Establishing a Document Layout Analysis Baseline for Historical Cipher Keys

Raphaela Heil
Stockholm University
Sweden
raphaela.heil@ling.su.se

Alicia Fornés
Universitat Autònoma de Barcelona
Spain
afornes@cvc.uab.es

Benedek Láng
Eötvös Loránd University
Hungary
lang.benedek@gtk.elte.hu

Beáta Megyesi
Stockholm University
Sweden
beata.megyesi@ling.su.se

Abstract

Historical cipher keys encode mappings between plaintext elements and cipher symbols and are characterized by complex, heterogeneous handwritten layouts. This paper establishes a baseline for document layout analysis (DLA) of historical cipher keys using a newly annotated dataset of 350 images from European archives dating from ca. 1300 to 1850 CE. We evaluate four YOLO-based architectures under three conditions: training from scratch, cross-domain transfer from models pre-trained on DocLayNet and CATMuS in a class-agnostic setting, and fine-tuning of these pre-trained models on cipher key data. Results show that training from scratch is limited by data scarcity and unstable convergence, while direct transfer across DLA domains performs poorly. In contrast, fine-tuning consistently improves performance across all architectures, demonstrating the feasibility of adapting existing DLA models to cipher keys and supporting downstream tasks such as key extraction and comparative cryptographic analysis.

1 Introduction

A historical cipher key is a document that specifies the correspondence between plaintext elements and their encoded representations used in a cipher system. Such keys were widely used in diplomatic, military, and administrative contexts to enable the encryption and decryption of sensitive correspondence. Unlike ciphertexts, which consist mostly of encoded symbols with or without cleartext, cipher keys typically combine explanatory text, structured mappings, and graphical elements, resulting in complex and highly variable document layouts. Figure 1 shows two representative examples of historical cipher keys from Europe.

At a structural level, cipher keys most commonly contain one or more *alphabets*, which define the mapping between individual letters (or letter combinations) and their corresponding cipher symbols. In addition, many keys include *nomenclature elements*, which extend the alphabet by assigning codes to higher-level linguistic units such as names, places, titles, syllables, or frequently used words and phrases. These mappings are often arranged in structured forms such as tables, lists, or aligned rows and columns, but their exact visual realization varies across documents and time periods (Megyesi et al., 2024).

Cipher keys frequently exhibit heterogeneous layouts that combine multiple organizational principles within a single page. Alphabet and nomenclature sections may coexist with running text, marginal annotations, figures, decorative elements, stamps, or watermarks. Furthermore, mappings can be arranged horizontally or vertically, may involve one-to-one, one-to-many, or many-to-one relationships, and are often written by hand with limited visual consistency. This structural and visual diversity distinguishes cipher keys from both standard textual documents and more regular tabular material.

Figure 1: Examples of historical cipher keys from the Swedish National Archives. Left: DECODE record 4323, right: DECODE record 4329.

The automatic analysis of cipher keys therefore poses challenges that go beyond traditional handwritten text recognition. Before individual symbols or mappings can be transcribed or interpreted, semantically relevant regions of the page must first be identified and localized. Document layout analysis (DLA) addresses this need by segmenting cipher key pages into meaningful regions, making it a crucial preprocessing step for downstream tasks such as symbol recognition, key extraction, comparative cryptographic analysis, and assisted decryption.

On a general level, DLA identifies regions of interest, such as text blocks or figures, in document images, which can then be processed further, for example by applying handwritten text recognition (HTR) to text lines, or by performing structured information extraction from tables. In the specific case of cipher keys, the regions detected through DLA are directly tied to cryptographic interpretation. For example, identifying alphabet and nomenclature regions enables the extraction of plaintext–cipher symbol pairs, which can be used for decryption attempts or for comparing symbol inventories across different keys and potentially related ciphertexts. Beyond individual documents, automatic region identification also supports large-scale comparative studies, such as analyses of nomenclatures across entire collections (see e.g. Megyesi et al., 2024).

Given the relevance of layout analysis for such downstream tasks, this work aims to establish a baseline for document layout analysis of historical cipher keys. To this end, we compare the performance of four YOLO-based architectures (Redmon et al., 2016) under different training and fine-tuning conditions, using a newly established dataset of historical cipher keys. Concretely, we investigate the following questions:

1. How well do YOLO-based models perform on cipher key DLA, when trained *from scratch* on a comparatively small dataset?

2. How well do YOLO-based models, pretrained on other DLA domains, namely PDF documents and medieval manuscripts, perform in identifying regions of interest in cipher key images, in a class-agnostic setting?

3. Can the performance of models trained from scratch be improved by fine-tuning pretrained models with the cipher key dataset?

4. Can the performance for cipher key-specific classes, such as alphabet and nomenclature key regions, be further improved by limiting the training data?

2 Related Work

Historical cipher keys and their internal structure have been studied primarily from a cryptological and historical perspective. Large-scale analyses of European cipher keys have shown that most keys combine alphabet mappings with nomenclature sections that encode higher-level linguistic units such as names, places, titles, words, or phrases, often supplemented with nulls and special symbols (Megyesi et al., 2024). These mappings are typically organized in tables, lists, or aligned rows and columns, but exhibit substantial variation in layout, orientation, and complexity across time periods and regions. Studies of key usage and instructional material further highlight that cipher keys frequently include explanatory text and operational rules, which contributes to heterogeneous

page layouts (Láng et al., 2025). From a computational perspective, prior work has discussed the challenges of automatically extracting structured mappings from cipher keys and emphasized the need to first identify key components such as alphabet and nomenclature regions (Tudor et al., 2020). In addition, corpus-building efforts such as the DECODE project have provided standardized terminology and large collections of cipher keys and ciphertexts, enabling systematic comparative analysis (Megyesi et al., 2019). However, despite this growing body of work, the automatic layout analysis of cipher keys has received little attention, motivating the present study.

