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.

Handwritten German Administrative Forms (HGAF): A Dataset for Handwriting Recognition with Fine-Tuned and Zero-Shot Baselines

Tobias Kalmbach

tobias.kalmbach@proton.me

Karlsruhe University of Applied Sciences

Andreas Wagner

Karlsruhe University of Applied Sciences

Research Article

Keywords: Handwriting dataset, Handwritten text recognition, Form understanding, Vision-languagemodels, TrOCR

Posted Date: November 5th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11348739/v1

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

Additional Declarations: No competing interests reported.

Handwritten German Administrative Forms (HGAF): A Dataset for Handwriting Recognition with Fine-Tuned and Zero-Shot Baselines

Tobias Kalmbach[1*] and Andreas Wagner[1,2]

[1]Karlsruhe University of Applied Sciences, Faculty of Management Science and Engineering, Karlsruhe, Germany.
[2]Fraunhofer Institute for Industrial Mathematics ITWM, Kaiserslautern, Germany.

*Corresponding author(s). E-mail(s): tobias.kalmbach@proton.me;
Contributing authors: andreas.wagner@h-ka.de;

Abstract

Administrative processes remain largely paper-based, and handwritten form contents must be transferred into structured data. Where forms carry personal information, the proprietary cloud services that lead published benchmarks are not suitable, so only on-premise models remain. Currently, no public corpus supports this setting: handwriting datasets hold continuous prose, and form datasets hold printed text. Therefore, a dataset of 291 German administrative forms, named Handwritten German Administrative Forms (HGAF), was collected from 95 writers. Each writer completed three layouts, A, B and C, with identical fields and identical synthetic contents. The visual structure therefore varies, while writer and content stay constant. The corpus carries no personal information and can be released. Two approaches were implemented and scored per field: a specialist pipeline around a fine-tuned line-level TrOCR model, and three open-weight vision-language models (VLM) applied without fine-tuning. On layout A, the only layout used for fine-tuning, two zero-shot models reach 4.5% and 4.7% character error rate (CER), against 6.8% for the specialist pipeline. On layouts B and C, the gap widens. No on-premise configuration transcribed more than 7.7% of forms without an error. In this scenario, VLMs reached the best performance. Although they do not require fine-tuning, they may still require configuration adjustments to fulfil the HTR task properly. We recommend prompting a general instruction-following model with a fixed output schema. This removes the extraction stage and allows the system to be adapted to different form layouts with little effort.

Keywords: Handwriting dataset, Handwritten text recognition, Form understanding, Vision-language models, TrOCR

1 Introduction

Even in today's digital age, physical forms are still common practice. Bitkom e.V. [7] reports that 38% of German companies still handle about half of their office and administrative processes on paper, and a further 20% handle three quarters or more. As companies move towards paperless work, large quantities of physical documents carrying handwritten text can be expected to require digitisation. This is a promising field of application

for optical character recognition (OCR) and handwritten text recognition (HTR). The same holds for historical administrative records, such as census returns or civil registers [3]. Their digitisation faces the same problem of extracting handwritten text from a form.

Form recognition differs from sentence- or paragraph-level recognition in two ways. First, form contents are short and frequently arbitrary, so they are poorly predicted from linguistic context. Second, each value has to be associated with the field it belongs to rather than merely transcribed. Based on our literature review, the recognition and extraction of handwritten text from modern, non-English forms is an under-studied field.

One simple solution would be to use commercial multimodal large language models (MLLMs), such as OpenAI's ChatGPT or Google's Gemini. These models perform very well on handwriting benchmarks [12], with character error rates below 2%, and are increasingly applied to handwritten forms [33]. However, commercial models require the data to be uploaded to third-party servers. This may be undesirable for confidentiality reasons, or prohibited under data protection regulations.

This work therefore studies which options an organisation has to recognise and extract handwritten form contents reliably on-premise, without the documents leaving its own infrastructure. The focus is on German administrative forms of identical layout and structure, filled out with different contents and handwriting.

Two families of approaches are compared: a classic HTR pipeline built around a fine-tuned line-level[1] model, and vision-language models (VLMs, a subclass of MLLMs) applied zero-shot. They differ in two aspects. First, the position of every field on a form is fixed by the layout. The HTR pipeline exploits this and passes only the handwritten fields to a small recogniser. A VLM processes the whole page and reads the printed labels and headers again on every form, even though they are identical across documents. Second, specialist HTR models typically require fine-tuning on annotated in-domain data, whereas VLMs are intended to work without task-specific training. The comparison therefore addresses a practical question: is the effort of fine-tuning a specialist pipeline still justified, or does a pre-trained model used without adaptation reach the same accuracy?

In practice, an organisation processes several types of forms, and existing forms are revised over time. HTR models are known to lose accuracy on data that differs from their training distribution [18]. For this reason, both approaches are also evaluated on two further layouts, without retraining the line-level model. In this work, a form layout is considered one type of form, with a given set of fields and a given visual structure.

A dataset of 291 forms filled in with real, modern, mostly German handwriting was created and named Handwritten German Administrative Forms (*HGAF*) [24]. It consists of three different layouts (see Fig. 1). The current state of the art was then surveyed, and the relevant approaches, models and processing steps were studied. On this basis, a processing pipeline was implemented for each model.

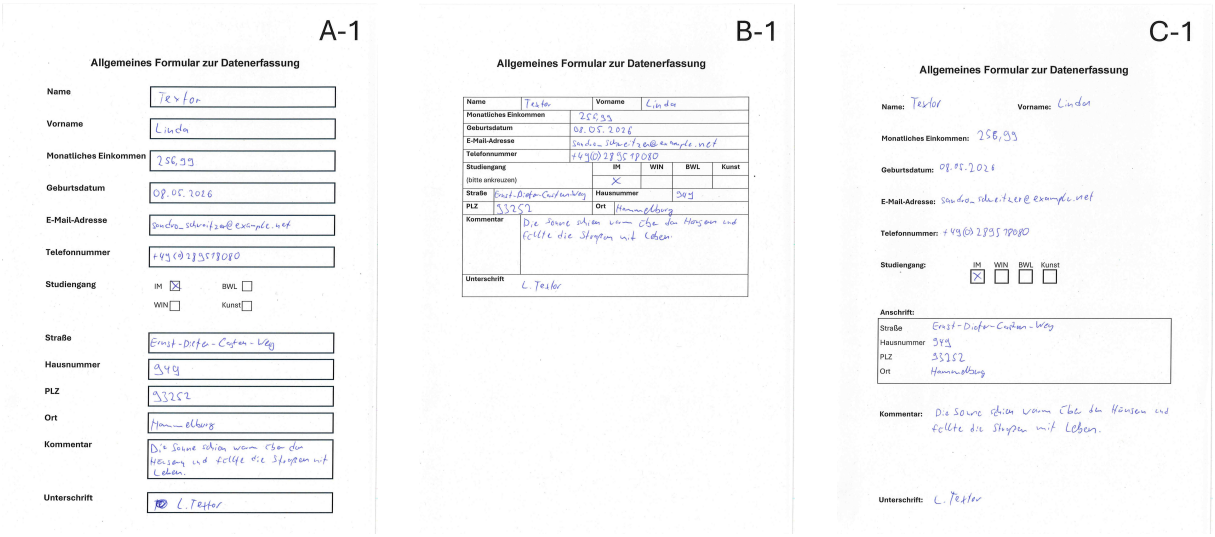
For handwritten German administrative forms processed on-premise, this work addresses two research questions (RQs):

- **RQ1:** On a known form layout, how does the recognition accuracy of a fine-tuned line-level HTR pipeline compare to that of zero-shot VLMs?
- **RQ2:** Does this result hold on layouts that contain the same fields in a different visual structure?

The contributions of this work are the following:

- *HGAF*, a dataset of 291 forms with modern, mostly German handwriting across three distinct layouts that share an identical field set. Each writer filled in all three layouts with the same content, so writer and content are held constant across layouts. Because the content is synthetic, the forms contain no real personal data. This allows the dataset to be published, a property most handwriting corpora of administrative documents lack. The dataset supports: (1) HTR research on short, discrete-field inputs, in contrast to the line- and paragraph-level focus of most existing corpora; (2) isolating the

[1] Line-level models recognise one line of text at a time. They therefore expect input images that contain only one line of text.



