# Synthetic Data Augmentation for Hierarchical Object Detection in Historical Handwritten Substitution Cipher Keys

Stanislav Marochok[1], Ondrej Páleník[1]

[1]Slovak University of Technology in Bratislava, Faculty of Electrical Engineering and Information Technology, Institute of Computer Science and Mathematics, Ilkovicova 3, 812 19 Bratislava, Slovakia

**Abstract**

Historical handwritten cipher keys are difficult material for object detection. They are visually heterogeneous, structurally dense, and rarely available with manual annotations. This paper evaluates whether rule-based synthetic data improves hierarchical object detection in historical handwritten substitution cipher keys. The task uses three annotation levels: individual plaintext or ciphertext elements, plaintext–ciphertext pairs, and larger document sections. A Python generator was developed to create synthetic cipher-key images with automatic bounding-box annotations. The generator models alphabetic substitutions, homophonic substitutions, n-grams, nulls, and dictionary codes. It also simulates selected document properties, including aged paper, stains, ink drops, bleed-through, mild perspective distortion, uneven lighting, compression artefacts, and handwriting variation. The evaluation uses YOLOv11m and compares synthetic-only, larger synthetic-only, real-only, combined real and synthetic, and larger combined datasets on the same real test set of eight manually annotated historical pages. Synthetic-only training does not outperform real-only training. The best result is obtained by combining real and synthetic data. The larger combined dataset reaches AP50:95 = 0.5884 and AP50 = 0.9166, improving over real-only training by 0.0610 AP50:95 and 0.0501 AP50. The largest gain is observed for pair detection, where AP50:95 increases from 0.5278 to 0.6480.

**Keywords**

synthetic data, historical cryptology, nomenclator keys, object detection, YOLOv11, document analysis, data augmentation

CEUR-WS.org/Vol-4323/paper43.pdf

## 1. Introduction

Historical cipher keys are primary sources for the study of early modern cryptographic practice. Nomenclator keys are especially relevant because they combine several mechanisms on the same page: alphabetic substitutions, homophonic alternatives, n-grams, null symbols, and dictionary codes for words, names, or phrases. The layouts are dense and irregular. Rows, columns, separators, handwriting style, ink intensity, page degradation, and acquisition conditions vary across archives and individual documents [1, 2, 3].

Automatic processing of this material requires more than optical character recognition. A digitization pipeline must first identify the structural units on the page and recover their relations. Previous work defined a three-level detection task for handwritten substitution cipher keys: elements, plaintext–ciphertext pairs, and sections [4]. Elements represent individual plaintext or ciphertext tokens. Pairs represent local correspondences between plaintext units and code values. Sections represent larger blocks, tables, or columns. The pair level is central because it contains the cryptographic relation needed for later transcription, mapping, and comparison with encrypted text.

The motivation for automatic processing is not only to digitize large collections of cipher keys. These documents are also interesting from a document-analysis perspective because their layout contains important information about the cryptographic structure. Detecting elements, plaintext–ciphertext pairs, and sections makes it possible to study whether this structure can be identified automatically. Such methods could later support transcription, reconstruction of cipher systems, comparison of keys, and analysis of larger archival collections.

*ITAT'26: Information Technologies – Applications and Theory, September 25–29, 2026, Biele Karpaty, Slovakia*

stanislav.marochok@stuba.sk (S. Marochok); xpaleniko@stuba.sk (O. Páleník)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-11-09

The practical obstacle is the lack of annotated data. Manual bounding-box annotation is slow because one page can contain hundreds or thousands of small objects. Annotation decisions are also non-trivial. Elements are densely written, pairs may contain multiple codes, and section boundaries are sometimes defined by whitespace rather than visible lines. Synthetic data generation offers a possible response. If generated pages approximate the visual and structural properties that matter for detection, they can increase the amount of training data without proportional manual annotation effort. The relevant question is empirical: plausible generated pages do not automatically improve detection on real historical documents.

This paper evaluates a rule-based generator of synthetic annotated nomenclator-key images. The generator was developed as a configurable software tool for producing historical-looking cipher-key pages and annotations [5]. Here, the generator is evaluated as a source of training data for hierarchical object detection. The study keeps the detector and real test set fixed and measures the effect of synthetic data. The contribution is threefold: the paper describes a rule-based generator for synthetic historical handwritten cipher keys, reports a controlled low-resource experiment with synthetic-only, real-only, and combined training datasets, and analyzes where synthetic augmentation helps or remains limited.

The target documents are not treated as ordinary handwritten pages. A nomenclator key is a structured cryptographic object. Its visual layout encodes relations between plaintext units and ciphertext values. Nomenclators usually combine alphabet substitutions, homophonic alternatives, n-grams, nulls, and dictionary codes in one system [1, 2, 3]. These parts are often arranged as separate blocks, columns, or tables. Their boundaries can be marked by lines, spacing, headings, or only by a change in the local layout [2, 1].

This structure motivates the three annotation levels used in this work. The element level marks individual written tokens, such as plaintext letters, syllables, words, numbers, or cipher symbols. The pair level groups elements that form one cryptographic correspondence. A pair may contain one plaintext unit and one ciphertext value, but it may also contain several homophonic values assigned to the same plaintext unit. The section level marks larger functional regions of the key, such as alphabet substitutions, n-gram tables, null-symbol areas, or dictionary-code blocks. This three-level formulation follows the hierarchical detection task previously defined for handwritten substitution cipher keys [4].

