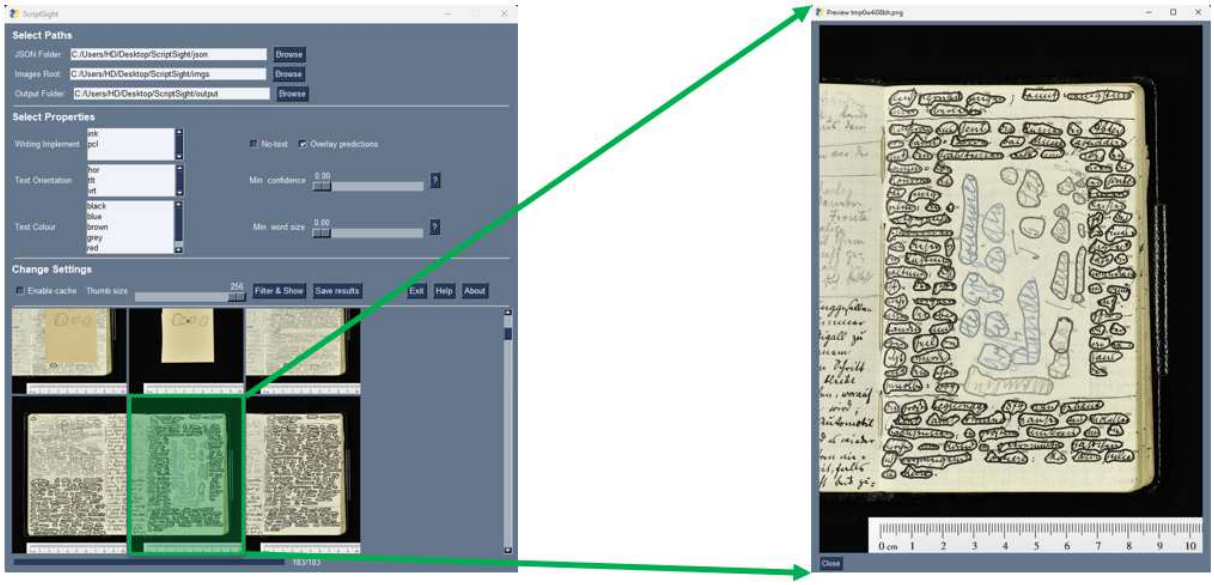


Fig. 11 The GUI of ScriptSight, showing the general layout, the thumbnails displayed on the canvas, and the selected image enlarged in a separate window. A larger version of the image, as well as a short demo, is available at https://www.csmc.uni-hamburg.de/publications/software/scriptsight.html.

4.4 Recognising Visual Attributes via CLIP-Based Classification

We address both orientation and writing implement classification using a CLIP-based visual attribute recognition strategy. Each task is framed as patch-level classification on tightly cropped word instances from the detector (see Fig. 8). Given the limited number of annotations and the subtle visual differences between classes, we leverage frozen CLIP ViT-L/14 features, targeted task-specific augmentations, and automatic classifier selection to ensure robustness and generalisation.

4.4.1 CLIP as a Feature Backbone

We use the CLIP visual encoder in a frozen state to extract high-level embeddings from each word patch. This choice is motivated by CLIP's demonstrated ability to respond to text-like visual patterns, an emergent property of its contrastive vision–language training on large-scale image–text pairs [38]. These embeddings reside in a shared visual–textual space, enabling label-efficient learning for textual attributes without the need for fine-tuning. Fig. 9 illustrates this behavior, showing neuron activations for concepts like "historical," "math," and "star," highlighting CLIP's built-in sensitivity to linguistic structure even in the absence of explicit OCR supervision.

4.4.2 Task-Specific Attribute Training

We apply the same CLIP-based classification framework across different word-level visual attributes. While the overall pipeline remains consistent, each task is distinguished by its specific label set and augmentation policy. To enable robust learning under limited annotations, we combine targeted augmentations with automatic classifier selection.

Orientation: Each word patch is classified as horizontal, vertical, or tilted to support layout-aware analysis in the CVC. We apply geometry-preserving augmentations, such as affine jitter, perspective distortion, and grid occlusion, to simulate realistic deformations while maintaining orientation cues.

Writing Implement: Each patch is also classified as ink or pencil. Since this depends on appearance rather than geometry, we apply light photometric augmentations like blur, brightness shifts, and hue variation to simulate variation in scanning and paper conditions while preserving material differences.

Augmentation and Class Balancing. To address class imbalance, we oversample under-represented classes using additional augmented variants, enhancing diversity without disrupting label semantics.

Automatic Classifier Selection. Each task uses a lightweight MLP classifier head trained atop frozen CLIP embeddings. Rather than hand-designing the head, we search a compact architecture space (varying depth, width, and dropout), evaluating candidates via stratified K-fold validation (default $K = 5$). The model with the highest mean validation accuracy is then retrained on the full dataset for final inference. This approach reduces overfitting and adapts model capacity to the available data.

4.4.3 Inference and Scalability

Once trained, our CLIP-based attribute classifiers are deployed as plug-and-play modules within the CVC pipeline.