(a) **(b)** **(c)**

Fig. 1 The three form layouts, all for document ID 1: **a** layout A, **b** layout B, **c** layout C. Full-page reproductions, together with the instruction for completion, are given in Supplementary Information (Figs. S1–S4).

effect of the layout on the handwriting and its recognition; and (3) structured field extraction research on German handwritten forms.

- Evidence that a fine-tuned line-level pipeline and a zero-shot vision-language model can reach comparable transcription accuracy on short texts after post-processing, but diverge sharply on long texts and once the layout changes.
- Evidence that form layout can have an impact on HTR performance for all evaluated models, even whole-page models.

The remainder of this work is structured as follows. Sect. 2 presents the research this work builds upon, Sect. 3 the datasets and models, Sect. 4 the implementation of the pipelines, Sect. 5 their evaluation, and Sect. 6 the interpretation of the results. Sect. 7 answers the research questions and gives an outlook on future work.

2 Related Work

2.1 Datasets and Evaluation Metrics

Regarding modern handwriting, defined in this work as handwriting whose spellings and scripts follow modern writing conventions, the most widely used benchmark is *IAM* [31]. It consists of English sentences copied from a text corpus and provides line- and word-level ground truth for continuous prose. A second group of frequently used datasets is historical, for example the *Washington* and *Parzival* datasets [15] and the *U-DIADS-TL* dataset [41]. These datasets target manuscripts several centuries old, where the recognition challenge lies in archaic scripts, degraded material and unfamiliar orthography.

Two properties of this landscape are relevant to the present work. First, the modern benchmarks consist predominantly of continuous prose. Administrative forms, in contrast, consist of short, isolated field entries whose content is frequently arbitrary (postal codes, house numbers, email addresses) and therefore poorly predicted from linguistic context. Second, the field is strongly English-centred. For German, the only modern line- and word-level dataset of comparable size known to us is the *ScaDS.AI German Line- and Word-Level Handwriting Dataset* [8]. It provides 5,843 line images and 37,406 word images, segmented from 775 pages on which the participants transcribed excerpts of Wikipedia articles. Since it consists of continuous prose, it does not contain the short field entries of an administrative form.

Datasets of forms exist, but most of them contain no handwriting. *FUNSD* [23] and its multilingual extension *XFUND* [38], which includes German, provide scanned forms of primarily printed text. They are annotated with key-value pairs for semantic entity recognition and relation extraction. Among datasets of genuinely handwritten

forms, *NIST Special Database 19* [16] remains the best known. Its handwriting sample form comprises digits, isolated characters and a single shared paragraph of running text. This limits its usefulness for evaluating field-level transcription of varied content. Another example is *NAF* [13], where a variety of forms and other documents containing both printed and handwritten text from the 1800s and early 1900s are presented.

The dominant evaluation metric is the character error rate (CER), defined as the normalised Levenshtein distance [27] between prediction and reference at character level. Some works additionally report accuracy as the proportion of images transcribed without any error, as in Liu et al. [30].

2.2 Specialist Handwritten Text Recognition: Line-Level Models

Traditional HTR models usually operate at line level and require input images that contain only one line of text. These models can therefore rarely be applied on their own. They generally require further steps alongside the recognition itself, referred to in this work as a pipeline. In general terms, such a pipeline consists of the following steps: (1) image acquisition, (2) image pre-processing, (3) text detection and segmentation, (4) text recognition, and (5) post-processing [1].

Common pre-processing techniques include binarisation, noise removal, skew correction and image normalisation [1], among other techniques that manipulate the images. Segmentation divides the original image or document into the crops that the models require as input, for instance into individual text lines for line-level models. Common, simpler techniques are projection-based approaches, such as those surveyed in Likforman-Sulem et al. [29]. The HTR models themselves, which are generally machine-learning-based, are then applied to the pre-processed images. These models usually have to be fine-tuned or adapted to the domain they will work on [18]. Post-processing may then be applied to improve the recognised text. Common post-processing techniques include error models, lexical approaches and context-aware language models that verify and correct the recognised text [32].

Both convolutional neural network (CNN) and transformer architectures achieve strong performance on standard benchmarks, although transformer models report better scores. Li et al. [28], for instance, report a CER of 2.89% for their large model on the *IAM* dataset. CNN architectures reach between 4% and 5% CER on the same dataset (as reported, for example, by Retsinas et al. [34] and Yousef et al. [39]).

A line-level model is normally not deployed as it comes. Fine-tuning it on the target domain is the expected step, and comparatively little in-domain data is needed for it. Aradillas Jaramillo et al. [4] attain a CER of 5.3% on *Washington* with only 325 line images from the target domain. Lauar and Laurent [26] adapt TrOCR from English to Spanish in the same manner, by fine-tuning it on synthetic images of printed text in the target language.

2.3 Generalist Handwritten Text Recognition: Vision-Language Models

VLMs for OCR and document intelligence have gained considerable traction as a line of research in recent years. Liu et al. [30] present a benchmark of such models across a range of document-related tasks. On these tasks, VLMs match or exceed the performance of the models previously used. The tasks were, however, measured predominantly on printed text and documents. For HTR, the reported performance is considerably weaker: the best model reaches 55.4% overall accuracy on the *IAM* dataset, against 91.24% for their non-VLM reference, AttentionHTR [25].[2] More recent work reports a best CER of 1.71% for GPT-4o-mini on *IAM*, while open-weight models such as Qwen2-VL-7B reach 2.30% [12]. The same study reports that overall performance degrades for datasets in other languages and for historical datasets, particularly for open-weight models. This indicates both a visual and a textual bias of these models towards modern English.

VLM approaches permit end-to-end processing of the documents. The five steps described in Sect. 2.2 are therefore not necessarily required as separate stages.

[2] This accuracy is measured as the proportion of images transcribed without any error.

VLMs pair a vision transformer (ViT) family encoder [14] with a general-purpose, instruction-tuned large language model as decoder. The decoder receives the visual embeddings interleaved with a natural-language prompt [5].

VLMs are not normally fine-tuned by the user. For languages and scripts that are well represented in the training data, they are expected to work zero-shot [12]. Adapting them is expensive: Chung and Choi [11], for instance, use 60,000 synthetic word images to adapt one to a single low-resource script. Adaptation is therefore usually carried out once by the model provider, as in the OCR-specialised models of Chen et al. [10], Wei et al. [37]. These models are built on VLMs but optimised for capturing fine-grained information in dense images such as document text.

Pather et al. [33] excluded open-weight models from part of their evaluation because of poor performance. The question of which on-premise, open-weight models can nevertheless be used therefore remains open.

2.4 Extraction of German Handwritten Text from Forms

Research on German handwriting recognition concentrates almost entirely on historical material, as in Hodel et al. [22], Ströbel et al. [35], Tikhonov et al. [36]. To our knowledge, the *ScaDS.AI German Line- and Word-Level Handwriting Dataset* [8] is the only resource for modern German handwriting of comparable scale, and it consists of continuous prose. Modern German handwriting in short, structured fields is therefore represented neither in the available literature nor in the available data.

Work on extracting handwritten content from forms already exists. Ambata et al. [2], for example, recognise handwriting in a highly structured form at character level. This suits their case, because each field is completed in capital letters, one letter per box. Grabowski [20] addresses handwritten exams with a deep neural network that performs detection and classification jointly. It reaches an overall recognition rate of 88.28% on real examination data. Two recent studies do apply VLMs to handwritten forms. Grabowski [21] evaluates large language models (LLMs) on 61 anonymised German examinations comprising 3,141 answer positions, reaching 98.4% accuracy and reducing the false-negative rate to 0.58% by supplying the reference solution as prompt context. Pather et al. [33] benchmark seventeen frontier and open-weight models on a handwritten medical form. They report accuracies of around 85% and weighted F1 scores near 90% across discrete fields, with the strongest results obtained by proprietary models.