2.1 Document Layout Analysis

A significant body of literature examines DLA for various types of documents, ranging from printed corporate forms, to ancient handwritten documents. Binmakhashen and Mahmoud (2019) present a summary of DLA works until ca. 2019.

Recent works explore various kinds of deep learning approaches, such as the cross-attention-based HookNet, proposed by Wu et al. (2025).

Additionally, YOLO-based approaches have for example been examined in the context of printed documents (Li et al., 2025) and historical, Greek dictionaries (Ioakeimidou et al., 2024).

Besides this, several HTR frameworks, such as Loghi (van Koert et al., 2024; Klut et al., 2023) and Kraken (Kiessling, 2026), incorporate DLA approaches into their pre-processing pipeline, to identify textual regions which are then segmented into individual lines.

We are not aware of any prior works that study DLA for historical cipher keys.

2.2 DLA Datasets

Various DLA datasets have been presented in the literature, both as stand-alone projects and in the context of competitions. These datasets are often focused on a single domain or are limited to images of a specific document type (file format, content, layout).

The two datasets that form the basis for some of the pre-trained models used in this work are DocLayNet (Pfitzmann et al., 2022), consisting of PDF pages, i.e. contemporary digital material, and CATMuS (Clérice et al., 2024), focusing on medieval handwritten pages.

Other DLA datasets, which focus on a similar time frame, but different domain compared to our work, are for example the HORAE dataset (Boillet et al., 2019), containing images of prayer books from the late Middle Ages, the DIVA-HisDB (Simistira et al., 2016), a collection of challenging medieval documents, and SAM (Zottin et al., 2024), the ICDAR 2024 Competition on Few-Shot and Many-Shot Layout Segmentation of Ancient Manuscripts.

To the best of our knowledge, no prior dataset for DLA of historical cipher keys exists.

3 Study Design

The following sections briefly outline the different components of our study design, concluding with a description of the conducted experiments.

3.1 Data

The experiments in this work are based on a collection of 350 images of, predominantly handwritten, cipher keys from archives in Europe, sourced from the DECODE database (Héder and Megyesi, 2022). The original documents are estimated to have been created between ca. 1300 CE and ca. 1850 CE, with the bulk of the data originating from ca. 1500 CE to 1700 CE. The keys are constructed using various combinations of letters, numbers and graphical symbols.

During the digitisation process, the cipher key images were arranged as records, grouping documents that stem from the same book or collection. Each record has been annotated with several metadata fields, e.g. pertaining to age and provenance, as far as it could be established.

3.1.1 Data Annotation

All images were annotated manually by one annotator, with the support of several experts, when uncertainties arose. The annotation scheme was specifically designed for the use case of cipher keys, and their application in downstream recognition tasks and analyses. We follow the terminology established by (Mikhalev et al., 2023), resulting in the following cipher key-related regions, as well as a number of more general DLA classes. An annotation example is shown in Figure 2.

Alphabet Key Describes which alphabet element, including double letters, belongs to which alphabet code element(s). Note that this does not have to be a one-to-one mapping, and some (or all) alphabet elements may be represented by several alphabet code elements.

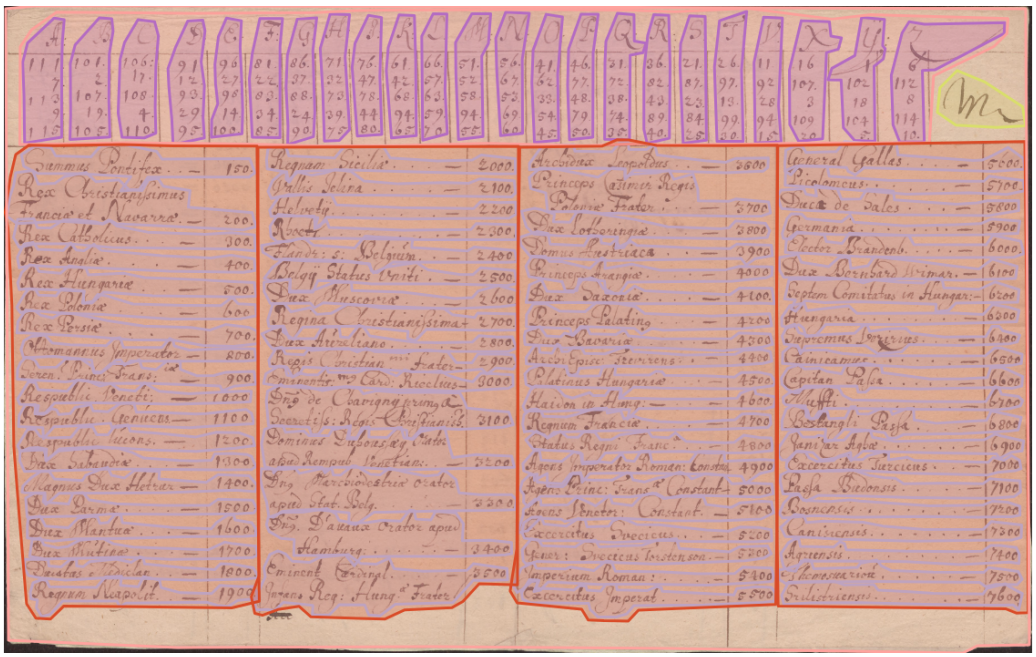


Figure 2: Sample of a cipher key, annotated with an alphabet key region, containing key columns, at the top, followed by four nomenclature key columns, containing key rows. Original image source: Swedish National Archives, DECODE record 4180.