Inference and Integration. During inference, each detected word patch is embedded by the frozen CLIP encoder and passed through the selected classifier head. The output is a categorical prediction (e.g., orientation or writing implement) along with a softmax-based confidence score. These predictions are written into the CVC record schema alongside other word-level metadata, enabling consistent search, filtering, and downstream analysis across modules.

Extensibility. The modular pipeline design supports straightforward scaling to additional visual attributes. To add a new classification task, such as identifying non-textual elements or distinguishing word-level styles, only a new set of labels and a lightweight classifier head are required. The frozen CLIP features and existing training pipeline remain unchanged, allowing for rapid prototyping and seamless integration without altering the core system.

4.5 The Computational Catalogue

The Computational Visual Catalogue itself is a COCO-format [42] JSON file consisting of a single root object with three sections: info, images, and annotations, which respectively capture dataset provenance, image metadata, and region-level descriptors. The info section records basic provenance (e.g., description, contributor, version and creation date). The images section lists each page with an identifier, filename and dimensions. The annotations section links detections to images and stores region geometry, confidence, and the visual attributes used for cataloguing, such as colour name and RGB colour code, word orientation, and writing implement. See the generated catalogue for the notebooks of Rainer Maria Rilke in [43]. The schema is intentionally lightweight and extensible: further or alternative visual features can be added as additional fields, and existing values (including scores and labels) can be updated with outputs from more accurate models without disrupting downstream use. See the illustration in Fig. 10.

5 ScriptSight

ScriptSight [44] was developed to enable curators, archivists and researchers to explore and sort page-image collections using Computational Visual Cataloguing (CVC) outputs, by concurrently filtering visual attributes, such as text orientation, writing implement (pencil versus ink) and text colour, previewing matching pages as thumbnails with optional overlays, and saving results in organised, date-stamped folders for review or downstream processing.

Implemented as a lightweight Python application, ScriptSight ingests a directory of CVC catalogues together with a corresponding image root, indexes page-level attributes, and performs concurrent filtering with thresholds on detector confidence and minimum relative word size. Thumbnails are produced in the background, overlays render polygonal text regions and tool labels, configuration is persisted between sessions, and exports encode active filters and the current date to support auditability and reuse.

The interface is a single-window GUI aimed at ease of use: users select paths for JSON, images and output; choose writing implements, orientations and colours via multi-select lists; adjust confidence and word-size thresholds; and toggle "no-text" and overlay modes. Results appear as a scrollable grid of thumbnails; clicking opens a modal preview. Help pop-ups, caching and adjustable thumbnail size further streamline interactive exploration while keeping the workflow flexible for different collections. See Fig. 11.

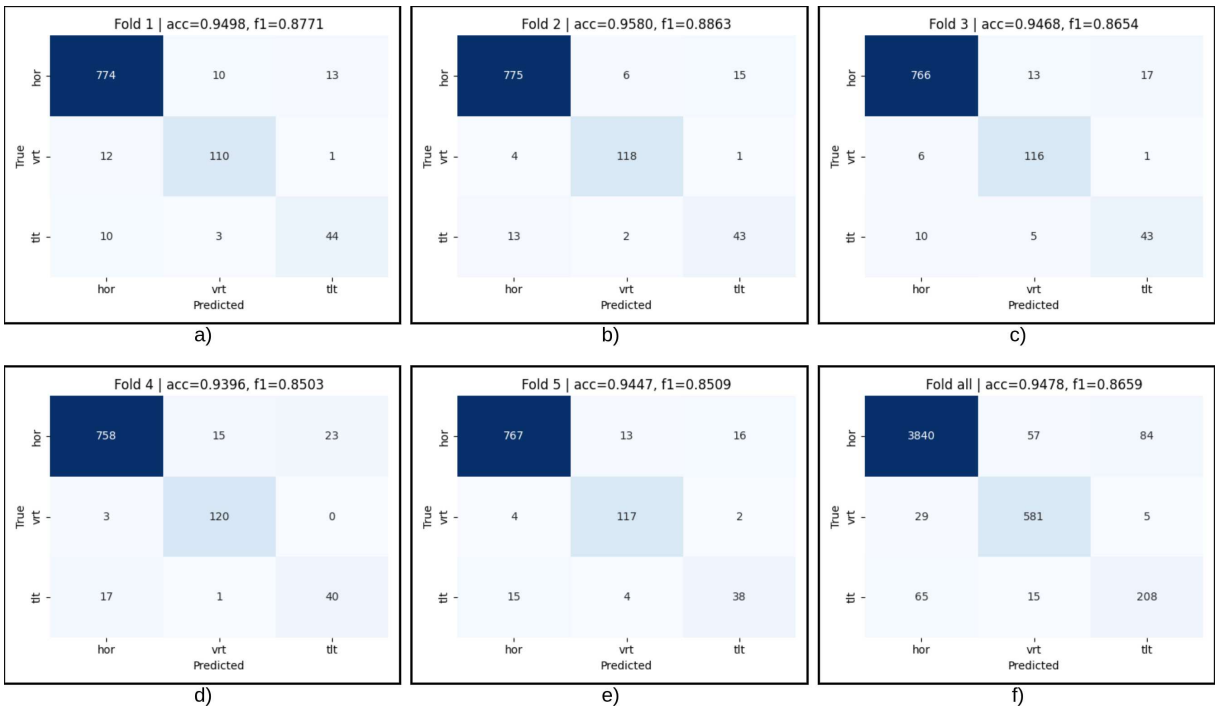


Fig. 12 Confusion matrices of the best-performing MLP classifier (Version 7) built atop frozen CLIP embeddings for the orientation classification module, evaluated on the training set across the five folds of the 5-fold stratified cross-validation and the aggregated result. (a–e) correspond to Folds 1–5, while (f) shows the combined evaluation over all folds. Version 7, which was automatically identified as the best module through the search process based on its highest combined macro F1-score of 0.8659 among all MLP architectures listed in Table 1.