No publicly available resource provides complete, filled-in modern administrative forms with field-level ground truth for the transcription task. None provides identical field content across several layout variants, which would be required to isolate the effect of the layout on the handwriting and its recognition. The present work differs from the works discussed above in three respects. The first is the recognition target. Ambata et al. [2], Grabowski [20] operate on constrained entries written into predefined cells, and Grabowski [21] on answers that can be checked against a known reference solution. The fields considered in this work, in contrast, contain complete values of heterogeneous type, among them names, dates, monetary amounts, email addresses, postal codes and free-text comments, for which no closed answer set exists. The second is the experimental factor. None of those studies evaluates the effect of layouts on HTR, whereas the present work holds field content constant across three layouts to isolate the effect of the layout on the handwriting and its recognition. The third is the deployment constraint. The best-performing systems reported by Pather et al. [33] are proprietary cloud services, which are unavailable wherever personal data may not leave the organisation. This work therefore restricts itself to models that can be operated on-premise. It additionally includes a fine-tuned line-level specialist as a comparison condition, rather than evaluating zero-shot models alone.

3 Datasets and Models

3.1 Datasets

Handwritten German Administrative Forms (HGAF)

The datasets presented in Sect. 2.1 show a clear gap. A dedicated dataset of German handwritten forms was therefore created. The goal was to acquire real handwritten text that contains no personal information and can therefore be shared.

Three layouts were created to test the effect that different types of forms have on HTR. Layout A is a simple form with big fields and enough space for any type of handwriting. It was designed as the easiest case. Layout B is a compact form in tabular format with small fields, which forces the user to write smaller. In layout C, the fields are not clearly delimited, which leaves the decision of where and how to fill them out to the user. The three layouts are shown in Fig. 1.

The fields were kept identical across the three layouts to allow a clear cross-layout comparison. They were chosen to cover a wide range of contents: ten single-line text and number fields covering the usual fields in administrative forms (last name, first name, phone, email address, etc.), one long free-text field (*comment*), one checkbox group (*studies*) and a signature, which is not scored. Table 1 gives the complete list, with the German label printed next to each field.

A set of 100 combinations of these fields was generated synthetically. The majority of the contents was generated with Faker[3], using its German data provider. Microsoft Copilot, prompted to produce both German and international contents, was also used.

Each participant received one instruction for completion and one copy of each layout, all four documents carrying the same document ID. A participant with document ID 42, for example, filled in three forms, one per layout, with the contents of ID 42. This fixes the combination of writer, ground truth and document ID for the whole dataset. The collection was anonymous: no participant identifiers were recorded, and no mapping between document ID and person exists. Three document IDs were discarded because of errors in printing or filling in. The *HGAF* dataset therefore consists of 291 forms, 97 per layout, written by 95 participants, mostly university students. Two participants each filled in two document IDs, one in cursive and one in block letters.

The forms were split by document ID into training (40%), validation (20%) and test data (40%), consistently across all three layouts. The three forms of a given document ID therefore always fall into the same split. Table 2 gives the resulting splits. The 40/20/40 split was preferred over the more typical 60/20/20 or 70/15/15 for two reasons. First, a small training split is closer to a real scenario, where annotating forms has a cost that is kept to a minimum. Second, a large test split allows better conclusions about the real performance of the models.

[3] https://faker.readthedocs.io/en/master/

Table 1 Overview of the forms' fields. The German label is the text printed next to the field on all three layouts. The stated type refers to the kind of content the field predominantly contains; it does not imply that a field consists exclusively of that character class. Every field is scored except the signature.

Field	Label on the form	Type
Last name	Name	Text
First name	Vorname	Text
Monthly income	Monatliches Einkommen	Number
Birthday	Geburtsdatum	Number
Email	E-Mail-Adresse	Text
Phone	Telefonnummer	Number
Studies	Studiengang	Checkbox
Street	Straße	Text
House number	Hausnummer	Number
Postal code	PLZ	Number
City	Ort	Text
Comment	Kommentar	Long text
Signature	Unterschrift	Signature

Table 2 Overview of the splits of the *HGAF* dataset. Fields and lines are counted per layout; the last column gives the number of layout A text lines available for fine-tuning TrOCR.

Split	%	Doc. IDs	Fields	Lines
Training	40	39	468	467
Validation	20	19	228	226
Test	40	39	468	467

ScaDS.AI German Line- and Word-Level Handwriting Dataset

The *ScaDS.AI German Line- and Word-Level Handwriting Dataset* [8] was used as pre-training data for TrOCR. The base TrOCR model is fine-tuned on *IAM*, an English dataset, whereas the target domain of this work is German forms. The model therefore has to be adapted to the target language, as done in related work such as Lauar and Laurent [26]. The line images were split by page in an 80/20 ratio, resulting in 4,691 images for training and 1,152 for validation. Since the

final tests are performed on the *HGAF* dataset, no test split was used for this dataset.

Its lines are continuous prose without printed boxes or lines. The dataset therefore supplies the target language, but not the visual characteristics of a cropped form field.

3.2 Models and Training

Specialist line-level model

TrOCR [28] was selected as the line-level model, in the variant `trocr-base-handwritten`. The decisive factors were its excellent reported performance on datasets such as *IAM* and the scale of its pre-training, which should allow a language adaptation as in Lauar and Laurent [26]. The adaptation to the target domain was conducted in two stages, similar to Aradillas Jaramillo et al. [4]. First, `trocr-base-handwritten` was pre-trained on the *ScaDS.AI* dataset as source domain. This stage provides the textual information, i.e. it adapts the model to the German language. Second, the model was fine-tuned on line images of the layout A fields of the *HGAF* dataset as target domain. This stage provides the visual information, that is, how a cropped form field looks. This approach should address both textual and visual divergence, as presented in Garrido-Munoz and Calvo-Zaragoza [18].

The hyperparameters were: batch size 8, learning rate 7.5e-6, warm-up ratio 0.1, weight decay 0.01, and early stopping with patience 3.

Generalist vision-language models

Three VLMs were selected. `Qwen3-VL-8B-Instruct` [5] represents the most generalist case: a general instruction-following VLM with no specialisation towards document transcription. `chandra-ocr-2`[4] and `DeepSeek-OCR-2` [37] are both OCR-specialised, but take opposite approaches to encoding a page. Chandra encodes the full page at high token cost, comparable to Qwen, whereas DeepSeek compresses it into a few hundred vision tokens, substantially reducing the compute required per page on local hardware. The three models thus cover both general instruction-following models (Qwen) and OCR-specialised models (Chandra and DeepSeek). Within the specialised models, two contrasting page-encoding strategies are covered. In the remainder of this work, they are referred to as Qwen, Chandra and DeepSeek.

The three models were evaluated zero-shot. This follows from the similarity in script between English and German and from the amount of data that fine-tuning such models requires [11]. Here, *zero-shot* denotes the absence of fine-tuning or gradient-based adaptation to the *HGAF* dataset, as well as the absence of training data examples in the prompt. It does not, however, denote the absence of configuration for a specific layout: prompts, output formats and extraction strategies were adjusted on the validation split of each layout. Table 3 summarises the four models and the configuration each of them requires.

4 Implementation

The objective of the implementation is to extract the handwritten contents of each field in a structured format, such as a table or a JSON file, so that the information can be digitised and analysed. The implementation was designed to be form-agnostic: the field schema is user-defined and not hard-coded. All parameters of the pipelines, such as pre- and post-processing methods and prompts, were configured on validation data only. The test split was evaluated once, in a single pass, for all model variants. Fig. 2 compares the step sequence of each implementation.

4.1 The Fine-Tuned Specialist Pipeline

The specialist condition is a staged pipeline. It localises each form field, prepares the resulting crop for recognition, applies the fine-tuned TrOCR model [28], and corrects the transcription against field-level constraints. Table 4 presents the method used in each stage, and Tables S1 and S3 in Supplementary Information give the parameter values. Decoding is greedy throughout, so no reported result depends on beam search.

Known limitations of the line-level pipeline

TrOCR was fine-tuned on line crops of layout A, extracted with the first three stages of this same pipeline. Both the visual characteristics of

[4] https://huggingface.co/datalab-to/chandra-ocr-2

Table 3 The four models compared. *Extractor* denotes whether the field values have to be recovered from the model output by a separate rule-based stage; for TrOCR the values come directly from the segments, and Qwen is prompted to emit them as JSON. *Checkboxes* denotes whether the model itself reports the marked option; in the specialist pipeline they are classified by a rule instead.