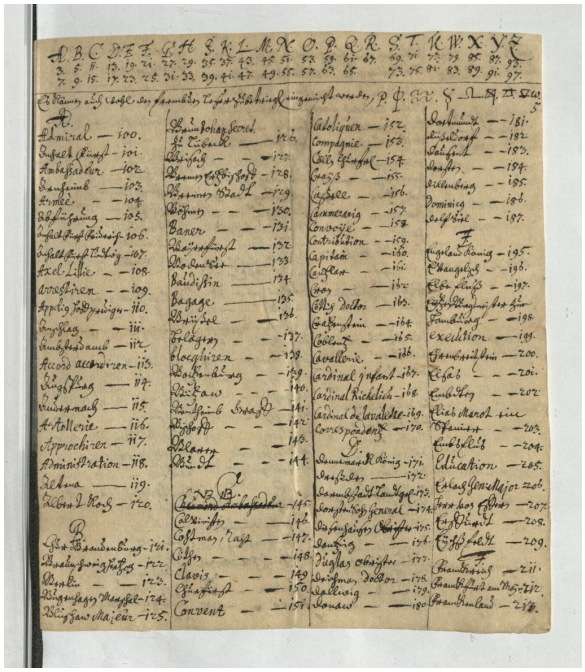
The levels are hierarchical, but not equally difficult to detect. Elements are small and dense, so their detection is sensitive to handwriting quality, touching symbols, and page degradation. Pairs depend on spatial relations between nearby elements. This makes the task harder than token detection, because the detector must approximate a semantic relation expressed through layout. Sections are larger and visually more stable, but their borders are often ambiguous when the historical key has no ruled table or explicit separator. The evaluation therefore reports the three levels separately [4].

For synthetic generation, this hierarchy is important because annotations can be produced directly from the rendering process. The generator first creates the functional content of the key, then places the content into sections, pairs, and elements. This makes it possible to export complete ground truth without manual annotation. The generated hierarchy also follows the downstream use case: elements support token-level detection, pairs support plaintext–ciphertext mapping, and sections support page organization and reading-order reconstruction [5, 4].

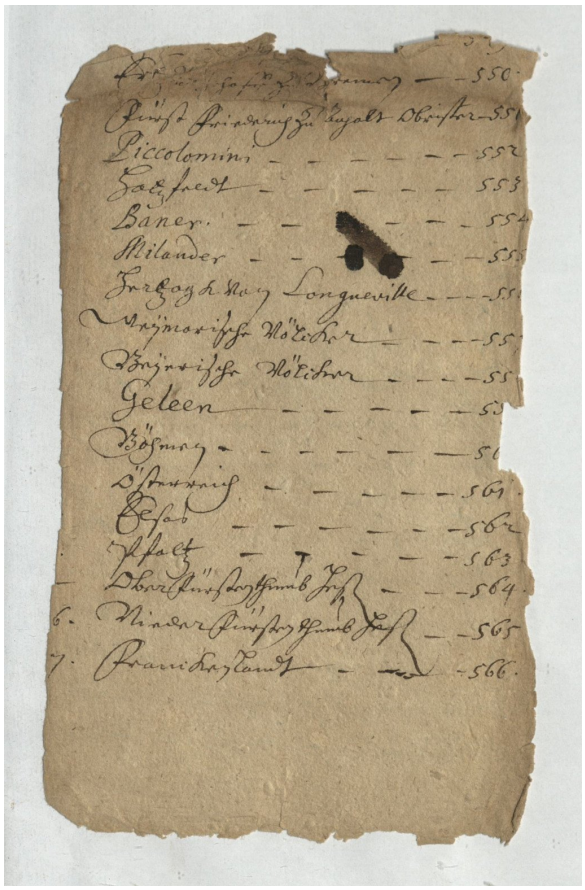
The examples in Fig. 1 show the visual range that motivates the generator: aged paper, physical damage, horizontal entries, vertical tables, and mixed handwriting conventions. These source images are cited from the source material used during generator development [5], with their archival sources retained [6, 7].

## 2. Related Work

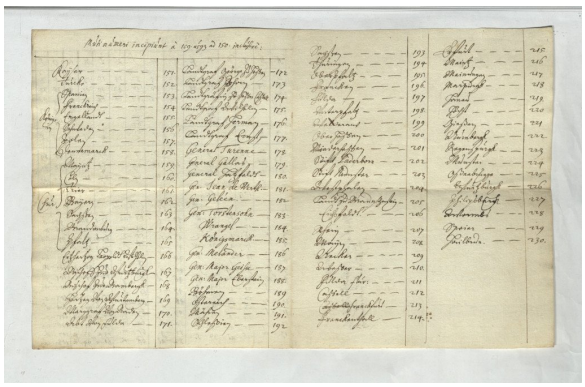
Digital historical cryptology depends on collections, tools, and domain-specific datasets. HCPortal provides an online platform for historical cryptology, including a database of cryptograms and tools for working with historical ciphers [8, 9]. Later work extended the portal with educational modules, a nomenclator-focused module, and a virtual museum concept [10]. These resources are important



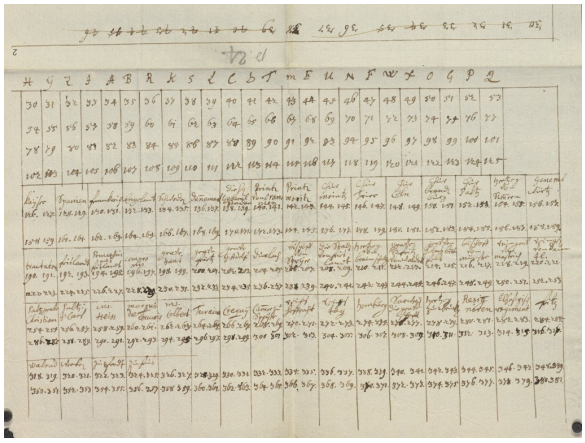
(a) Nomenclator key



(b) Paper grain and damage



(c) Horizontal layout



(d) Vertical layout

**Figure 1:** Historical cipher-key examples and layout patterns used as visual reference material during generator development. Images (a)–(c) are cited from HLA-HStAM 4d, nr. 1218; image (d) from HLA-HHStAW 130 i, nr. 1576 [6, 7, 5].

because historical encrypted documents and cipher keys are scattered across archives and often require specialized processing before computational methods can be applied.