Beyond the GUI, ScriptSight normalises and post-processes CVC attributes before filtering. Colour is not taken from raw RGB strings: each annotation's code is converted to CIE Lab, low-chroma and lightness thresholds identify greyscale, black and white, and other cases are binned by hue into a small, human-readable palette. The interface deliberately ignores per-JSON colour lists and exposes this fixed palette, with optional super-grouping of finer hues, and filtering then operates on these derived labels rather than the original codes. Pages marked "No-text" are determined algorithmically: a minimum detection-confidence threshold is first applied, and only images with zero remaining word annotations are retained. Word-size filtering is relative, computing the largest detected word per page and keeping only words whose area exceeds a user-set fraction of that maximum. Thumbnails and overlays use polygonal segmentations and optional tool labels; a cache and the export directory structure encode the active selections.

Using the tool, one can perform indirect analysis as well, such as determining frequencies and proportions of features, revealing whether phenomena are isolated or patterned. In Rilke's notebooks, for example, the ratio of pencil to ink can be measured and systematically evaluated. An initial analysis indicates that such measurements enable conclusions about Rilke's working practices. One might ask whether there is a relationship between writing direction and writing instrument. Do these correlate with writing locations (on the move vs at the desk) and writing modes (focused writing vs quick notes)? Can the observation that Rilke corrected pencil drafts in ink, using ink within pencil passages as a means of fixation, be systematically confirmed? And how do his practices change over time?

6 Implementation and Results

This section presents our experimental setup, including dataset details, model training strategy,

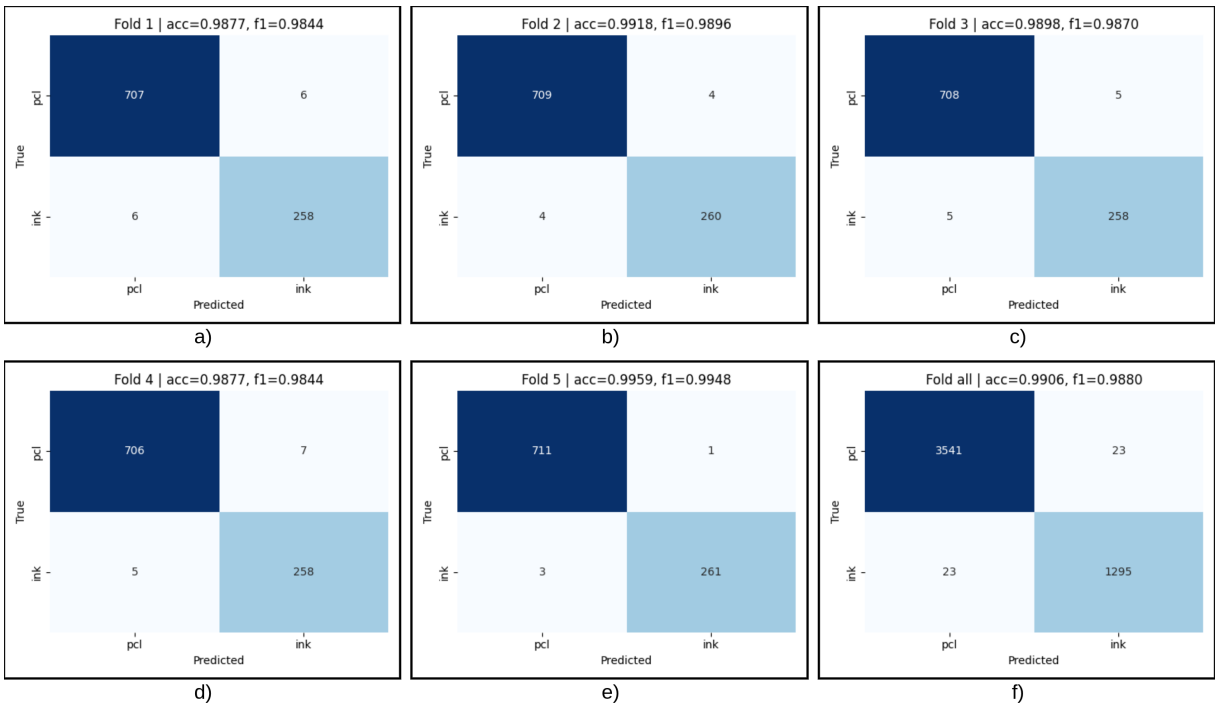


Fig. 13 Confusion matrices of the best-performing MLP classifier (Version 17) built atop frozen CLIP embeddings for the writing implement classification, evaluated on the training set across the five folds of the 5-fold stratified cross-validation and the aggregated result. (a–e) correspond to Folds 1–5, while (f) shows the combined evaluation over all folds. Version 17, which was automatically identified as the best module through the search process based on its highest combined macro F1-score of 0.988 among all MLP architectures listed in Table 1.