	TrOCR	**Qwen**	**Chandra**	**DeepSeek**
Type	Line-level	Instruction VLM	OCR VLM	OCR VLM
Input	Field crop	Full page	Full page	Full page
Adaptation	Fine-tuned	Zero-shot	Zero-shot	Zero-shot
Prompt	—	Own	`ocr`	`markdown`
Output	Plain text	JSON	HTML	Grounded Markdown
Extractor	No	No	Yes	Yes
Checkboxes	No	Yes	Yes	Yes

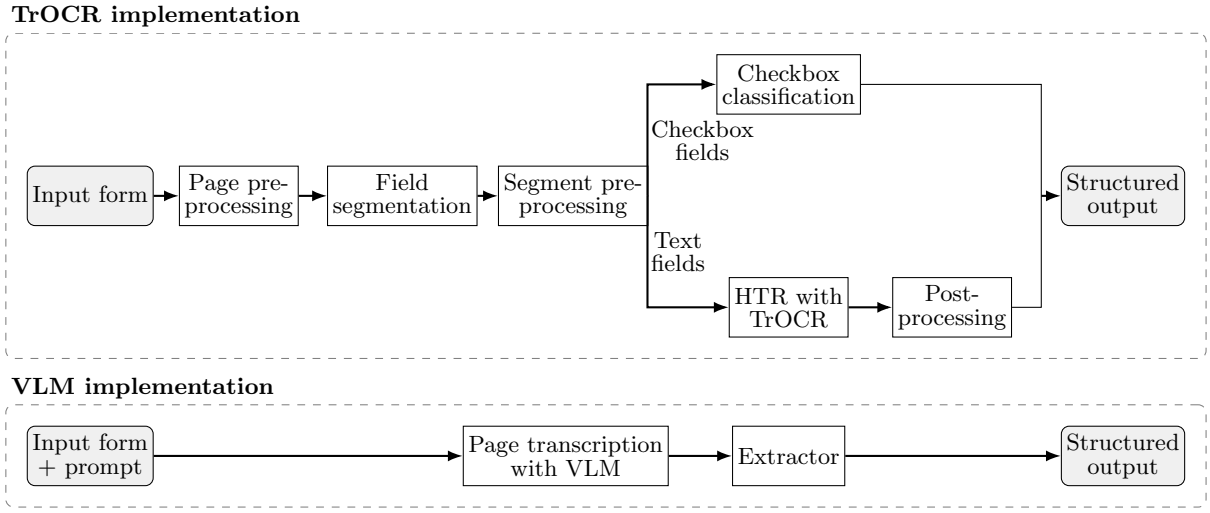


Fig. 2 Comparison of the TrOCR implementation (top) and the VLM implementation (bottom).

layout A and the artefacts generated by the pre-processing may therefore be induced in the model. Moreover, the methods used in the different stages of the pipeline are relatively simplistic, which could impact its performance negatively.

4.2 The Zero-Shot Generalist Baselines

For the VLMs, no pre-processing, segmentation or post-processing was applied. The scanned page is submitted as it is, and the field values are recovered from the model output afterwards. As with TrOCR, decoding is greedy throughout.

Prompting

For instruction VLMs, such as Qwen, the user has to write a prompt that explains the task and the expected output; this can be any task related to interpreting images. OCR-specific models such as Chandra and DeepSeek, in contrast, were developed for the specific task of transcribing documents. Instead of requiring the user to write a prompt, they provide a selection of prompts that specify the task (i.e. transcribing the document and its structure to HTML). The user can, however, provide a custom prompt that differs from this selection. For a more symmetric comparison, the Qwen prompt was also tried for Chandra and DeepSeek. The outputs of these models, however, were either empty or did not transcribe the handwritten text at all. The implementation therefore had to be asymmetric between the two types of models. Qwen received a custom prompt that defines the HTR task and states which fields are to be extracted from the forms. Chandra and DeepSeek received one of their predefined prompts, with no information about the forms

Table 4 Stages of the fine-tuned specialist pipeline.

Stage	Method
Page pre-processing	Skew correction from the minimum-area bounding rectangle of the dominant ink contour of the binarised page.
Field segmentation	User-defined handwritten text fields on a reference page, projected onto every other page by homography from ORB (oriented FAST and rotated BRIEF) keypoints fitted under random sample consensus (RANSAC).
Segment pre-processing	Removal of the printed lines by morphological opening with elongated structuring elements, inpainting of the mask, cropping of the empty white space. For multi-line fields only: row-wise ink profile, candidate lines as local maxima by topographic prominence, merged when the valley between them is too shallow.
Recognition	TrOCR fine-tuned on the training split of layout A, applied to each segment containing text.
Checkbox classification	Comparison of the ink ratio across the sibling boxes of a group.
Post-processing	Field-specific correction rules, such as letter-digit error models and lexical corrections.

themselves. The prompts used are reproduced in Supplementary Information (Sect. S5).

Field extractor

Qwen was prompted to produce JSON, which pairs values with fields directly. Chandra and DeepSeek produced transcriptions of the whole page in HTML and Markdown, respectively. Besides the handwritten text, these transcriptions contain headings, field labels and other structural elements of the forms. These two models therefore required an extractor, which pairs the fields with their values in the raw output. The extractor searches for the field labels in the raw output (e.g. *Name* or *Vorname*) and extracts their values based on a set of rules. Qwen thus receives information about the form in its prompt, whereas Chandra and DeepSeek receive it in the extractor.

Known limitations of the generalist condition

Prompt and extractor were selected on validation data. The condition is therefore zero-shot with respect to the model weights, but not with respect to the system configuration. The extractor logic necessarily differs across models, so a difference between conditions may originate in the extractor as well as in the recognition. This is a common issue in document-level HTR [19].

4.3 Metrics and Evaluation

The main metric is CER, the standard metric in HTR [19]. When a group of fields is measured, for example one form or a group of forms, the edit distances of the individual fields are summed and divided by the summed length of the corresponding references. The reported value is therefore the share of all transcribed characters in the group that are wrong.

Since the forms consist of fields, a field-level metric is additionally adopted: the field error rate (FER), i.e. the proportion of fields whose prediction does not exactly match the ground truth.

Checkbox fields are classified rather than transcribed, so a character-level error rate is not meaningful for them. They are scored by checkbox accuracy (CA), i.e. the proportion of groups for which the selected option matches the reference option.

The perfect form rate (PFR) is the proportion of forms in which all scored fields, text and checkboxes alike, exactly match the ground truth. It is the strictest of the reported metrics and the closest to the deployment question, since a form containing any error has to be corrected by hand, regardless of how small the error is.

RQ1 is answered on the test split of layout A, the layout on which the specialist pipeline was fine-tuned. RQ2 is answered by repeating the same evaluation on the test splits of layouts B

and C. Cross-layout degradation is also measured. For a metric m and a target layout $L \in \{B, C\}$, it is defined as $(m_L - m_A)/m_A$, so positive values denote worse performance than on layout A. Layout A was chosen as the common reference because, by design, it is the least demanding of the three layouts.

4.4 Reproducibility

All experiments were run on a single NVIDIA RTX A6000 GPU with 48 GB of memory. Random seed 42 was fixed for all operations involving sampling, including the assignment of document IDs to splits.

5 Results

In the comparisons of this section, TrOCR is reported after post-processing, and the VLMs are reported on their raw outputs.

Gemini 2.5 Flash appears in the tables and figures of this section in italics, marked with †. As a cloud model, it falls outside the on-premise comparison. It serves only as a commercial reference for what could be achievable without data protection constraints.

5.1 Layout A Performance

Layout A was the layout used to fine-tune TrOCR. Nevertheless, the two best models on this layout are the VLMs Chandra and Qwen, which lie close to each other on all metrics (Table 5). The fine-tuned TrOCR pipeline follows at roughly 1.5 times their CER, and DeepSeek is last on every metric. CA is the only metric on which the specialist matches the two leading VLMs. Of the four compared models, only Qwen and Chandra transcribe any form without a single field error. Fig. 3 gives the per-form CER distributions.