Nomenclature Key Describes which nomenclature element belongs to which nomenclature code element(s). A nomenclature element is anything "larger" than an alphabet element, i.e. it can be a syllable, a name, a function, a content word, as well as a phrase.

Key Row Indicates a horizontal entry in a cipher key, i.e. a combination of a plaintext element and its corresponding code element, placed horizontally next to each other. Any order of plaintext and code is possible and mappings may appear in any form (i.e. one-to-one, one-to-many, many-to-one, many-to-many).

Key Column The vertical counterpart to key rows, i.e. a combination of a plaintext element and its corresponding code element, placed vertically above/below each other. Apart from the orientation, key rows and key columns do not differ in their structure.

Operational Element Groups various operational elements, such as nullities, nullifiers, duplication signs, and punctuation.

Text Any kind of textual area that does not fall into one of the other categories. This can for example be running text (paragraphs), headlines, page numbers, or annotations (marginalia).

Figure Any kind of drawing, illustration or figure that is not explicitly of textual nature. This may also include decorations and "doodles".

Table Any kind of table that is **not** a form of cipher key.

Stamp A short text, sometimes with a logo, stamped onto the page, for example indicating which archive indexed a given document.

Watermark Any form of watermark, either physically applied during the paper production process, or digitally added after the digitisation.

Page Indicates the borders of a page. This is primarily intended to delineate the actual page content from noise, stemming from the digitisation surface (e.g. table) and pages (or other artefacts) in the background. Images, showing a page spread, are annotated with two separate bounding boxes, unless the content runs seamlessly across the binding.

Fragment Marks the rare case in which a given image contains a page fragment, i.e. a piece of paper, torn off from the current, or another, page.

Other A catch-all label for any content that cannot be definitively assigned to any of the aforementioned categories, primarily designed to ac-

count for unforeseen annotation cases and to mark areas that should be reviewed by an expert.

3.1.2 Data Splitting

The 350 images were split into 50 pages for testing and 300 pages for 5-fold cross-validation, i.e. 60 images per validation fold. To reduce the risk of data leakage and biases, the original record-level grouping was maintained during the data splitting. Images from any given record always belong to the same subset and are never split across several. Additionally, a number of individual images were grouped into *virtual* records, as their metadata indicated that they were created and/or used by the same person(s). These virtual records were treated the same as all other records, i.e. never split across several subsets.

Besides this, an attempt was made to balance the cross-validation and test sets with respect to age and combination of symbol sets. However, due to the constraint of not splitting records, as well as some uncertainty in the recorded metadata, a perfectly balanced split cannot be guaranteed. Figure 3 summarises the age and symbol set distributions across the cross-validation and test sets. Note that individual folds within the cross-validation set were primarily balanced by the number of images, and a balanced distribution regarding other metadata fields therefore cannot be guaranteed.

3.2 Model Selection

Four different versions of the deep neural network architecture "YOLO" (stemming from the phrase "You only look once") (Redmon et al., 2016) were selected for evaluation, primarily motivated by the public availability of pre-trained DLA models.

Concretely, three different sizes (medium, large, extra large) of the YOLO *detection* architecture, version 11, and the extra large implementation of the YOLO *segmentation* architecture, version 8, were chosen. Both the medium-sized detection (Brunello, 2025), and the segmentation model (Yoann Schneider, 2024) were pre-trained on DocLayNet (Pfitzmann et al., 2022), adhering to the dataset's standard labels. The large and extra large versions of YOLO v11 (Mattingly, 2025) were pre-trained on CATMuS (Clérice et al., 2024), employing a subset of the SegmOnto (Gabay et al., 2024) vocabulary. Table 1 summarises the classes, covered by the two datasets, in comparison to ours. Detailed descriptions of each class can be obtained from the respective publications. However, even without these, the names should provide sufficient indications that these datasets differ in domain and only marginally overlap with the tasks investigated in this paper. We chose these pre-trained models despite these discrepancies because they are closer to the domain of cipher keys than for example YOLO models, pre-trained on the widely used COCO dataset (Lin et al., 2014), which consists of images of everyday scenes, containing for example animals and kitchen utensils. Besides this, smaller mismatches between domains and vocabularies reflect the reality of developing models for specialised collections, for which no perfectly fitting pre-trained model exists.

In addition to the pre-trained versions of the aforementioned models, we also evaluate the same architectures when trained from scratch. Table 2 summarises the different models, pre-training conditions and parameter sizes, and specifies the name by which each of them will be referred to for the remainder of this paper. Generally, names consisting only of a version and size refer to the models trained from scratch, while those containing a dataset name refer to their pre-trained, respectively fine-tuned, counterparts.