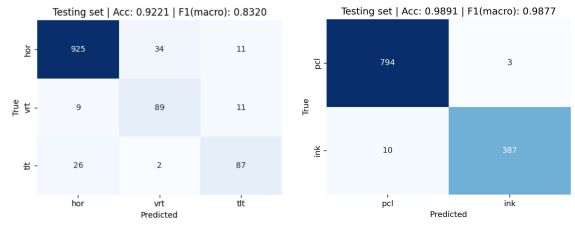


Fig. 14 Confusion matrices of the two attribute classification tasks on the test set. (a) Orientation classification (CLIP+Version 7) achieves an accuracy of 0.9221 and a macro F1-score of 0.8320. (b) Writing implement classification (CLIP+Version 17) achieves an accuracy of 0.9891 and a macro F1-score of 0.9877.

and evaluation methodology. We also report qualitative and quantitative results to demonstrate the effectiveness of the proposed CVC pipeline on historical manuscripts.

6.1 Experimental Setup

We describe the main elements of our experimental setup, including data preparation, pretraining, cross-validation, and hardware. These details support reproducibility and help evaluate each module in the CVC pipeline.

Dataset and Pretraining: We use a curated collection of historical manuscripts annotated at the word level. To adapt the vision backbone to the handwriting domain, we initialise the CLIP ViT-L/14 encoder with weights pretrained on the Imgur5K-Handwriting dataset [45]. The Imgur5K-Handwriting dataset contains approximately 5,000 full-page handwriting images collected from the web. Although it includes images of handwriting relevant to our target domain, the dataset contains sparsely written, in-the-wild text and provides only word-level bounding-box annotations rather than polygonal contours, which limits its suitability for structured manuscript analysis.

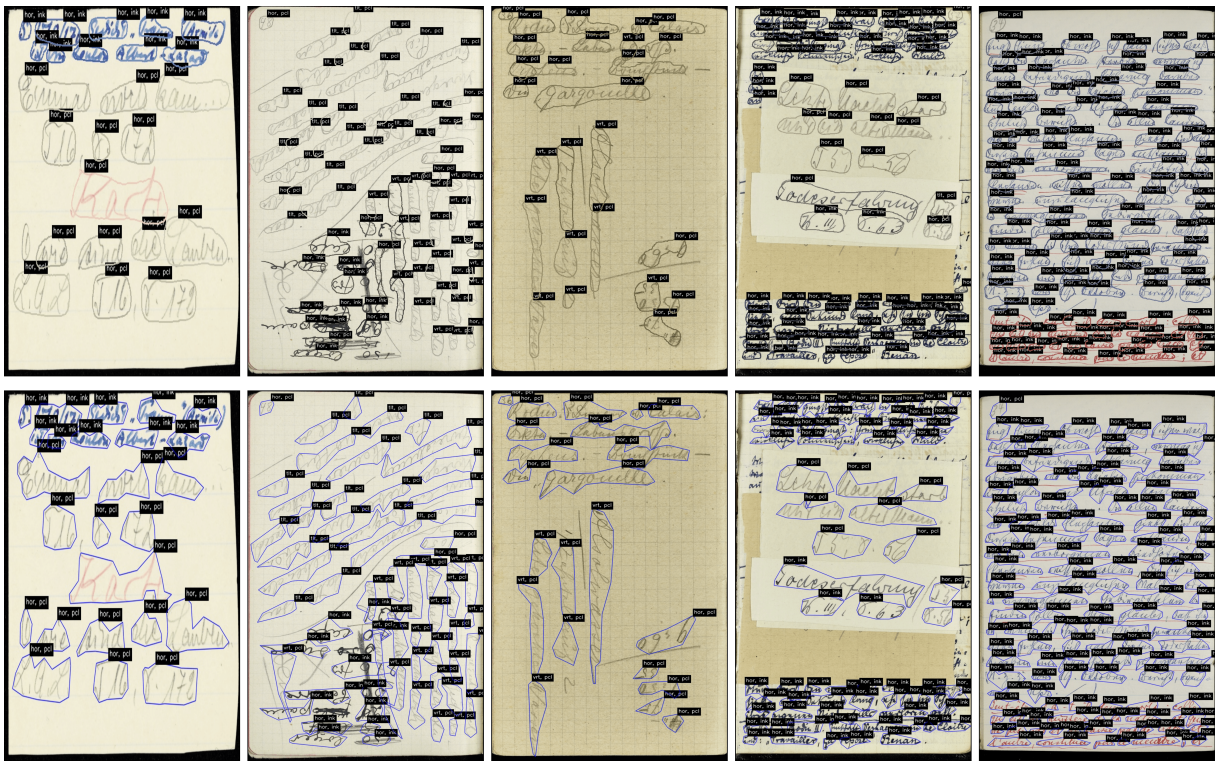


Fig. 15 Comparison between the model predictions and human annotations. The top row shows the outputs generated by our pipeline, while the bottom row displays the corresponding human-annotated ground truth. Predicted orientation and writing implement labels are overlaid on black backgrounds. In the top row, the polygon colors represent the predicted text colors from our color classification module: the closer the predicted color is to the actual word color, the more accurate the prediction. Best viewed with zoom for clarity.