Broken down by field type (Table 6), the gap between the specialist and the two leading VLMs is not uniform. On text fields, the three are close, with TrOCR even slightly ahead in CER. On numeric fields, TrOCR falls behind both. The long-text field, which in these forms is only *comment*, separates them most clearly: by CER, it is the worst field type for TrOCR and the best for Qwen and Chandra. By CER, DeepSeek is last on all three field types.

Table 5 Overall CER, FER, CA and PFR (all in %) for all models, TrOCR after post-processing of its structured fields and the VLMs on the raw outputs. For CER and FER lower is better; for CA and PFR higher is better. Best performance per layout in bold. †: commercial reference, not part of the comparison and excluded from the bold marking.

Layout	Model	CER	FER	CA	PFR
A	TrOCR	6.8	35.4	**100**	0.0
	Qwen	4.7	27.5	**100**	**7.7**
	Chandra	**4.5**	**27.0**	**100**	**7.7**
	DeepSeek	16.1	57.1	28.2	0.0
	Gemini†	*2.7*	*15.6*	*100*	*25.6*
B	TrOCR	12.1	48.5	71.8	0.0
	Qwen	**7.2**	37.3	**100**	**2.6**
	Chandra	**7.2**	**35.2**	94.9	**2.6**
	DeepSeek	20.7	64.6	7.7	0.0
	Gemini†	*4.1*	*19.6*	*100*	*20.5*
C	TrOCR	15.4	53.8	48.7	0.0
	Qwen	6.5	36.8	**100**	5.1
	Chandra	**5.8**	**33.3**	**100**	**7.7**
	DeepSeek	26.1	63.4	12.8	0.0
	Gemini†	*3.3*	*21.0*	*100*	*10.3*

5.2 Cross-Layout Performance

On layouts B and C, Qwen and Chandra remain the two best models, with Chandra ahead on FER on both layouts and on CER on layout C. TrOCR degrades considerably, and DeepSeek remains last (Table 5). The PFR of Qwen drops on both layouts, that of Chandra only on layout B. TrOCR and DeepSeek still transcribe no form without errors. CA separates the models most sharply. Qwen keeps a perfect score on all three layouts, and Chandra drops only slightly on layout B. TrOCR falls from a perfect score on layout A to roughly half on layout C, and DeepSeek remains far behind on both.

Cross-layout degradation, defined in Sect. 4.3, is shown in Fig. 4. For CER, the models fall into two groups. Qwen and Chandra degrade more on layout B than on layout C, whereas TrOCR and DeepSeek show the opposite ordering. Within the second group, the magnitudes differ sharply. DeepSeek's relative degradation is of the same order as that of the other VLMs, while TrOCR more than doubles its CER on layout C, the largest degradation observed. For FER, this grouping does not hold. Apart from Gemini, only

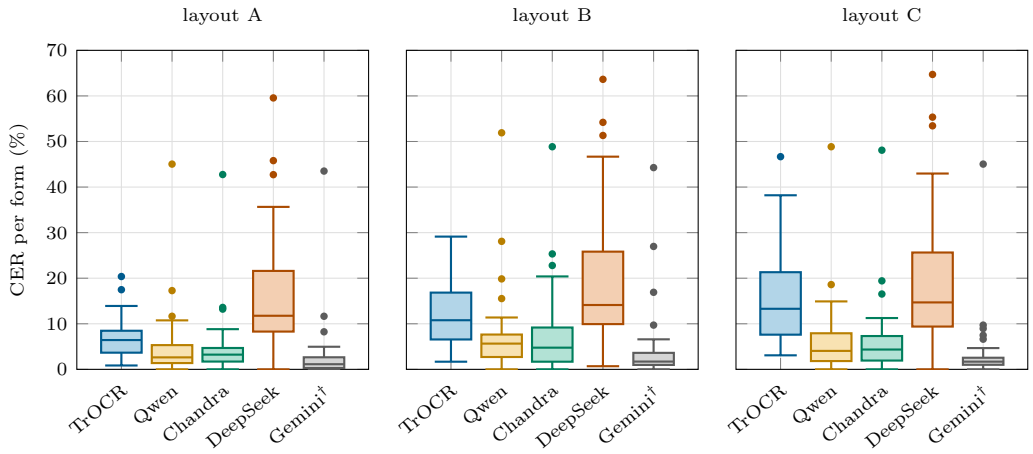


Fig. 3 CER distribution for models across layouts. 1 of 585 form–model points lies above 70% and is not drawn (DeepSeek, layout C). †: commercial reference, not part of the comparison.

Table 6 Performance per field type across layouts. Best performance per layout in bold, for each field type and metric. †: commercial reference, not part of the comparison and excluded from the bold marking.

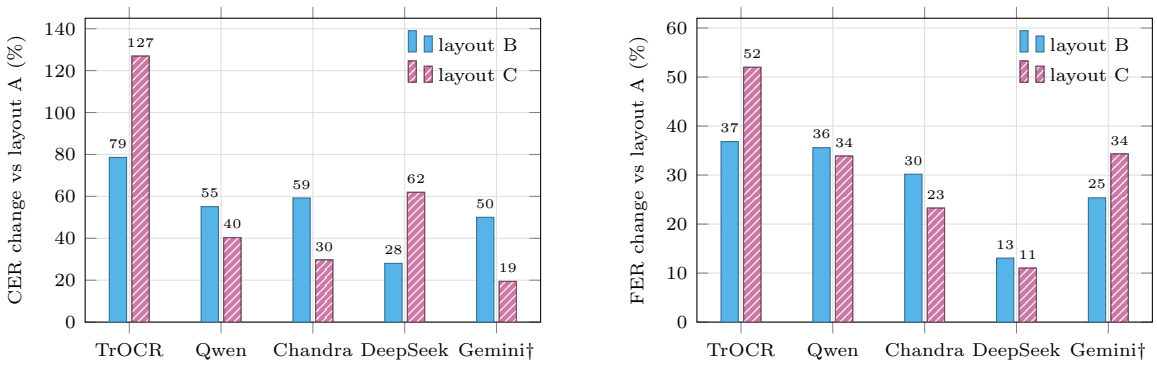
		CER (%)			FER (%)		
Layout	**Model**	**Number**	**Text**	**Long text**	**Number**	**Text**	**Long text**
A	TrOCR	5.3	**5.9**	9.0	26.2	33.9	89.7
	Qwen	3.9	6.7	2.7	22.1	**32.3**	30.8
	Chandra	**3.6**	6.7	**2.5**	**19.5**	34.4	**28.2**
	DeepSeek	11.5	19.2	15.9	44.1	65.1	82.1
	Gemini†	*2.3*	*3.6*	*2.0*	*12.3*	*16.4*	*28.2*
B	TrOCR	9.3	11.4	15.1	40.0	46.7	100.0
	Qwen	7.1	**11.1**	**2.5**	**26.7**	48.7	33.3
	Chandra	**5.4**	11.9	2.6	28.7	**43.6**	**25.6**
	DeepSeek	20.3	28.3	11.6	56.9	70.8	71.8
	Gemini†	*4.7*	*5.7*	*1.7*	*17.4*	*22.6*	*15.4*
C	TrOCR	18.1	12.1	17.3	48.7	49.7	100.0
	Qwen	6.2	**8.6**	4.2	28.7	42.1	51.3
	Chandra	**4.6**	8.7	**3.1**	**25.6**	**38.5**	**46.2**
	DeepSeek	15.4	42.4	14.1	50.8	71.8	84.6
	Gemini†	*2.9*	*4.3*	*2.3*	*15.4*	*23.1*	*38.5*

TrOCR degrades more on layout C than on layout B. DeepSeek shows the smallest relative degradation of the four models on both layouts, and its FER on layout C is lower than on layout B.

5.3 Error Type Decomposition

Extraction errors for OCR VLMs

Chandra and DeepSeek require an extractor to isolate the fields and their values from the raw model output. As a result, handwriting may be recognised correctly by the model but still be counted as an error if the extraction process fails to assign the text to the correct field.



(a) **(b)**

Fig. 4 Cross-layout degradation by model, relative to layout A: **a** CER, **b** FER. Positive values mean worse performance than on layout A. †: commercial reference, not part of the comparison.