3.3 Evaluation

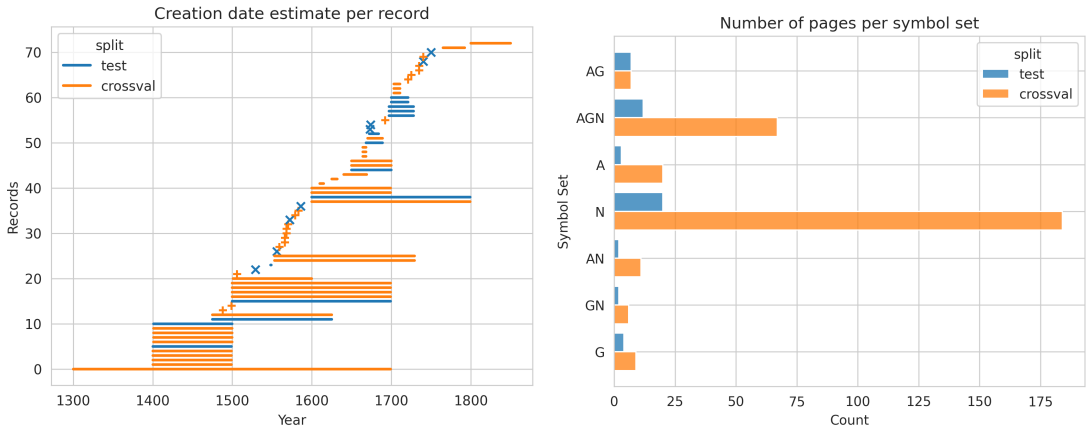
Unless otherwise stated, all models are trained and evaluated on all previously introduced classes, except for "other", which is excluded, due to its volatile nature. Prediction performance is reported as mean average precision, averaged over intersection over union thresholds, ranging from 50% to 95%, with a step-size of 5%. We use the standard implementation provided by the Ultralytics framework (Jocher et al., 2023b). Where applicable, test performances are summarised across the five fold-based models and standard deviations are reported in parentheses.

3.4 Experiments

Overall, we have constructed three major and two minor experiments to investigate the aforementioned research questions. Following the description of the general experiment protocol, each experiment is briefly introduced below.

3.4.1 General Experiment Protocol

All experiments are based on the Ultralytics YOLO implementations (Jocher et al., 2023a; Jocher and Qiu, 2024) and follow their default



(a) Lines indicate ranges of creation date estimates, e.g. the top-most record is estimated to have been created between 1800 and 1850 CE. Symbols (+ and x) mark records for which concrete years of creation have been determined.

(b) A = alphabet, G = graphic signs, N = numerical; combinations of letters indicate combinations of symbols sets, e.g. AGN = alphabet + graphic signs + numerical symbols

Figure 3: Creation date estimate per record (a) and distribution of symbol sets (b)

Table 1: Annotation classes used by the respective dataset, sorted by their dataset-specific IDs.

Dataset	Classes
Cipher Keys	Alphabet Key, Nomenclature Key, Figure, Table, Text, Stamp, Watermark, Other, Key Row, Key Column, Page, Fragment, Operational Element
DocLayNet (Pfitzmann et al., 2022)	Caption, Footnote, Formula, List-Item, Page-footer, Page-header, Picture, Section-header, Table, Text, Title
CATMuS (Clérice et al., 2024)	MarginTextZone, DefaultLine, MainZone, RunningTitleZone, NumberingZone, QuireMarksZone, HeadingLine, DropCapitalZone, StampZone, GraphicZone, InterlinearLine, DigitizationArtefactZone, DropCapitalLine, DamageZone, MusicLine, TitlePageZone, SealZone, MusicZone

parameters for learning rate, optimiser selection, etc., including the use of RandAugment (Cubuk et al., 2020) for augmentations. Modifications to the standard protocol were made regarding the batch size (8), to adapt to available GPUs (NVIDIA Tesla T4, NVIDIA Tesla A40). Training was terminated with a patience of 50 epochs, or when the wall time exceeded four hours, whichever condition was reached first. We follow the pre-trained models' defaults regarding input image sizes, i.e. 640px for 11l and 11x-based models, 1024px for 8x, and 1280px for 11m. Due to memory constraints, evaluations are limited to 200 detections per image.

3.4.2 Experiment 1: Training Models from Scratch

In a first instance, each of the selected models is trained from scratch, using each of the five cross-validation folds, i.e. yielding five distinct checkpoints. Each model is evaluated on the test set and the averaged performance is reported.

3.4.3 Experiment 2: Class-agnostic Evaluation of Pre-trained Models

Each of the four pre-trained models is evaluated on the test set, while disregarding class labels, i.e. assuming all detected and ground truth regions are of the same class. Class labels are discarded in this step, as the pre-trained models employ vocabularies that are distinctly different from that in our data and no meaningful mapping across all relevant categories could be established. As reference, we also evaluate the models from the previous experiment, i.e. those trained from scratch, under the same class-agnostic conditions.

3.4.4 Experiment 3: Fine-tuning of Pre-Trained Models

The four pre-trained models from experiment 1 are fine-tuned on the cross-validation data, following the aforementioned protocol. The obtained mod-

Table 2: Summary of models, considered in this work.