Data Split: Our dataset comprises 100 fully annotated manuscript images. The data are divided into 80% for training and 20% for held-out testing. Both subsets are publicly released to ensure transparency and enable fair comparison across future studies using the same benchmark. For model selection in the orientation and writing implement classification tasks, we apply stratified 5-fold cross-validation on the training set.

6.2 Automatic Classifier Architecture Search

Each attribute recognition module, namely orientation and writing implement classification, employs a lightweight MLP head built atop frozen CLIP embeddings. We perform an automatic architecture search within a compact design space of MLP classifiers by varying their depth, hidden width, and dropout rate, as summarised in Table 1.

A set of manually annotated word-level image patches is prepared for training the attribute modules. Each patch is extracted from polygon-defined text regions and labelled as *horizontal*, *tilted*, or *vertical* for the orientation task, and as *pencil* or *ink* for the writing implement task. These samples are used in a 5-fold stratified cross-validation procedure to evaluate all candidate MLP classifiers.

The best-performing configurations are determined based on the highest combined macro F1-scores across all folds, as summarised in Table 1. The results for the orientation and writing implement modules are shown in Figs. 12 and 13, respectively. The selected MLP heads are subsequently retrained on the fully balanced and augmented training set for final model training. The resulting CLIP+MLP models are then evaluated on the test set, achieving the quantitative performance presented in Fig. 14.

Table 1 Search space of the MLP classifier architectures and their combined macro F1-scores across the two attribute recognition tasks. Each module classifier is built atop frozen CLIP embeddings, differing only in their output dimensionality: three classes for the orientation task (Ori) and two classes for the writing implement task (Tool). The best-performing configurations for each task are highlighted in bold.

ID	Depth	Widths	Drop	BN	F1 (Ori)	F1 (Tool)
V01	0	-	-	No	0.5167	0.9352
V02	1	128	-	No	0.7935	0.9767
V03	1	256	-	No	0.8195	0.9776
V04	1	256	0.2	No	0.8139	0.9801
V05	2	512-256	0.3	No	0.8425	0.9824
V06	2	512-256	0.4	No	0.8465	0.9822
V07	2	512-256	0.3	Yes	**0.8659**	0.9850
V08	2	1024-512	0.4	No	0.8469	0.9847
V09	3	1024-512-256	0.4	Yes	0.8594	0.9868
V10	4	1024-512-256-256	0.5	No	0.8444	0.9814
V11	2	128-128	0.2	No	0.8002	0.9809
V12	2	160-160	0.2	Yes	0.8170	0.9819
V13	2	192-192	0.2	No	0.8283	0.9815
V14	2	224-224	0.2	No	0.8360	0.9822
V15	2	256-256	0.2	No	0.8302	0.9794
V16	2	256-256	0.2	Yes	0.8629	0.9828
V17	3	192-192-192	0.2	Yes	0.8598	**0.9880**
V18	3	224-224-224	0.3	No	0.8303	0.9830
V19	4	160-160-160-160	0.3	Yes	0.8312	0.9825
V20	3	256-256-256	0.4	Yes	0.8452	0.9848

6.3 Ablation Studies for Word Detection

To evaluate the effectiveness of key components in our word detection model, we conduct a series of ablation studies on the DLA-RMR dataset. These experiments examine how various architectural and training choices influence detection performance, measured by the polygon-level F1-score.

Effect of Freezing the Encoders. We first analyse the impact of fine-tuning versus freezing the CLIP encoders. Specifically, we adjust the learning rate (LR) factors for the image and text encoders. As shown in Table 2, freezing the text encoder ($LR = 0.0$) yields the best overall performance. This confirms that retaining the pretrained text-language prior of CLIP is beneficial for historical manuscript data, while moderate adaptation of the image encoder preserves visual discriminability without overfitting.

Table 2 Ablation study of the image and text encoder learning rate (LR) factors. Freezing the text encoder leads to the highest F1-score.

Image Encoder LR	Text Encoder LR	F1
0.1	0.0	**0.879**
0.1	0.1	0.865

Effect of Prompt Components. We further examine the influence of different prompt designs within the CLIP-based word detection framework. Using only a predefined prompt (PP) with the default phrase "*text pixels in a historical manuscript*" slightly improves performance. Combining it with a learnable prompt (LP) brings additional gains, while the introduction of a language prompt generator (LG) and a visual prompt generator (VG) produces the most notable improvements, raising the F1-score by 0.8% and 1.8%, respectively (Table 3). The VG module proves particularly effective in the historical domain, as it dynamically generates page-specific prompts, improving contextual understanding in scarce-data scenarios.

Table 3 Ablation results for different prompt configurations in the word detection model. Here, "PP", "LP", "LG", and "VG" refer to the predefined prompt, learnable prompt, language-guided prompt generator, and visual-guided prompt generator, respectively.