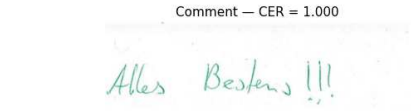
```
<|ref|>text<|/ref|><|det|>[[113, 480, 598,
    529]]<|/det|>
Studiengang: IM WIN BWL Kunst
X ☐ ☐ ☐
```

Fig. 5 DeepSeek's output for the *studies* field of form C-14, using a symbol for marked checkboxes that does not occur in the validation data, so that the extractor does not recover the selected option. A correctly extracted case is shown in Supplementary Information, Fig. S5b.

No extraction errors were detected for Chandra. For DeepSeek, 15 were identified: one on layout A, ten on layout B (a quarter of its test split) and four on layout C. Five of them concern the *studies* field and are caused by the symbol used to mark a checkbox (Fig. 5). One form had all its values wrapped in curly braces (e.g. {*John*}). In the majority of the remaining cases, fields that the model had not recognised received other parts of the document as their value instead of an empty string. An example of an extractor error is form A-60: DeepSeek transcribed the *comment* field correctly, but the extractor did not assign it to the field (Fig. 6).

Unexpected outputs by DeepSeek

Separately from the extraction errors, DeepSeek produced outputs that are recognition failures rather than extraction failures. Fig. 7 shows a degenerate repetition in the *comment* field of form A-78. Fig. S6 in Supplementary Information adds a repeated placeholder in the *email* field of form C-12 and a sequence of non-Latin characters in the *monthly income* field of form C-81.



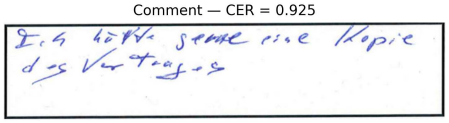

```
GT:   Alles Bestens!!!
Pred:
```

(a)

```
<|ref|>text<|/ref|><|det|>[[400, 833, 690,
    880]]<|/det|>
Alles Besten III.
```

(b)

Fig. 6 Form A-60, where the *comment* field was recognised by DeepSeek but not assigned to the field by the extractor: **a** the *comment* field with empty prediction, **b** excerpt of the raw output for that field.




```
GT:   Ich hätte gerne eine Kopie des Vertrages
Pred: E. G. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4. 4.
```

Fig. 7 Unexpected output by DeepSeek: a degenerate repetition in the *comment* field of form A-78. Two further examples are given in Supplementary Information, Fig. S6.

Field localisation in the specialist pipeline

The segments for TrOCR's pipeline were defined as the space to the right of each field label. Layout C, however, has no boxes or lines that constrain where the fields are filled out. CER can therefore also depend on how well such a rigid, template-matching field localisation works on a layout whose fields have no delimiting boxes or

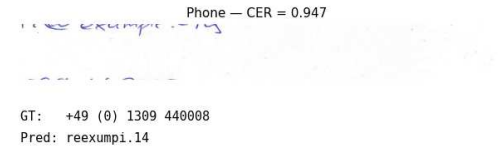


Fig. 8 A field where TrOCR's localisation failed because of layout C's unconstrained fields: the *phone* field of form C-36. Two further examples are given in Supplementary Information, Fig. S7.

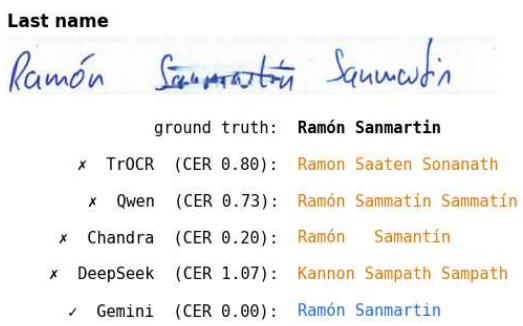


Fig. 9 Struck-out text in the *last name* field of form A-90. The *first name* field of the same form is shown in Supplementary Information, Fig. S8.

lines. Fig. 8 shows an example of a poorly cropped field, and Fig. S7 in Supplementary Information shows two more. Similar failures happened less often on layout B, where some participants wrote outside the provided fields.

Forms with struck-out text

Some participants corrected themselves while filling out the forms. Fig. 9 and Fig. S8 in Supplementary Information show form A-90. No on-premise model transcribes the two affected fields perfectly, but Chandra omits the struck-out part, while the other three transcribe it as content.

5.4 Post-Processing

Post-processing was defined for TrOCR, based on the errors observed on the validation data of layout A. These rules, however, can also be applied to the VLMs. Table S4 in Supplementary Information compares raw and post-processed results. At the aggregate level, post-processing improves every model on every layout, with consistently larger gains on FER than on CER.

At field level, however, post-processing also broke correctly recognised values. In form A-67, TrOCR predicted the correct city *Schwarzenberg*, which the lexicon corrector changed to *Schwarzenbek*. In form A-90, the street *Alameda Anastasio Ariño* was recognised as *Alameda Anastasio Arino*. It was then changed to *Alameda Anastasio Aring*, matching the common German street suffix *-ring*.

5.5 Commercial Reference

As a reference for the performance achievable without data protection constraints, the test splits were also processed by Gemini 2.5 Flash. The model was used through its free web interface, without API calls or a similar implementation. Its results are reported in the rows marked with † in Tables 5 and 6, and in Figs. 3 and 4. It is not part of the comparison in Sects. 5.1 and following, since the on-premise requirement excludes cloud services. Gemini achieves the lowest CER and FER and the highest PFR on all three layouts, and it matches the perfect CA of Qwen. Broken down by field type, it is ahead of every compared model on all three field types and all three layouts. The only exception is the long-text field on layout A, where its FER equals that of Chandra.

6 Discussion

6.1 Dataset and Evaluation Discussion

Four properties of the dataset limit what the numbers can support. First, the fields are independent of each other. As an example, the combination of street, house number, city and postal code is not necessarily a real address. The cross-field validation that a deployed system would perform on authentic data is therefore closed here by construction. The post-processing evaluated in this work is thus weaker than what the same architecture would allow. Second, the test split contains 39 document IDs, amounting to 468 scored fields per layout. This is enough to separate models that differ by several percentage points of CER, but not to rank models that differ by a few tenths. Third, the forms are modern, clean and scanned under controlled conditions, so degraded documents and photographed input are absent. The forms also consist of one page only, so multi-page forms are not evaluated within this scope.

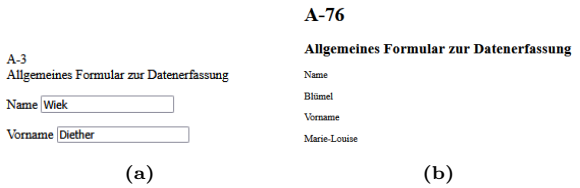
Fourth, two participants each filled in two document IDs, one in cursive and one in block letters. Since the collection was anonymous and no mapping between document ID and participant exists, writer-disjoint splits cannot be guaranteed for these four document IDs. With at most two of 95 writers affected, the effect on the results is likely small.

6.2 Layout A Discussion

The specialist pipeline showed acceptable performance only on layout A. This was the layout used to fine-tune both the model and the whole pipeline, including its post-processing. Two VLMs, however, performed better on layout A than this pipeline: Qwen and Chandra. It is worth noting how asymmetric the comparison was. The specialist TrOCR saw 467 line images of exactly this layout, cropped by exactly the pipeline that was later used at inference. Its output was corrected, on every field except *comment*, by rules derived from its own errors on this layout. The two VLMs, in contrast, received no gradient update at all, saw no training samples in their prompt and had their output left uncorrected. Besides fine-tuning, the asymmetry also concerns the stage at which the pipelines receive information about the form. For TrOCR, the fields, their types and their bounding boxes are used to segment the individual handwritten text lines. Qwen receives the expected fields and types in the prompt; Chandra and DeepSeek receive them in the extractor. DeepSeek, on the other hand, appears to be poorly suited to handwriting, since its performance was subpar even on the simplest layout. In its case, this cannot be attributed to the Markdown extractor, since extraction errors on layout A were negligible.