Name	Data Foundation	Architecture	Param Count
YOLO11m	Cipher Keys	YOLO11m	20M
doclaynet-11m	DocLayNet		
cipher-11m	DocLayNet → Cipher Keys		
YOLO11l	Cipher Keys	YOLO11l	25M
catmus-11l	CATMuS		
cipher-11l	CATMuS → Cipher Keys		
YOLO11x	Cipher Keys	YOLO11x	56M
catmus-11x	CATMuS		
cipher-11x	CATMuS → Cipher Keys		
YOLO8x-seg	Cipher Keys	YOLO8x-seg	71M
doclaynet-8x-seg	DocLayNet		
cipher-8x-seg	DocLayNet → Cipher Keys		

els are evaluated both in the class-based setting, for comparison with the results from the initial experiment, and in a class-agnostic fashion, in relation to experiment 2.

3.4.5 Experiment 4: Cipher Key-specific Modifications

The impact of the following modifications is examined, to determine whether a closer focus on selected classes can improve the prediction performance for these regions of interest:

1. limiting the fine-tuning to alphabet and nomenclature regions;

2. limiting the fine-tuning to classes of immediate relevance to cipher keys analysis, i.e. alphabet and nomenclature keys, cipher key rows and columns, operational elements, as well as the general page extent;

The outlined modifications are applied both to the cross-validation sets, and during evaluation, to the test set.

Table 3: Prediction performance across the four models, trained from scratch.

Model	mAP50-95
YOLO11m	0.1921 (± 0.11)
YOLO11l	0.1050 (± 0.14)
YOLO11x	0.1388 (± 0.13)
YOLO8x-seg	0.2046 (± 0.02)

4 Results and Discussion

4.1 Experiment 1: Training Models From Scratch

Table 3 summarises the performance of the four architectures, trained from scratch. All models exhibit low prediction performances, with mAPs below 0.21. As indicated by the comparably high standard deviations, all three YOLO11 architectures displayed stability issues during training, with about one third of the configurations failing to converge. Despite outperforming all other models in this experiment, and exhibiting much more stable training behaviour, the performances of the YOLO8 models remain low. Both observations, i.e. unstable training and low overall performance, can be explained by the limited amount of training data, which is insufficient to train the examined DLA models from scratch.

4.2 Experiment 2: Class-agnostic Evaluation of Pre-trained Models

The results for the pre-trained models, shown in Table 4, indicate that neither of the two DLA domains are directly transferable to cipher key regions. However, the two CATMuS-based models slightly outperform the ones pre-trained on DocLayNet, which may stem from some overlap in appearance between the medieval documents and the cipher key images, some of which also originate from the Middle Ages.

The models, trained from scratch, consistently outperform the pre-trained models, demonstrating some adaptation to the cipher key domain, despite the overall low class-based performance, as dis-

cussed in the previous section.

Table 4: Class-agnostic prediction performance of pre-trained models and their counterparts, trained from scratch.

Model	mAP50-95
doclaynet-11m	0.0672
catmus-11l	0.0705
catmus-11x	0.0822
doclaynet-8x-seg	0.0499
YOLO11m	0.3881 (± 0.21)
YOLO11l	0.1654 (± 0.22)
YOLO11x	0.2147 (± 0.19)
YOLO8x-seg	0.3987 (± 0.01)

4.3 Experiment 3: Fine-tuning of Pre-trained Models

Table 5 and Table 6 summarise the prediction performances of the fine-tuned models. As can be seen when comparing the results with those from the two previous experiments, all fine-tuned models consistently outperform their counterparts by considerable margins, across both evaluation modalities. These results highlight the efficacy of fine-tuning and the resulting successful domain transfer, both from PDFs and medieval manuscripts. In contrast to the models, trained from scratch, the fine-tuning process yields stable results across all five folds, with no further convergence issues being observed.

Considering the class-level performances, it can be summarised that nomenclature keys and key rows consistently achieve higher prediction performances (mAP50-95 of 0.53-0.66, respectively 0.57-0.65) than their counterparts, alphabet keys and key columns (mAP50-95 of 0.36-0.41, respectively 0.42-0.55). A straightforward explanation for this can be found in the imbalanced distribution within the two pairs, with the former appearing four to six times more frequently than the respective latter classes.

Regarding classes with low mAP scores, figures and operational elements stand out, with performances below 0.02. Overall, these results can be explained by the low number of occurrences. However, for the latter class, its high level of visual variation may be a contributing factor. These variations stem from the underlying annotation scheme. *Operational elements* summarise several sub-categories, such as nullities and duplication signs, which may appear in several different forms, for example in the shape of tables, i.e. similar to alphabet/nomenclature keys, or as running text. In order to properly handle these diversities, a different approach will have to be found, such as explicitly defining and using the respective subcategories. However, a closer investigation of this is beyond the scope of this work.

Table 5: Prediction performance of fine-tuned models, summarised across all classes.

Model	mAP50-95
cipher-11m	0.4114 (± 0.03)
cipher-11l	0.4031 (± 0.05)
cipher-11x	0.4102 (± 0.04)
cipher-8x-seg	0.3757 (± 0.01)

Table 6: Class-agnostic prediction performance of fine-tuned models.

Model	mAP50-95
cipher-11m	0.5884 (± 0.02)
cipher-11l	0.5157 (± 0.03)
cipher-11x	0.5440 (± 0.02)
cipher-8x-seg	0.5500 (± 0.02)

Table 7: Prediction performance (mAP50-95) for the models, fine-tuned only on alphabet and nomenclature key regions (*limited*), and their counterparts, trained on all regions, with the evaluation limited to alphabet and nomenclature key regions (*original*).