Method	PP	LP	LG	VG	F1
Word Detection	✓	✗	✗	✗	0.851
	✓	✓	✗	✗	0.853
	✓	✓	✓	✗	0.861
	✓	✓	✓	✓	**0.879**

Effect of the Auxiliary Word-Segmentation Branch. Finally, we investigate the contribution of the auxiliary word-segmentation branch, which refines local boundaries and stabilises optimisation under limited annotation. As shown in Table 4, the auxiliary task significantly improves performance, demonstrating that it helps the pretrained CLIP image encoder better perceive localised text regions. The best result is achieved with $\lambda = 1$, confirming the importance of this branch.

Table 4 Ablation study of the auxiliary word-segmentation branch.

Auxiliary Weight (λ)	F1
0	0.845
0.5	0.862
1.0	**0.879**

6.4 Qualitative Results

To complement the quantitative evaluation, we present qualitative end-to-end results from the CVC pipeline. These examples illustrate how the individual components jointly contribute to robust and interpretable manuscript analysis. For each manuscript page, the detected word polygons, predicted colours, and attribute classifications are visualised, demonstrating the system's effectiveness across diverse page layouts with changing orientations, varied writing implements, and different colour usages, as shown in Fig. 15.

Word Detection: Tight, stroke-aligned polygons accurately isolate individual words even in densely written layouts.

Colour Estimation: The median RGB values of binarised text pixels are combined with perceptual CSS4 colour matching to produce stable and interpretable colour labels. This procedure is entirely training-free, as no annotated ground-truth colours are available. Consequently, the evaluation is qualitative: predicted polygon edge colours (representing model outputs) are visually compared with the actual text colours, allowing for intuitive assessment of prediction accuracy.

Attribute Recognition: Orientation and writing implement classifiers deliver consistent and reliable predictions across varying layouts and material conditions, demonstrating the robustness and generalisation ability of the proposed modules.

6.5 Failure Case Analysis

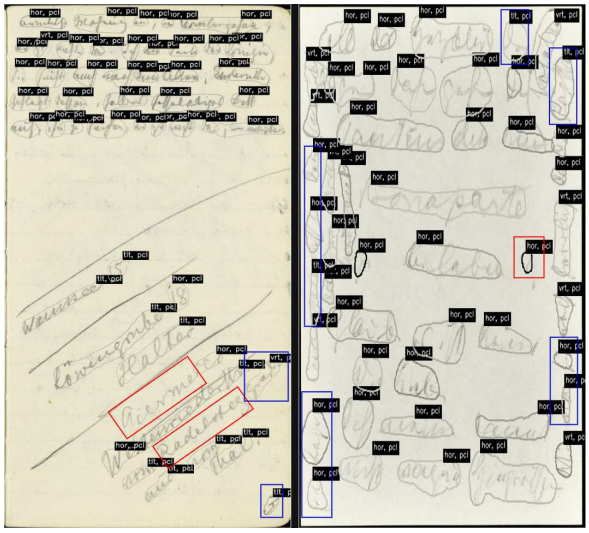


Fig. 16 Failure cases of the proposed pipeline. Red boxes denote text regions that are missed or incorrectly detected by the text detection module. Blue boxes indicate incorrect predictions in word orientation classification. Best viewed with zoom for clarity.

As shown in Figure 16, we illustrate representative failure cases of the proposed pipeline, focusing on the two most challenging components: word detection and orientation classification. Missed or inaccurately detected text regions are highlighted with red boxes, while blue boxes indicate incorrect predictions in word orientation classification. For text detection, false positives typically occur when ink traces, stains, or non-textual marks are erroneously recognised as text, whereas false negatives often arise when elongated or connected word structures are only partially detected, resulting in fragmented word instances. Orientation errors mainly appear in cases where local word structures are visually ambiguous or when the word orientation lies near the boundary between different orientation categories, making reliable discrimination difficult. These examples reflect inherent challenges posed by dense handwriting and complex manuscript layouts.

7 Discussion and Conclusion

We have presented Computational Visual Cataloguing (CVC), a computer-vision pipeline designed to complement expert cataloguing. It extracts page-anchored, identifies content-bearing pages, localises cross-script words with their orientation, and captures colour and writing-implement cues before writing them to structured, searchable catalogue fields. The system comprises an extensible framework and schema that can accommodate additional attributes, task-specific methods and models for each stage of the pipeline. In addition, ScriptSight is developed as an interactive tool that allows curators, archivists and researchers to explore, filter and export the system's outputs for practical use. Using the notebooks of Rainer Maria Rilke as a challenging, culturally significant use case, we have reported quantitative results for individual components and qualitative assessment of the integrated system by domain experts, demonstrating applicability to complex pages and usability for exploratory cataloguing at scale. All models, source code, catalogues and software are publicly released, and a video demonstration is available to guide software usage.

From both archival and literary perspectives, the current analysis of Rilke's notebooks is highly significant. On the archival level, the results show the potential to make large collections accessible, far beyond the individual cataloguing of single textual witnesses typical of analogue methods. Automated detection of writing instrument, writing direction, and ink colour generates quantitative data that are difficult to obtain through individual autopsy and are rarely recorded as catalogue metadata during conventional, item-by-item description.