Several studies describe nomenclator keys and related historical material. Megyesi et al. analyzed keys with nomenclatures in early modern Europe and provide a broad context for the structure and use of these sources [1]. Antal and Mírka studied cipher keys from the Hessisches Staatsarchiv Marburg used during the Thirty Years' War, with attention to key design and structural properties that could reduce cryptographic security [2]. Antal, Mírka, and Kováč investigated obfuscation techniques in cipher key instructions from the Vienna Cipher Office, including nulls, annullants, and false separators

[3]. These works motivate the generator design because they describe the components and layout conventions that synthetic nomenclator keys should approximate.

Machine learning has already been applied to historical encrypted manuscripts. Antal and Marák proposed an automated transcription method based on detection, classification, and transcription of handwritten digit symbols using Mask R-CNN [11]. Antal et al. presented encrypted documents and cipher keys from aristocratic family archives in Slovakia and used YOLOv7 for digit detection and segmentation in encrypted text [12]. The HHCS dataset provides annotated cipher symbols from handwritten historical encrypted documents for detection and segmentation tasks [13]. Šebeňa and Kačur addressed line detection in nomenclator-encrypted historical manuscripts using transfer learning with YOLOv8 [14]. These works show that modern detectors can process encrypted manuscripts, but they also show the need for specialized annotated data.

Synthetic data has been widely studied as a response to limited annotated datasets in computer vision and document analysis. Standard augmentation can improve generalization in low-resource settings, but its effect depends on the match between source and target distributions [15]. Historical document analysis adds further difficulty because the target domain includes paper ageing, ink degradation, scanning artefacts, writer variability, and non-standard layout. Raman et al. generate structured documents with automatic layout labels [16]. DocEmul produces structured historical documents with configurable degradations [17]. Other approaches use generative models for handwritten historical document synthesis [18]. General-purpose augmentation libraries such as Albumentations provide transformations for increasing visual variability [19]. These methods reduce annotation cost, but they do not directly model plaintext–ciphertext correspondences, homophonic alternatives, nulls, dictionary codes, or the nested relation between elements, pairs, and sections.

## 3. Synthetic Data Generator

The generator is implemented in Python and is available online at https://github.com/Ondrej850/thesis. It creates complete synthetic cipher-key images together with automatic bounding-box annotations. The design is rule-based. Document content, layout, handwriting parameters, paper appearance, and degradation effects are sampled from configurable ranges. This design was chosen because annotated real data is scarce, while the target documents contain structural rules that can be encoded directly.

The application follows a layered architecture. The model layer stores document, font, paper, and dataset configuration. The generator layer renders cipher pairs, table codes, paper backgrounds, handwriting variation, and post-processing effects. The annotation layer exports COCO and YOLO labels. The presentation layer provides a graphical interface for interactive configuration and batch dataset generation. Fig. 2 shows the generator interface. Fig. 3 shows representative generated documents.

The generator supports several cipher-key structures. Column-pair layouts can include alphabetic substitutions, general substitution entries, bigrams, trigrams, and dictionary entries. Table layouts can include alphabetic entries, bigrams, trigrams, words, and nulls. Code values can be numerical, double-character, or special-character based. Pair order can vary between text-first and number-first formats. Layout parameters control the number of records, number of columns, row spacing, column gaps, separators, and jitter.

The visual layer simulates selected properties of historical manuscript images. Paper ageing is controlled by a sampled ageing level. The generator can add stains, holes, cracks, ink drops, book or scan edges, and bleed-through. Bleed-through is simulated by generating or reusing reverse-side content, mirroring it, applying blur, and blending it with the front side at partial opacity. Post-processing uses Albumentations [19] for sepia tone, brightness and contrast variation, Gaussian or ISO noise, random shadows, mild perspective deformation, vignette, and small blur effects. Bounding boxes are transformed with the image so that annotations remain aligned after spatial augmentation.

Handwriting variation is simulated through font selection, font size, local shifts, and style parameters. The experimental dataset uses the Playwrite Desas Variable font, with low, medium, and high handwriting variation levels. Font size is sampled between 15 and 24 pixels for the main content. The