Model	Limited	Original
cipher-11m	0.4851 (± 0.02)	0.4849 (± 0.02)
cipher-11l	0.4095 (± 0.06)	0.4704 (± 0.05)
cipher-11x	0.4095 (± 0.04)	0.4611 (± 0.03)
cipher-8x-seg	0.5525 (± 0.03)	0.5385 (± 0.02)

4.4 Experiment 4: Cipher Key-specific Modifications

The following subsections briefly summarise the results and analyses for several smaller experiments, all pertaining to cipher key-specific modifications, i.e. disregarding classes that are not of relevance to (parts of) the cipher key analysis, such as watermarks and stamps.

4.4.1 Limiting Fine-tuning to Alphabet and Nomenclature Key Regions

Table 7 presents the performance differences between models, trained only on alphabet and nomenclature key regions, and those trained on all available labels. Both original cipher-11l and cipher-11x architectures considerably outperform their limited counterparts. For the two models, based on DocLayNet (cipher-11m and cipher-8x-seg), limiting the training data did improve the overall performance, albeit by a very modest margin. Given that other classes in the dataset are also of relevance for downstream analyses, the performance gain is not large enough to justify the limited training approach. It may, however, be of interest as part of a pipeline that approaches the segmentation hierarchically, i.e. identifying alphabet/nomenclature regions before applying a secondary model for the subsegmentation of rows/columns.

4.4.2 Limiting Fine-tuning to Cipher Key-relevant Classes

Table 8 summarises the performance differences between models, trained only on the classes "alphabet key" and "nomenclature key", "key row" and "key column", "operational element" and "page", and their counterparts trained on all labels but evaluated only on the selected class-subset. The prediction performances differ only marginally, which can likely be explained by the fact that the selected classes, with the exception of operational elements, are well-represented in this focused cipher key dataset, and are therefore expected to perform well, regardless. While the excluded classes may not have immediately obvious downstream use cases, their presence clearly does not hinder the training. It is therefore not necessary to exclude them, and they can be maintained to allow for a more complete view of the data in subsequent processing steps.

5 Qualitative Analysis

In order to complement the extensive quantitative analyses of the previous sections, we present a brief qualitative analysis of the DLA results. All presented annotations were obtained from the best-performing checkpoint of experiment 3.

Figures 4 and 5 exemplify two typical failure cases, in which the DLA models identify non-textual artefacts as regions of interest. This generally pertains to blank areas, page edges, or parts of the digitisation surface (e.g. table surface) that are included in the document image. While the latter two can be easily removed by cropping the image, the former requires adaptations during model training, e.g. by including more diverse samples of watermarks.

Table 8: Prediction performance (mAP50-95) for the models, fine-tuned only on the classes alphabet and nomenclature keys, cipher key rows and columns, operational elements and page (*limited*) and their counterparts, trained on all regions, with the evaluation limited to the same classes (*original*).

Model	Limited	Original
cipher-11m	0.4492 (± 0.02)	0.4537 (± 0.01)
cipher-11l	0.4589 (± 0.03)	0.4497 (± 0.03)
cipher-11x	0.4540 (± 0.01)	0.4617 (± 0.02)
cipher-8x-seg	0.4857 (± 0.01)	0.4834 (± 0.01)

Finally, Figure 6 shows a successful example of DLA. While minor artefacts remain, alphabet and nomenclature regions, as well as the contained key columns, respectively rows, are generally correctly identified and delineated.

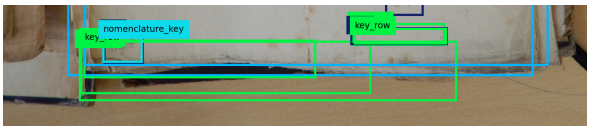


Figure 4: Example for edge artefacts mistakenly being identified as regions of interest.

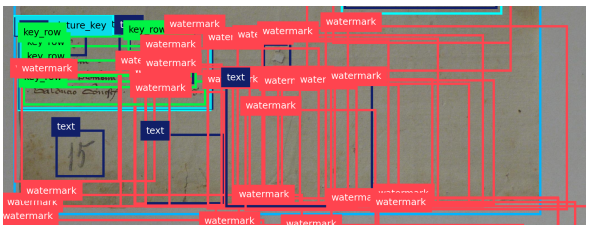


Figure 5: Example of overexpression of watermark regions (red) in empty areas.

6 Conclusion

In this work, we have studied four YOLO-based models in order to establish a baseline for DLA of cipher keys. We have demonstrated that:

1. a small dataset of 240 training images (per

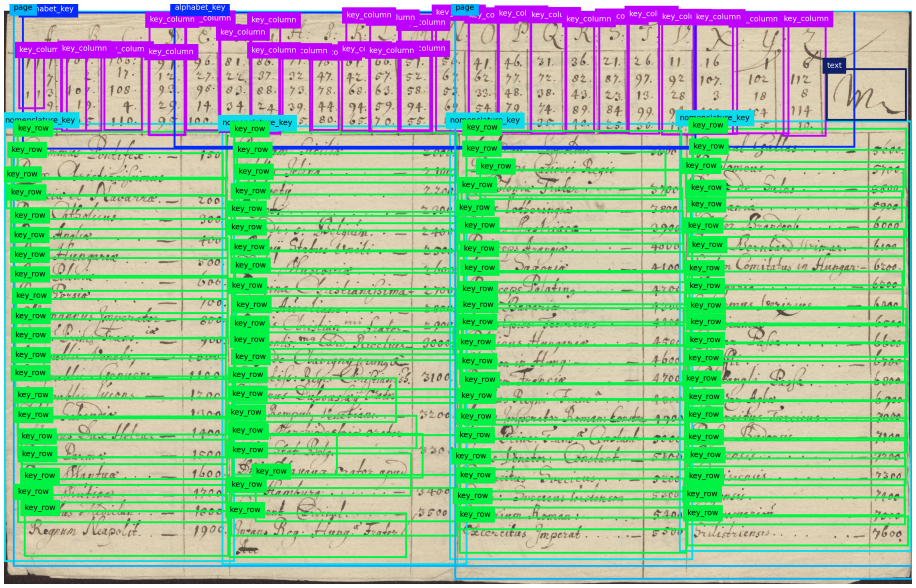


Figure 6: Sample output of the best-performing model from experiment 3. Ground truth shown in Figure 2.

fold) is not sufficient to train the selected architectures from scratch.

2. models, pre-trained on other DLA domains, are not directly applicable to our dataset, even when used in a class-agnostic fashion.

3. considerable improvements can be obtained by fine-tuning pre-trained DLA models, adapting them to the cipher key domain. Improvements are obtained consistently, across all pre-trained configurations, despite a small training set, and a considerable domain gap, between the pre-training and fine-tuning datasets.

4. the two examined training modifications did not yield sufficiently large or consistent improvements. Overall, neither of them can therefore be recommended for implementation.

Overall, the evaluated models only achieved a moderate mean average precision on our newly introduced cipher key dataset. Besides the small amount of data, the difference in performance, compared to state-of-the-art DLA models, can also be explained by the kinds of information that we are looking to extract from cipher key images. In contrast to conventional DLA, which focuses on physical aspects of the layout, our cipher key annotation scheme requires a certain level of semantic analysis.

Even though the presented work is limited to YOLO-based models, it presents a first baseline for DLA of cipher keys. Future work should expand these investigations to other model architectures. In addition to this, an extension of the annotations, to more images, either genuine or through the creation of synthetic samples, may be of interest. As an alternative to increasing the dataset size, few-shot approaches could be explored.

Data Availability

Due to image copyright constraints, the dataset cannot be shared publicly at the time of writing. All trained models are available in the following repository: 10.5281/zenodo.19911174.

Acknowledgements

This work has been supported by Riksbankens Jubileumsfond, grant M24-0028: Echoes of History: Analysis and Decipherment of Historical Writings (DESCRYPT). The computations were enabled by resources provided by the National Academic Infrastructure for Supercomputing in Sweden (NAISS), partially funded by the Swedish Research Council through grant agreement no. 2022-06725. Alicia Fornés acknowledges financial support for her general research activities from ICREA under the ICREA Academia (Departament de Recerca i Universitats de la Generalitat de Catalunya), and also, she has partial support from the Spanish project PID2024-157778OB-I00

(SUKIDI) from the Ministerio de Ciencia e Innovación, the Departament de Cultura of the Generalitat de Catalunya, and the CERCA Program / Generalitat de Catalunya. We gratefully acknowledge Adelaida López for her help in annotating the data.

References

Galal M. Binmakhashen and Sabri A. Mahmoud. 2019. Document Layout Analysis: A Comprehensive Survey. *ACM Comput. Surv.*, 52(6), October.

Mélodie Boillet, Marie-Laurence Bonhomme, Dominique Stutzmann, and Christopher Kermorvant. 2019. HORAE: an annotated dataset of books of hours. In *Proceedings of the 5th International Workshop on Historical Document Imaging and Processing*, HIP '19, pages 7–12. Association for Computing Machinery.

Alessandro Brunello. 2025. Armaggheddon/yolo11-document-layout (Hugging Face). `https://huggingface.co/Armaggheddon/yolo11-document-layout`, version 0a03ab4.

Thibault Clérice, Ariane Pinche, Malamatenia Vlachou-Efstathiou, Alix Chagué, Jean-Baptiste Camps, Matthias Gille Levenson, Olivier Brisville-Fertin, Federico Boschetti, Franz Fischer, Michael Gervers, Agnès Boutreux, Avery Manton, Simon Gabay, Patricia O'Connor, Wouter Haverals, Mike Kestemont, Caroline Vandyck, and Benjamin Kiessling. 2024. CATMuS Medieval: A Multilingual Large-Scale Cross-Century Dataset in Latin Script for Handwritten Text Recognition and Beyond. In *Document Analysis and Recognition - ICDAR 2024*, pages 174–194. Springer Nature Switzerland.

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. 2020. Randaugment: Practical Automated Data Augmentation With a Reduced Search Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June.

Simon Gabay, Ariane Pinche, Kelly Christensen, and Jean-Baptiste Camps. 2024. SegmOnto: A Controlled Vocabulary to Describe and Process Digital Facsimiles. *Journal of Data Mining & Digital Humanities*, Dec.

Mihály Héder and Beáta Megyesi. 2022. The DECODE Database of Historical Ciphers and Keys: Version 2. In *Proceedings of the 5th International Conference on Historical Cryptology, HistoCrypt 2022*, pages 111–114, Linköping University Electronic Press.