Further characteristics would also be valuable for cataloguing and study, notably automated detection of language use. Which languages does he employ, and how do shifts occur? Initial examinations suggest the notebooks are remarkable for the way in which Rilke acquired languages in the sense of 'soft multilingualism', learning individual terms and codes [46]. In German and French, he sometimes switches mid-sentence, confirming his functional bilingualism. Detecting such patterns and relationships across the entire corpus would help to confirm impressions formed through page-by-page reading.

Acknowledgements. The research for this work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2176 'Understanding Written Artefacts: Material, Interaction and Transmission in Manuscript Cultures', project no. 390893796. The research was conducted within the scope of the Centre for the Study of Manuscript Cultures (CSMC) at Universität Hamburg.

In addition, we would like to thank our colleagues from the CSMC, José Maksimczuk and Konrad Hirschler, for their insightful discussions, and Hui Xu for her help with data preparation.

Declarations

- Funding
- Conflict of interest/Competing interests (check journal-specific guidelines for which heading to use)
- Ethics approval and consent to participate
- Consent for publication
- Data availability
- Materials availability
- Code availability
- Author contribution

If any of the sections are not relevant to your manuscript, please include the heading and write 'Not applicable' for that section.

References

[1] Muehlberger, G., Seaward, L., Terras, M., Ares Oliveira, S., Bosch, V., Bryan, M., Colutto, S., Déjean, H., Diem, M., Fiel, S., *et al.*: Transforming scholarship in the archives through handwritten text recognition: Transkribus as a case study. Journal of documentation **75**(5), 954–976 (2019)

[2] Kiessling, B., Stokes, P.A., contributors: eScriptorium Documentation. Project documentation (2024). https://escriptorium.readthedocs.io/

[3] Kiessling, B., contributors: kraken — OCR/HTR system optimised for historical and non-Latin scripts. Project documentation (2024). https://kraken.re/

[4] Zant, T., Schomaker, L., Haak, K.: Handwritten-word spotting using biologically inspired features. Ieee transactions on pattern analysis and machine intelligence **30**(11), 1945–1957 (2008)

[5] Schomaker, L.: The birth of a massive search engine for historical and multi-cultural handwritten collections. CERL Seminar Presentation, Oslo (2014)

[6] Würsch, M., Ingold, R., Liwicki, M.: Divaservices—a restful web service for document image analysis methods. Digital Scholarship in the Humanities **32**(suppl_1), 150–156 (2017)

[7] Pelagios Network: Recogito: Semantic Annotation Platform for Texts and Images. Software/Platform (2024). https://recogito.pelagios.org/

[8] Oliveira, S.A., Seguin, B., Kaplan, F.: dhsegment: A generic deep-learning approach for document segmentation, 7–12 (2018). IEEE

[9] Tensmeyer, C., Davis, B., Wigington, C., Lee, I., Barrett, B.: Pagenet: Page boundary extraction in historical handwritten documents, 59–64 (2017)

[10] Stamatopoulos, N., Gatos, B., Georgiou, T.: Page frame detection for double page document images. In: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, pp. 401–408 (2010)

[11] Baek, Y., Lee, B., Han, D., Yun, S., Lee, H.: Character region awareness for text detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9365–9374 (2019)

[12] Wang, W., Xie, E., Li, X., Hou, W., Lu, T., Yu, G., Shao, S.: Shape robust text detection with progressive scale expansion network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9336–9345 (2019)

[13] Wang, W., Xie, E., Song, X., Zang, Y., Wang, W., Lu, T., Yu, G., Shen, C.: Efficient and accurate arbitrary-shaped text detection with pixel aggregation network, 8440–8449 (2019)

[14] Liao, M., Wan, Z., Yao, C., Chen, K., Bai, X.: Real-time scene text detection with differentiable binarization. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 11474–11481 (2020)

[15] Liu, Y., Chen, H., Shen, C., He, T., Jin, L., Wang, L.: Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9809–9818 (2020)

[16] Monnier, T., Aubry, M.: docextractor: An off-the-shelf historical document element extraction. In: 2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 91–96 (2020). IEEE

[17] Long, S., Qin, S., Panteleev, D., Bissacco, A., Fujii, Y., Raptis, M.: Towards end-to-end unified scene text detection and layout analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1049–1059 (2022)

[18] Nayef, N., Patel, Y., Busta, M., Chowdhury, P.N., Karatzas, D., Khlif, W., Matas, J., Pal, U., Burie, J.-C., Liu, C.-l., et al.: Icdar2019 robust reading challenge on multi-lingual scene text detection and recognition—rrc-mlt-2019, 1582–1587 (2019). IEEE

[19] Nayef, N., Patel, Y., Bušta, M., Chowdhury, P.N., Karatzas, D., Khlif, W., Matas, J., Pal, U., Burie, J.-C., Liu, C.-L., Ogier, J.-M.: Icdar2019 robust reading challenge on multi-lingual scene text detection and recognition (rrc-mlt-2019). In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1582–1587 (2019). https://doi.org/10.1109/ICDAR.2019.00254