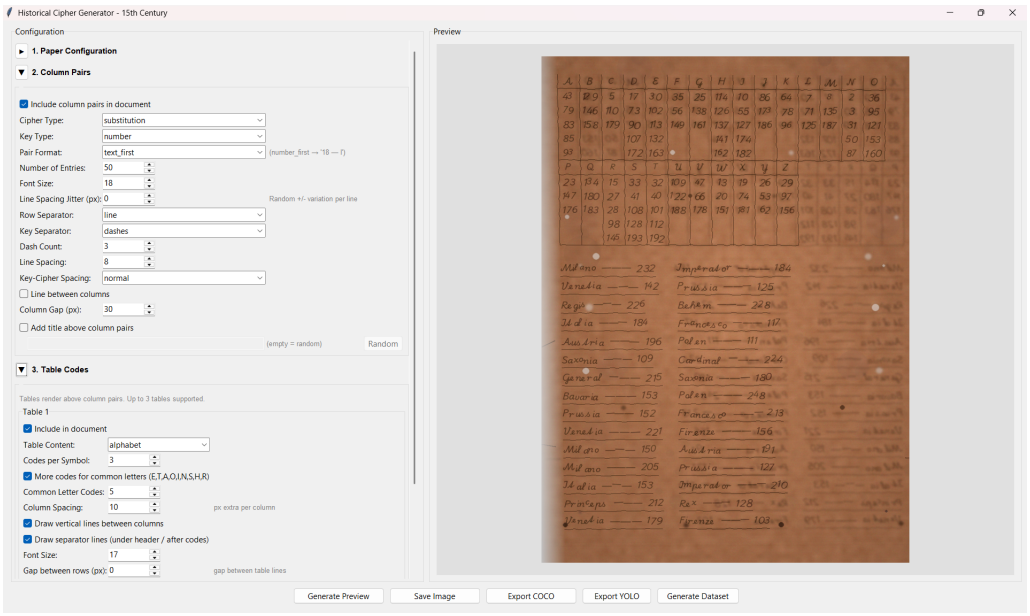
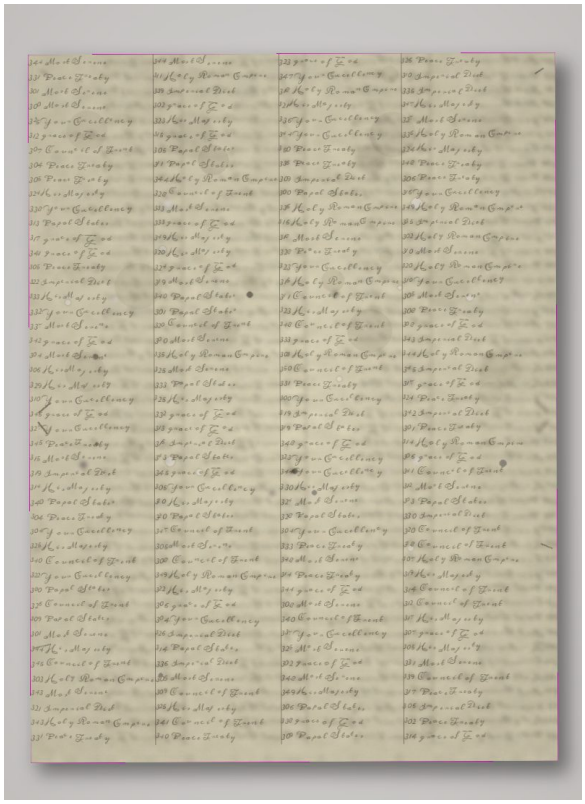
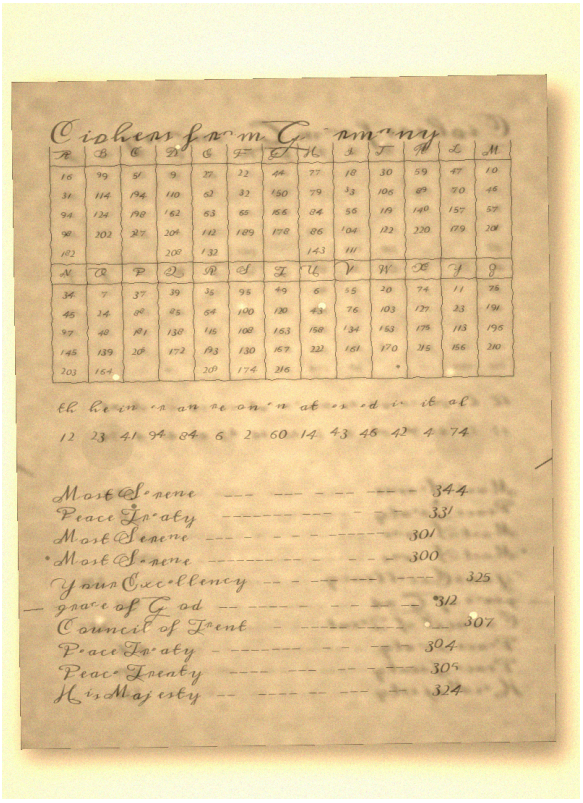



**Figure 2:** Generator interface used for interactive configuration and batch dataset generation [5].



(a) Generated document

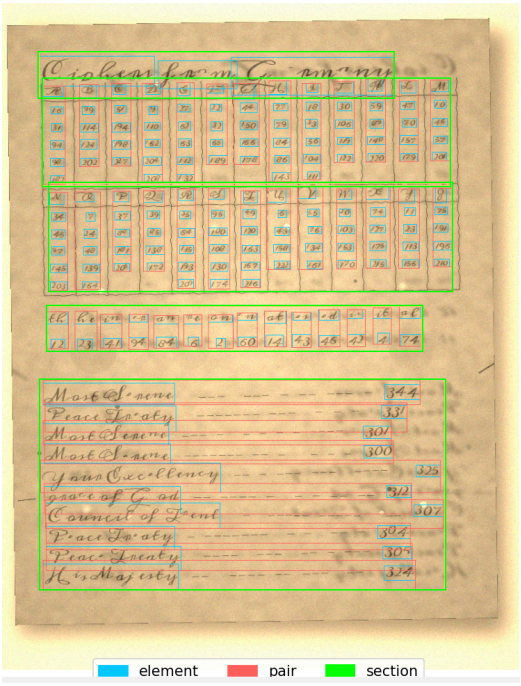


(b) Generated document

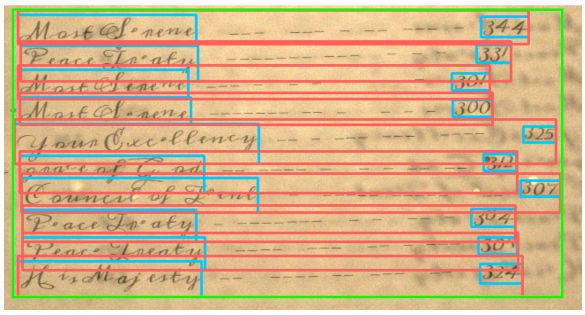
**Figure 3:** Representative synthetic cipher-key images generated by the rule-based pipeline [5].

generator therefore does not synthesize handwriting from a learned writer model. It creates controlled variation around font-rendered glyphs. This limitation is relevant when interpreting the domain gap between synthetic and real pages.