Since our TrOCR model was fine-tuned primarily on short, structured handwritten lines from the *HGAF* dataset, it performs well on the short text fields, where the three models are practically tied. This result, however, is partly owed to the post-processing. The lexicon and format corrections act mostly on these fields, and without them, the CER of TrOCR on text fields lies above that of both VLMs. On numeric fields, the VLMs remain ahead even after TrOCR's post-processing, and on the long-text field, TrOCR performs clearly worse. This could be attributed to the *comment* field contributing far fewer lines to the training data than the ten single-line fields together. Most likely, however, it is because *comment* is the only field whose crops depend on the automatic line segmentation. Its FER on this field is therefore a combination of recognition difficulty, segmentation errors and training imbalance. We believe that another factor hampering the performance of this pipeline is the simplistic approaches taken in some of its steps. Improving the removal of the printed lines (for example, with a stroke-reconstruction step as in Bai and Huo [6], Yu and Jain [40]) or the line segmentation of multi-line fields would provide cleaner images for TrOCR, or for any other line-level model.

For Chandra and DeepSeek, key-value pairs can only be recovered from the raw output with a good extractor. This is a difficult task, since the same type of form can yield different output structures, as shown in Fig. 10. Although not specialised in OCR, the more generalist Qwen achieved results comparable to Chandra, with the advantage of not requiring such an extractor, since the output format was induced in the prompt.



(a) **(b)**

Fig. 10 Chandra's HTML output for the first two fields of two forms of layout A, rendered: **a** form A-3, where the fields to complete are returned as input elements (`<input>`); **b** form A-76, where the same fields are returned as ordinary paragraphs (`<p>`).

6.3 Cross-Layout Performance Discussion

In terms of cross-layout performance, all four models performed better on the simplest layout (A) and worse on layouts B and C. For TrOCR, whose model and pipeline were developed on layout A, this degradation measures its generalisation to unseen layouts. For the VLMs, which were not trained on any of the three layouts, it measures how sensitive they are to the visual structure of the form.

For TrOCR's pipeline, an important bottleneck is the localisation of the segments with

handwriting, not the recognition itself. Layout C has no boxes or lines for its fields, which leaves the participants free to choose where to write. If a participant wrote below the label instead of next to it, the crop passed to TrOCR contained the wrong region or no ink at all, as shown in Fig. 8 for form C-36. This is consistent with layout C being the hardest layout for TrOCR. A second factor for the degradation is that TrOCR was fine-tuned on crops of layout A, extracted with the same pipeline stages that were then used for the final tests. This biases the model towards these preprocessing techniques and towards the artefacts they generate on the images, which are specific to the box geometry of layout A. This corresponds to the visual divergence described by Garrido-Munoz and Calvo-Zaragoza [18], with the difference that here it is introduced by the pipeline itself and not by the source material. The post-processing, on the other hand, was also derived from layout A, but improves TrOCR by a comparable margin on all three layouts. The degradation therefore cannot be attributed to it.

For the VLMs, the difficulty is harder to pinpoint. Both Qwen and Chandra degrade more on layout B than on layout C, although layout C offers the least visual guidance of the three. Layout B is the compact tabular arrangement, so two explanations are possible. Either the tabular structure imposes a reading order that the model resolves incorrectly, or the smaller cells simply place more ink closer together and increase the confusion between neighbouring fields. Distinguishing them would require a fourth layout that is tabular but generously spaced, which the *HGAF* dataset does not contain.

The per-form distributions in Fig. 3 add a second dimension to the comparison. The advantage of Qwen and Chandra lies not only in their average error, but also in their consistency across forms. The longer upper tails of TrOCR and DeepSeek correspond to individual forms on which they fail far more than on the rest. This is consistent with the failure modes identified in the error analysis: wrongly localised fields for TrOCR (Fig. 8) and degenerate outputs for DeepSeek (Fig. 7). For a deployed system, these outliers matter as much as the average, since every such form has to be reviewed manually.

6.4 General Performance Factors and Implications

Checkboxes

On layout A, DeepSeek is the only model with poor performance. Its raw output before extraction shows two causes. Partly, the model failed to recognise marked checkboxes. Partly, the extractor failed to recognise marked checkboxes that the model had recognised correctly. On layouts B and C, the CA of TrOCR's pipeline drops as well, which is likely because its parameters were fitted to layout A.

Struck-out text

Corrections made by the writer are a known difficulty for recognition systems [9]. As shown in Fig. 9, no model transcribes the name fields of A-90 perfectly. Chandra, however, ignores the struck-out part, while the other models transcribe it as content. A page-level model that has learned document conventions can treat a deletion as such.

Language bias

Neither Qwen nor Chandra presented difficulties with German texts. This is consistent with Crosilla et al. [12], who report the strongest degradation for languages and scripts far from modern English. For DeepSeek, this data does not allow us to state whether this is a language bias or a general weakness, since its performance is also poor on numeric fields, where language priors play no role.

Implications for the implementation

The decisive difference between the two conditions is not only the accuracy but also the effort required per new form design. For every new layout, the specialist pipeline needs the definition of the segments with handwritten text and of the checkbox regions, and a check that line removal and line segmentation behave correctly on the new geometry. Judging by the results on layouts B and C, it also needs a new fine-tuning. Its post-processing narrows the gap to the VLMs, but does not close it, even though the VLMs were evaluated without any correction of their output. Chandra needs a new extractor per layout, since its HTML structure varies even within one layout. Qwen needs neither: the output schema is

fixed in the prompt, the returned JSON pairs values with fields directly, and no form of any layout failed to parse. Together with the best or second-best score on every metric and a perfect CA on all three layouts, this makes Qwen the recommended default of the implementation when several form designs have to be supported. DeepSeek, on the other hand, cannot be recommended for this task. Its recognition quality, its output stability and its checkbox handling all fall short, and it produced the largest number of extraction errors.

Comparison with a commercial model

No open-weight model surpassed Gemini's performance on the *HGAF* dataset. This is in accordance with similar HTR benchmarks [12]. Gemini also required no extraction, since the same prompt as for Qwen could be used. Like Chandra, it is able to handle struck-out text. In the absence of data protection restrictions, a commercial model may therefore be preferable on accuracy grounds: within the evaluated configurations, Gemini 2.5 Flash achieved the strongest results, with an equal or better performance on every metric. The web-interface evaluation, however, does not offer the reproducibility or deployment control of the local pipelines.

7 Conclusion

This work compared a fine-tuned line-level HTR pipeline with three zero-shot vision-language models on handwritten German administrative forms, under the constraint that every model has to run on-premise. To make the comparison possible, the *HGAF* dataset was created. It consists of 291 forms, filled in by 95 writers across three layouts that share an identical field inventory and identical synthetic contents. In addition, a processing pipeline was implemented that integrates both conditions.

Regarding RQ1, the two leading VLMs, Qwen and Chandra, outperformed the specialist pipeline on layout A, the same layout the specialist was fine-tuned on. The comparison was asymmetric in the specialist's favour: in addition to the fine-tuning, its outputs were also post-processed. The performance gap is not uniform across the field types. On short text fields the three pipelines are practically tied, and on numeric fields the specialist is close behind. The specialist falls clearly behind in the multi-line *comment* field. Therefore, the results indicate that a fine-tuned line-level pipeline remains competitive on short text and numeric fields, despite being outperformed overall by the strongest VLMs.

Regarding RQ2, the result holds in its ordering but not in its magnitude. Transcription accuracy degraded for every model on the two more complex layouts, while the ranking of the models remained unchanged. The specialist degraded most, the two leading VLMs noticeably less. Form layout therefore affects recognition accuracy across all four systems. Which layout properties drive the effect remains open, especially for VLMs.

Two findings concern deployment rather than accuracy. First, no configuration evaluated here is sufficient for unsupervised digitisation of these forms. The best PFR of the on-premise models was 7.7%, and two of the four models did not transcribe a single form without an error. Any of these systems reduces manual transcription effort, but none removes the need for a human review step. Second, an OCR-specialised VLM does not solve key information extraction on its own. It transcribes the whole document, so the field values still have to be recovered from its output. This required an extractor that had to be adjusted for each of the three layouts, even though the fields were identical between them. Qwen was the exception: inducing the output schema in the prompt made the extractor unnecessary, and no form of any layout failed to parse.