Despoina Ioakeimidou, Stavros N. Moutsis, Konstantinos Evangelidis, Konstantinos A. Tsintotas, Panagiotis E. Nastou, Elpida Perdiki, Emmanouil Gkinidis, Nikos Tsoukatos, Antonis Tsolomitis, Maria Konstantinidou, and Stamatios Busses. 2024. Cyril's Lexicon Layout Analysis Through Deep Learning. In *2024 IEEE International Conference on Imaging Systems and Techniques (IST)*, pages 1–6.

Glenn Jocher and Jing Qiu. 2024. Ultralytics YOLO11. `https://github.com/ultralytics/ultralytics`, version 11.0.0.

Glenn Jocher, Ayush Chaurasia, and Jing Qiu. 2023a. Ultralytics YOLOv8. `https://github.com/ultralytics/ultralytics`, version 8.0.0.

Glenn Jocher, Jing Qiu, and Ayush Chaurasia. 2023b. Ultralytics YOLO, January. `https://github.com/ultralytics/ultralytics`.

Benjamin Kiessling. 2026. Version 5 of the Kraken ATR Engine for the Humanities. In *Document Analysis and Recognition – ICDAR 2025*, pages 443–458. Springer Nature Switzerland.

Stefan Klut, Rutger van Koert, and Ronald Sluijter. 2023. Laypa: A Novel Framework for Applying Segmentation Networks to Historical Documents. In *Proceedings of the 7th International Workshop on Historical Document Imaging and Processing*, HIP '23, pages 67–72. Association for Computing Machinery.

Dong-Lin Li, Shih-Kai Lee, and Yin-Ting Liu. 2025. Printed document layout analysis and optical character recognition system based on deep learning. *Scientific Reports*, 15(1):23761, Jul.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Computer Vision – ECCV 2014*, pages 740–755. Springer International Publishing.

Benedek Láng, Beáta Megyesi, Nils Kopal, Vasily Mikhalev, Crina Tudor, and Michelle Waldispühl. 2025. Cipher key instructions in early modern Europe: analysis and text edition. *Cryptologia*, 49(5):416–442.

William Mattingly. 2025. biglam/medieval-manuscript-yolov11 (Hugging Face). `https://huggingface.co/biglam/medieval-manuscript-yolov11`, version 8f5eddd.

Beáta Megyesi, Nils Blomqvist, and Eva Pettersson. 2019. The DECODE Database: Collection of Historical Ciphers and Keys. In *Proceedings of the 2nd International Conference on Historical Cryptology, HistoCrypt 2019*, pages 69–78, Linköping University Electronic Press.

Beáta Megyesi, Crina Tudor, Benedek Láng, Anna Lehofer, Nils Kopal, Karl de Leeuw, and Michelle Waldispühl. 2024. Keys with nomenclatures in the early modern europe. *Cryptologia*, 48(2):97–139.

Vasily Mikhalev, Nils Kopal, Bernhard Esslinger, Michelle Waldispühl, Benedek Láng, and Beáta Megyesi. 2023. What is the Code for the Code? Historical Cryptology Terminology. In *Proceedings of the 6th International Conference on Historical Cryptology, HistoCrypt 2023*, pages 130–138, Linköping University Electronic Press.

Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S. Nassar, and Peter Staar. 2022. DocLayNet: A Large Human-Annotated Dataset for Document-Layout Segmentation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, pages 3743–3751. Association for Computing Machinery.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June.

Foteini Simistira, Mathias Seuret, Nicole Eichenberger, Angelika Garz, Marcus Liwicki, and Rolf Ingold. 2016. DIVA-HisDB: A Precisely Annotated Large Dataset of Challenging Medieval Manuscripts. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 471–476.

Crina Tudor, Beáta Megyesi, and Benedek Láng. 2020. Automatic Key Structure Extraction. In *Proceedings of the 3rd International Conference on Historical Cryptology, HistoCrypt20*, pages 146–152, Linköping University Electronic Press.

Rutger van Koert, Stefan Klut, Tim Koornstra, Martijn Maas, and Luke Peters. 2024. Loghi: An End-to-End Framework for Making Historical Documents Machine-Readable. In *Document Analysis and Recognition – ICDAR 2024 Workshops*, pages 73–88. Springer Nature Switzerland.

Fei Wu, Mathias Seuret, Martin Mayr, Florian Kordon, Jochen Zöllner, Sebastian Wind, Andreas Maier, and Vincent Christlein. 2025. Lightweight cross-attention-based HookNet for historical handwritten document layout analysis. *International Journal on Document Analysis and Recognition (IJDAR)*, 28(3):409–427, Sep.

TEKLIA Yoann Schneider. 2024. yolo-v8-segmenter/DocLayNet. `https://gitlab.teklia.com/dla/models/-/tree/master/yolo-v8-segmenter/DocLayNet`, version 5d3d0478.

Silvia Zottin, Axel De Nardin, Gian Luca Foresti, Emanuela Colombi, and Claudio Piciarelli. 2024. ICDAR 2024 Competition on Few-Shot and Many-Shot Layout Segmentation of Ancient Manuscripts (SAM). In *Document Analysis and Recognition - ICDAR 2024 Proceedings, Part VI*, pages 315–331. Springer-Verlag.