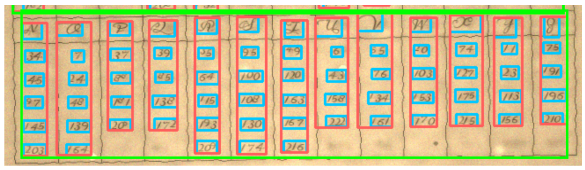
The generator exports annotations in COCO [20] and YOLO formats. COCO export stores images, annotations, and categories in a single JSON file. YOLO export creates one text file per image, with normalized `class x_center y_center width height` records. The categories correspond to the




(a) Three levels



(b) Columns



(c) Table

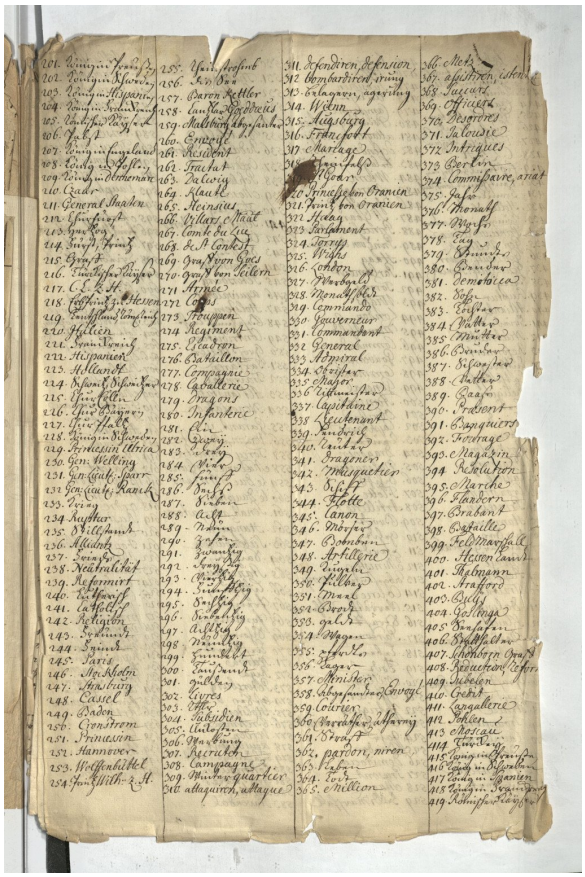
**Figure 4:** Annotation visualizations from the generated dataset. The figures illustrate the three-level hierarchy used for training and evaluation [5].

three hierarchy levels used by the detector: `element`, `pair`, and `section`. Fig. 4 shows examples of the visualized annotations.

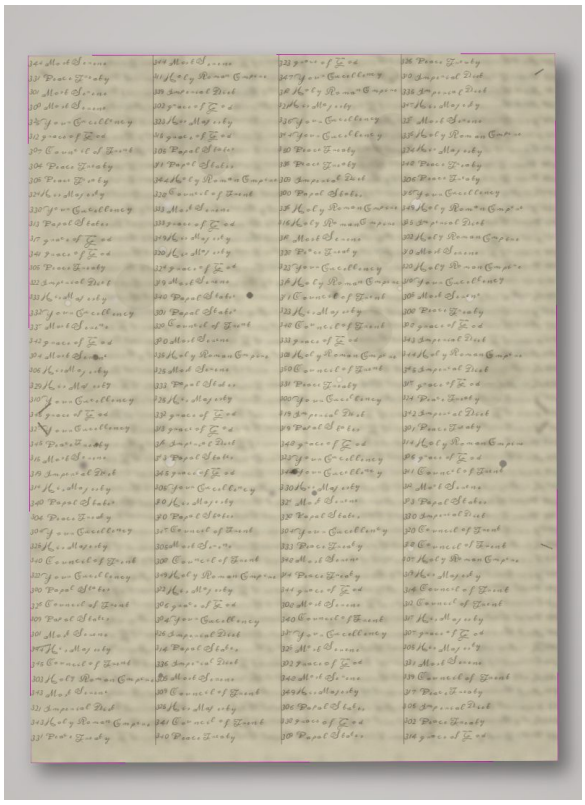
## 4. Experimental Setup

The experiment evaluates whether synthetic data improves detection performance on real historical cipher keys. All models use YOLOv11m [21] in a multi-class configuration with three categories: element, pair, and section. The goal is not to find the globally best YOLOv11 configuration. The goal is to isolate the effect of synthetic data under a fixed detector and evaluation protocol.