[20] Li, Z., Lin, Y., Chiang, Y.-Y., Weinman, J., Tual, S., Chazalon, J., Perret, J., Duménieu, B., Abadie, N.: Icdar 2024 competition on historical map text detection, recognition, and linking. In: International Conference on Document Analysis and Recognition, pp. 363–380

(2024). Springer

[21] Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. International Journal of Computer Vision **130**(9), 2337–2348 (2022) https://doi.org/10.1007/s11263-022-01653-1

[22] Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16816–16825 (2022)

[23] Sauvola, J., Pietikäinen, M.: Adaptive document image binarization. Pattern recognition **33**(2), 225–236 (2000)

[24] Wolf, C., Jolion, J.-M., Chassaing, F.: Text localization, enhancement and binarization in multimedia documents. In: 2002 International Conference on Pattern Recognition, vol. 2, pp. 1037–1040 (2002). IEEE

[25] Bernsen, J.: Dynamic thresholding of grey-level images. Technical report, Philips Electronics N.V. (1986). Technical report

[26] Andronache, I., Liritzis, I., Jelinek, H.F.: Fractal algorithms and rgb image processing in scribal and ink identification on an 1819 secret initiation manuscript to the "philike hetaereia". Scientific Reports **13**(1), 1735 (2023)

[27] Ezcurra, M., Góngora, J.M., Maguregui, I., Alonso, R.: Analytical methods for dating modern writing instrument inks on paper. Forensic Science International **197**(1-3), 1–20 (2010)

[28] Golovkina, A.G., Karpukhin, O.R., Kravchenko, A.V., Khairullina, E.M., Tumkin, I.I., Kalinichev, A.V.: Digital color analysis and machine learning for ballpoint pen ink clustering and aging investigation. Forensic Science International **364**, 112236 (2024)

[29] e.V., D.S., Marbach, D.L.: Deutsches Literaturarchiv Marbach (2025). https://www.dla-marbach.de/ Accessed 2025-11-05

[30] Nationalbibliothek: Startseite (2024). https://edition.onb.ac.at/context:hnb Accessed 2025-11-05

[31] McGillen, P.S.: Der Fontane Workshop. Realismus-Manufaktur Im Zeitalter der Druckmaschinen. Fontaneana, vol. 19. Königshausen & Neumann, Würzburg (2023). https://nbn-resolving.org/urn:nbn:de:101:1-202311192010502744794 7

[32] Richter, S.: Die Aufzeichnungen des Rainer Maria Rilke. Notizen, 'Skizzenbücher', 'Tagebücher', 'Taschenbücher': Materiale Artefakte und Interpretationsaufgabe. Jahrbuch der Deutschen Schillergesellschaft: Internationales Organ für Neue Deutsche Literatur **67**, 287–302 (2023) https://doi.org/10.46500/83535512-013

[33] e.V., D.S., Marbach, D.L.: Kallías – der online Katalog des Deutschen Literaturarchivs Marbach (2025). https://www.dla-marbach.de/find/opac Accessed 2025-11-05

[34] e.V., D.S., Marbach, D.L.: A: Rilke-Archiv Gernsbach/Notizbücher T1–T45 (2025). https://www.dla-marbach.de/find/opac/id/BF00048063/ Accessed 2025-11-05

[35] Marotta, G., Fedeli, A., Sathiyamani, S., Colini, C.: Archaeometric philology for the study of deteriorated and overlapping layers of ink: the colour code of an early qur'anic fragment. The European Physical Journal Plus **140**(5), 446 (2025)

[36] Mohammed, H., Dang, Q.-V.: The DLA-RMR dataset: Annotated subset of RMR notebooks for CVC development (2025). https://doi.org/10.25592/uhhfdm.17931

[37] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., *et al.*: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning, pp. 8748–8763 (2021). PmLR

[38] Goh, G., Cammarata, N., Voss, C., Carter, S., Petrov, M., Schubert, L., Radford, A., Olah, C.: Multimodal neurons in artificial neural networks. Distill **6**(3), 30 (2021)

[39] Yu, W., Liu, Y., Hua, W., Jiang, D., Ren, B., Bai, X.: Turning a clip model into a scene text detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6978–6988 (2023)

[40] YOLOv11, R.K.M.H.: An overview of the key architectural enhancements. arXiv: 241017725, 2024.(1. Shanghai Ocean University, Shanghai 201306, China **2** (2024)

[41] Bernse, J.: Dynamic thresholding of grey-level images. In: Proc. 8th Int. Conf. on Pattern Recognition, 1986, pp. 1251–1255 (1986)

[42] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision, pp. 740–755 (2014). Springer

[43] Dang, Q.-V., Mohammed, H.: Computational Visual Catalogue (CVC) - Rilke's notebooks - Full CVCs (2025). https://doi.org/10.25592/uhhfdm.17809

[44] Mohammed, H.: ScriptSight v1.5 (2025). https://doi.org/10.25592/uhhfdm.17615

[45] Krishnan, P., Kovvuri, R., Pang, G., Vassilev, B., Hassner, T.: Textstylebrush: transfer of text aesthetics from a single example. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(7), 9122–9134 (2023)

[46] Noorani, Y.: Hard and soft multilingualism. Critical Multilingualism Studies **1**(2), 7–28 (2013). Accessed 2025-11-05