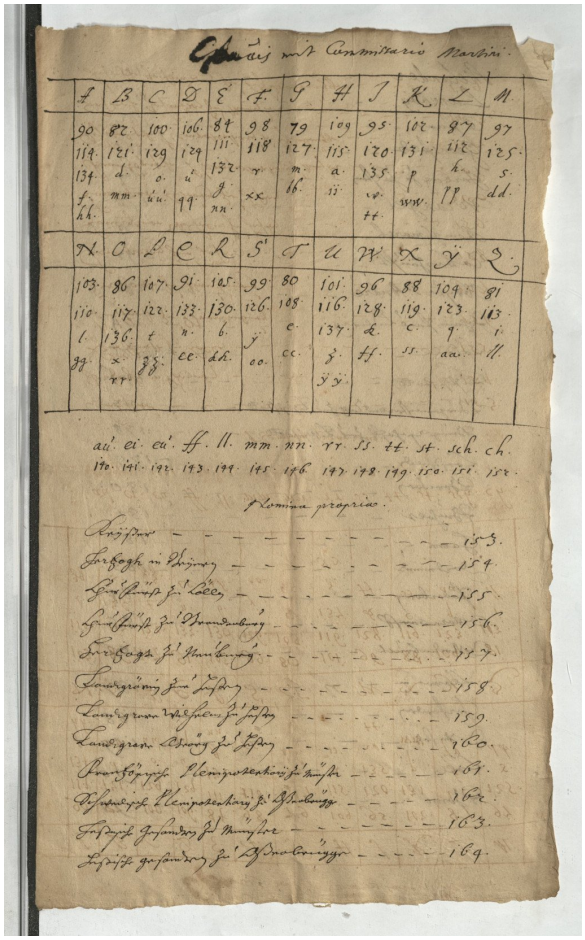
The real dataset follows the split used in the previous hierarchical detection study [4]. It contains 76 manually annotated pages. The training-side real data contains 68 pages split into 53 training and 15 validation pages. The test set contains 8 real pages and is kept fixed for all experiments. The real split includes 12,793 elements, 4,636 pairs, and 262 sections in training; 3,744 elements, 1,025 pairs, and 66 sections in validation; and 2,251 elements, 826 pairs, and 39 sections in testing. Fig. 5 shows representative real and synthetic pages cited according to the thesis sources [6, 22, 5].



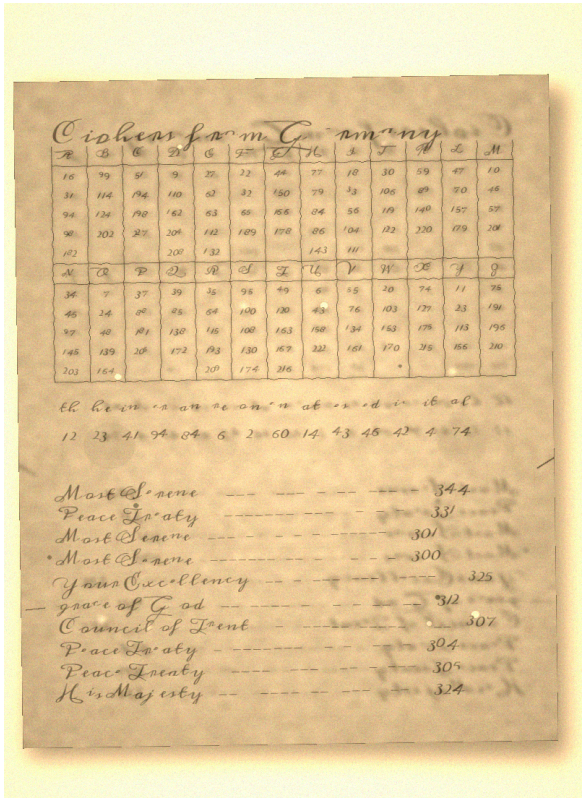
(a) Real document from HLA-HStAM 4f, nr. 271



(b) Synthetic document



(c) Real document from HLA-HStAM 4d, nr. 1218



(d) Synthetic document

**Figure 5:** Real and synthetic documents used to inspect the visual target domain. The real-image sources follow the cited archival references [22, 6, 5].

Two synthetic datasets are generated. The smaller synthetic dataset contains 200 generated images before splitting. It is divided into 156 training and 44 validation images. The larger synthetic dataset contains 500 generated images before splitting. It is divided into 390 training and 110 validation images. Both synthetic datasets are publicly available; access information is provided in the Data and Code Availability section. Combined datasets are formed by adding synthetic training and validation images to the corresponding real splits. The test set always contains the same eight real pages and never includes synthetic data.

Training is performed with YOLOv11m initialized from standard YOLOv11m weights. All runs use 400 epochs, patience 20, batch size 4, `imgsz`=640, AdamW optimization, initial learning rate $1 \times 10^{-4}$, cosine learning-rate scheduling, weight decay 0.0005, and `max_det`=1300. Here, `imgsz`=640 denotes the target model input size rather than the original dimensions of the manuscript images. The original real and synthetic pages are rectangular. During preprocessing, they are resized while preserving their aspect ratio and padded to the model input size. The same input size is used for all experiments so that dataset composition remains the primary experimental variable. Higher input resolutions may improve detection of small elements, but resolution optimization is outside the scope of this experiment. Training augmentation is conservative: rotation up to 2 degrees, translation 0.05, scale 0.1, no shear, no perspective, no vertical or horizontal flips, and no mosaic. These settings preserve the tabular and columnar structure of the pages. Evaluation uses confidence threshold 0.001, IoU threshold 0.70, image size 640, and `max_det`=1000.

## 5. Results

Table 2 reports detection results for all five training datasets on the same real test set. Synthetic-only training does not outperform real-only training. The smaller synthetic dataset reaches AP50:95 = 0.3975 and AP50 = 0.7175. The real-only dataset reaches AP50:95 = 0.5274 and AP50 = 0.8665. The gap indicates that simulated ageing, degradation, acquisition artefacts, and handwriting variation do not reproduce the full target distribution of real manuscripts.

Increasing the size of the synthetic dataset improves performance. AP50:95 increases from 0.3975 to 0.4595, and AP50 increases from 0.7175 to 0.8289. Element AP50:95 increases from 0.2675 to 0.3581. Pair AP50:95 increases from 0.4082 to 0.4921. Section AP50:95 changes only slightly, from 0.5169 to 0.5283. Additional generated variation therefore improves transferable element and pair cues, but has weaker effect on section localization.

At the AP50 threshold, Synth. L performs slightly better than the real-only model for elements (0.8314 versus 0.8218) and pairs (0.8613 versus 0.8573). However, the real-only model performs better for both classes when AP50:95 is used. This suggests that the synthetic-only model can often detect the correct object, but its bounding boxes are less precise. The small advantage at AP50 therefore disappears when stricter overlap thresholds are considered.

The best results are obtained by combining real and synthetic data. The smaller combined dataset reaches AP50:95 = 0.5628 and AP50 = 0.8902. The larger combined dataset reaches AP50:95 = 0.5884 and AP50 = 0.9166. Compared with real-only training, the larger combined dataset improves AP50:95 by 0.0610 and AP50 by 0.0501. The strongest class-level improvement is observed for pair detection. Pair AP50:95 increases from 0.5278 with real-only training to 0.6480 with the larger combined dataset. Pair AP50 increases from 0.8573 to 0.9569.

Section-level results are less consistent. Section AP50:95 improves from 0.6102 to 0.6281 with the larger combined dataset, but section AP50 decreases from 0.9203 to 0.8753. Section boxes are larger, less frequent, and more subjective than element or pair boxes. Synthetic section annotations follow deterministic layout rules, while real annotations may reflect ambiguous visual grouping decisions. The mismatch affects AP50 and stricter localization thresholds differently.

**Table 1**
Training and validation datasets used in the experiment.

| Dataset | Training images | Validation images | Description |
|---|---|---|---|
| Synth. | 156 | 44 | Smaller synthetic dataset |
| Synth. L | 390 | 110 | Larger synthetic dataset |
| Real | 53 | 15 | Manually annotated real pages |
| Comb. | 209 | 59 | Real data plus smaller synthetic dataset |
| Comb. L | 443 | 125 | Real data plus larger synthetic dataset |

**Table 2**
Detection results on the fixed real test set. E, P, and S denote element, pair, and section.

| Dataset | AP50:95 | AP50 | E 50:95 | E 50 | P 50:95 | P 50 | S 50:95 | S 50 |
|---|---|---|---|---|---|---|---|---|
| Synth. | 0.3975 | 0.7175 | 0.2675 | 0.6594 | 0.4082 | 0.7255 | 0.5169 | 0.7676 |
| Synth. L | 0.4595 | 0.8289 | 0.3581 | 0.8314 | 0.4921 | 0.8613 | 0.5283 | 0.7939 |
| Real | 0.5274 | 0.8665 | 0.4442 | 0.8218 | 0.5278 | 0.8573 | 0.6102 | 0.9203 |
| Comb. | 0.5628 | 0.8902 | 0.4805 | 0.9014 | 0.6011 | 0.9201 | 0.6067 | 0.8490 |
| Comb. L | 0.5884 | 0.9166 | 0.4892 | 0.9176 | 0.6480 | 0.9569 | 0.6281 | 0.8753 |

## 6. Discussion and Limitations

The results indicate that rule-based synthetic data is useful as augmentation, but not as a replacement for manually annotated real pages. Synthetic-only models remain below the real-only model on AP50:95. This indicates a remaining domain gap between generated and historical documents. Real cipher keys contain writer-specific conventions, irregular symbols, inconsistent spacing, marginal notes, corrections, page damage, and acquisition artefacts that are only partly represented by the generator.

The strongest results are obtained when synthetic and real data are combined. Compared with real-only training, the larger combined dataset increases AP50:95 from 0.5274 to 0.5884. The largest gain is observed for pair detection, where AP50:95 increases from 0.5278 to 0.6480. This is particularly relevant because pairs represent the relation between plaintext and ciphertext values and can support later reconstruction of substitution tables, semi-automatic annotation, and comparison between keys and encrypted messages. Increasing the synthetic dataset from Synth. to Synth. L also improves performance, although the experiment does not show whether this trend would continue with substantially larger synthetic datasets.

The study has several limitations. The real test set contains only eight pages, which limits the statistical confidence of the results. The generator mainly produces controlled tabular and column-pair layouts and does not cover the full diversity of historical nomenclator keys. Visual realism was assessed manually rather than through a formal perceptual study or quantitative comparison of synthetic and real image distributions. Section annotations are also less consistent because section boundaries can be ambiguous in historical documents. Future work should evaluate the method on a larger and more diverse real dataset and improve the representation of irregular layouts, handwriting, and section boundaries in the synthetic data.

## 7. Conclusion

This paper evaluated rule-based synthetic data generation for hierarchical object detection in historical handwritten substitution cipher keys. The generator creates synthetic nomenclator-key images with simulated historical paper, degradation, acquisition artefacts, handwriting variation, and automatic annotations in YOLO and COCO formats. The annotations cover three hierarchy levels: elements, plaintext–ciphertext pairs, and sections.

The evaluation compared YOLOv11m models trained on synthetic-only, real-only, and combined

datasets. Synthetic-only training did not outperform real-only training. Increasing the synthetic dataset improved performance, and the best result was obtained by combining real and synthetic data. The larger combined dataset reached AP50:95 = 0.5884 and AP50 = 0.9166 on the real test set, improving over real-only training by 0.0610 AP50:95 and 0.0501 AP50. The largest gain was observed for pair detection, where AP50:95 increased from 0.5278 to 0.6480. These results show that synthetic annotated cipher-key images can improve detection in low-resource historical document settings when used as augmentation. The findings remain preliminary because the real test set is small and the generator does not yet cover the full diversity of historical nomenclator layouts.

## Data and Code Availability

The synthetic datasets used in the experiment are publicly available. The smaller 200-image dataset is available at [23], and the larger 500-image dataset at [24]. The generator implementation is available at [25]. Access to the real historical pages depends on the archival and portal sources from which the images were obtained.

## Acknowledgments

This research was supported by the Ministry of Education, Science, Research, Development and Youth of the Slovak Republic through grant VEGA 2/0054/24.

## Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT for language editing and proofreading. The authors reviewed and edited the resulting text and take full responsibility for the content of the publication.

## References

[1] B. Megyesi, C. Tudor, B. Lang, A. Lehofer, N. Kopal, K. de Leeuw, M. Waldispuhl, Keys with Nomenclatures in Early Modern Europe, Cryptologia 48 (2024) 97–139. doi:10.1080/01611194.2022.2113185.

[2] E. Antal, J. Mírka, Wrong Design of Cipher Keys: Analysis of Historical Cipher Keys From the Hessisches Staatsarchiv Marburg Used in the Thirty Years' War, in: Proceedings of the 5th International Conference on Historical Cryptology, Linköping University Electronic Press, 2022, pp. 1–11. doi:10.3384/ecp188387.

[3] E. Antal, J. Mírka, D. Kováč, Development of Obfuscation Techniques in Vienna During the Early Modern Era, Cryptologia 50 (2026) 191–221. doi:10.1080/01611194.2025.2457096, published online 1 May 2025.

[4] S. Marochok, M. Balogh, Hierarchical Object Detection in Historical Handwritten Substitution Cipher Keys Using YOLOv11, in: DIVAI 2026, Springer, 2026. In press.

[5] O. Páleník, Generátor Syntetických Rukou Písaných Šifrovacích Kľúčov, Bachelor's thesis, Slovak University of Technology in Bratislava, Faculty of Electrical Engineering and Information Technology, 2026. Supervisor: Stanislav Marochok.

[6] Hessisches Landesarchiv – Hessisches Staatsarchiv Marburg, Politische Akten nach Philipp dem Großmütigen: Kanzlei- und Geheimeratskorrespondenz (4d), Nr. 1218, n.d. Archival source.

[7] Hessisches Landesarchiv – Hessisches Hauptstaatsarchiv Wiesbaden, Walramisches Hausarchiv (130 i), Nr. 1576, n.d. Archival source.

[8] E. Antal, Portal of Historical Ciphers, https://hcportal.eu/, 2019. Accessed 27 February 2025.

[9] E. Antal, P. Zajac, HCPortal Overview, in: HistoCrypt 2020: 3rd International Conference on Historical Cryptology, Linköping University Electronic Press, 2020, pp. 18–20.
[10] E. Antal, P. Zajac, HCPortal Modules for Teaching and Promoting Cryptology, in: HistoCrypt 2021: 4th International Conference on Historical Cryptology, Linköping University Electronic Press, 2021, pp. 1–11.
[11] E. Antal, P. Marák, Automated Transcription of Historical Encrypted Manuscripts, Tatra Mountains Mathematical Publications 82 (2022) 65–86. doi:10.2478/tmmp-2022-0019.
[12] E. Antal, P. Marák, P. Zajac, T. Lengyelová, D. Duchoňová, Encrypted Documents and Cipher Keys From the 18th and 19th Century in the Archives of Aristocratic Families in Slovakia, in: HistoCrypt 2023: 6th International Conference on Historical Cryptology, Linköping University Electronic Press, 2023, pp. 1–12. doi:10.3384/ecp195689.
[13] E. Antal, P. Marák, F. Mikuš, HHCS: A Dataset of Cipher Symbol Annotations From Handwritten Historical Encrypted Documents for Machine Learning Tasks, IEEE Access 14 (2026) 9226–9240. doi:10.1109/ACCESS.2026.3654267.
[14] M. Šebeňa, J. Kačur, Line Detection in Nomenclator-Encrypted Historical Manuscripts, in: Proceedings ELMAR-2025: 67th International Symposium, Croatian Society Electronics in Marine, Zadar, Croatia, 2025, pp. 307–310. doi:10.1109/ELMAR66948.2025.11194031.
[15] C. Shorten, T. M. Khoshgoftaar, A Survey on Image Data Augmentation for Deep Learning, Journal of Big Data 6 (2019). doi:10.1186/s40537-019-0197-0.
[16] N. Raman, S. Shah, M. Veloso, Synthetic Document Generator for Annotation-Free Layout Recognition, Pattern Recognition 128 (2022) 108660. doi:10.1016/j.patcog.2022.108660.
[17] S. Capobianco, S. Marinai, DocEmul: A Toolkit to Generate Structured Historical Documents, 2017. arXiv:1710.03474.
[18] L. Vogtlin, M. Drazyk, V. Pondenkandath, M. Alberti, R. Ingold, Generating Synthetic Handwritten Historical Documents With OCR Constrained GANs, 2021. arXiv:2103.08236.
[19] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, A. A. Kalinin, Albumentations: Fast and Flexible Image Augmentations, Information 11 (2020) 125. doi:10.3390/info11020125.
[20] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, C. L. Zitnick, Microsoft COCO: Common Objects in Context, in: European Conference on Computer Vision, Springer, 2014, pp. 740–755.
[21] R. Khanam, M. Hussain, YOLOv11: An Overview of the Key Architectural Enhancements, 2024. arXiv:2410.17725.
[22] Hessisches Landesarchiv – Hessisches Staatsarchiv Marburg, Staatenabteilung: Sachsen – Sulz (4f), Nr. 271, n.d. Archival source.
[23] S. Marochok, O. Páleník, Synthetic Nomenclator-Key Dataset: 200 Images, Google Drive, 2026. URL: https://drive.google.com/drive/folders/1-_xH_SHk64gC8_Nrync_P58qdrqzQLIm?usp=sharing, dataset used for training and validation in the synthetic-data augmentation experiments.
[24] S. Marochok, O. Páleník, Synthetic Nomenclator-Key Dataset: 500 Images, Google Drive, 2026. URL: https://drive.google.com/drive/folders/1wtuxlGgISHCI0aoAib5XAEpgisBXe_E7?usp=sharing, dataset used for training and validation in the synthetic-data augmentation experiments.
[25] O. Páleník, Synthetic Nomenclator-Key Generator, GitHub repository, 2026. URL: https://github.com/Ondrej850/thesis, source code for the synthetic nomenclator-key generator.