Several directions follow from this work. (1) The first concerns benchmarking. Neither of the two vision-language OCR benchmarks consulted in this work [17, 30] contains a handwritten form task. The reported strength of these models on document understanding therefore does not translate into evidence for field-level transcription of handwriting. Including such a task would close this gap. (2) The second concerns the layout factor itself. Three layouts are enough to establish that layout matters for form HTR tasks, but not enough to determine the role of the different layout properties. (3) A third direction is the identification of erroneously transcribed fields. Adding confidence scores, or flagging recognised contents based on pattern matching, could make these models easier and more efficient to deploy. This would reduce the effort for the person who

checks and corrects the model outputs. (4) Finally, the specialist condition was implemented with deliberately simple pre-processing. Repeating this comparison with a learned line segmenter and a less destructive line removal would show whether the gap measured here is a property of line-level HTR or of the stages built around it.

Supplementary information. A document called Supplementary Information accompanies this article. It contains full-page reproductions of the three layouts and of the instruction for completion (Figs. S1–S4), the parameter values of the specialist pipeline (Table S1), the post-processing rules and thresholds together with the raw and post-processed results of every model (Tables S2–S4), the complete set of qualitative error examples (Figs. S5–S8), and the prompts passed to the three vision-language models (Sect. S5).

Acknowledgements. The authors thank the students of the Faculty of Management Science and Engineering of Karlsruhe University of Applied Sciences for filling in the forms of the *HGAF* dataset, Karlsruhe University of Applied Sciences for providing the hardware to run the models, and Luis Bast for helping to design and create the dataset.

Declarations

Funding. No funding was received for conducting this study.

Competing interests. The authors have no competing interests to declare that are relevant to the content of this article.

Consent. Participants were informed of the intended use of the forms. By filling out the forms, they consented to the anonymous use of their handwriting for research on handwriting recognition and to its publication (Fig. S1 of Supplementary Information). No personal data were collected: the contents of all fields were synthetically generated, the collection was anonymous, and no mapping between document ID and participant exists.

Data availability. The *HGAF* dataset was made available on a public repository; see https://zenodo.org/records/22934701.

Code availability. The *HGAF* splits, the evaluation scripts and the raw outputs of all models are available under https://github.com/tkalmb/hgaf-evaluation.

Author contribution. Tobias Kalmbach: designed the study, designed and collected the dataset, implemented the pipelines, performed the experiments, and wrote the manuscript. Andreas Wagner: designed the study, designed and collected the dataset, reviewed and edited the manuscript, and supervised all tasks.

Use of generative artificial intelligence. Anthropic's Claude and Microsoft's Copilot were used to assist with programming, writing and typesetting (including LaTeX formatting, figures and tables), for literature search, and to generate roughly 30% of the synthetic field contents that participants were asked to copy onto the forms by hand.

References

[1] AlKendi W, Gechter F, Heyberger L, et al (2024) Advancements and Challenges in Handwritten Text Recognition: A Comprehensive Survey. Journal of Imaging 10(1):18. https://doi.org/10.3390/jimaging10010018

[2] Ambata LU, Go JDO, MacaranasIII PQ, et al (2025) Automated Extraction of Handwritten Text from Forms Using Advanced Handwritten Text Recognition (HTR). In: 2025 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), pp 419–424, https://doi.org/10.1109/I2CACIS65476.2025.11100803

[3] Andrés J, Wall C, Tarride S, et al (2025) The PARES Database: Information Extraction over Historical Parish Records. International Journal on Document Analysis and Recognition (IJDAR) 28(3):377–389. https://doi.org/10.1007/s10032-025-00531-z

[4] Aradillas Jaramillo JC, Murillo-Fuentes JJ, M. Olmos P (2018) Boosting Handwriting Text Recognition in Small Databases with

Transfer Learning. In: 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp 429–434, https://doi.org/10.1109/ICFHR-2018.2018.00081

[5] Bai S, Cai Y, Chen R, et al (2025) Qwen3-VL Technical Report. https://doi.org/10.48550/arXiv.2511.21631

[6] Bai ZL, Huo Q (2004) Underline detection and removal in a document image using multiple strategies. In: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., pp 578–581 Vol.2, https://doi.org/10.1109/ICPR.2004.1334314, iSSN: 1051-4651

[7] Bitkom e.V. (2024) 4 von 10 Unternehmen arbeiten überwiegend papierlos. URL https://www.bitkom.org/Presse/Presseinformation/4-von-10-Unternehmen-arbeiten-papierlos

[8] Burghardt T, Alzin A (2026) ScaDS.AI German Line- and Word-Level Handwriting Dataset. https://doi.org/10.5281/ZENODO.18301531

[9] Chaudhuri BB, Adak C (2017) An approach for detecting and cleaning of struck-out handwritten text. Pattern Recognition 61:282–294. https://doi.org/10.1016/j.patcog.2016.07.032

[10] Chen S, Guo X, Li Y, et al (2025) Ocean-OCR: Towards General OCR Application via a Vision-Language Model. https://doi.org/10.48550/arXiv.2501.15558

[11] Chung YHM, Choi D (2026) Fine-tuning vision–language models as OCR systems for low-resource languages: A case study of Manchu. Computational Humanities Research 2:e20. https://doi.org/10.1017/chr.2026.10042

[12] Crosilla G, Klic L, Colavizza G (2025) Benchmarking large language models for handwritten text recognition. Journal of Documentation 81(7):334–354. https://doi.org/10.1108/JD-03-2025-0082

[13] Davis B, Morse B, Cohen S, et al (2019) Deep Visual Template-Free Form Parsing. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp 134–141, https://doi.org/10.1109/ICDAR.2019.00030, iSSN: 2379-2140

[14] Dosovitskiy A, Beyer L, Kolesnikov A, et al (2021) An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: International Conference on Learning Representations (ICLR), URL https://openreview.net/forum?id=YicbFdNTTy

[15] Fischer A, Keller A, Frinken V, et al (2012) Lexicon-free handwritten word spotting using character HMMs. Pattern Recognition Letters 33(7):934–942. https://doi.org/10.1016/j.patrec.2011.09.009

[16] Flanagan PA (2016) NIST Handprinted Forms and Characters - NIST Special Database 19. https://doi.org/10.18434/T4H01C

[17] Fu L, Kuang Z, Song J, et al (2025) OCRBench v2: An Improved Benchmark for Evaluating Large Multimodal Models on Visual Text Localization and Reasoning. In: Advances in Neural Information Processing Systems 38 (NeurIPS 2025), Datasets and Benchmarks Track, https://doi.org/10.52202/085713-3253

[18] Garrido-Munoz C, Calvo-Zaragoza J (2025) On the Generalization of Handwritten Text Recognition Models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 15275–15286, URL https://openaccess.thecvf.com/content/CVPR2025/html/Garrido-Munoz_On_the_Generalization_of_Handwritten_Text_Recognition_Models_CVPR_2025_paper.html

[19] Garrido-Munoz C, Rios-Vila A, Calvo-Zaragoza J (2026) Handwritten Text Recognition: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 48(4):4367–4387. https://doi.org/10.1109/TPAMI.2025.3646002

[20] Grabowski H (2023) Intelligent Character Recognition of Handwritten Forms with Deep Neural Networks. In: Cavallucci D, Livotov P, Brad S (eds) Towards AI-Aided Invention and Innovation, vol 682. Springer Nature Switzerland, Cham, p 81–94, https://doi.org/10.1007/978-3-031-42532-5_6, series Title: IFIP Advances in Information and Communication Technology

[21] Grabowski H (2026) Towards Fully Automated Exam Grading: Fairness-Aware Recognition of Handwritten Answers with Foundation Models. https://doi.org/10.48550/arXiv.2606.11477

[22] Hodel T, Schoch D, Schneider C, et al (2021) General Models for Handwritten Text Recognition: Feasibility and State-of-the Art. German Kurrent as an Example. Journal of Open Humanities Data 7(0). https://doi.org/10.5334/johd.46

[23] Jaume G, Ekenel HK, Thiran JP (2019) FUNSD: A Dataset for Form Understanding in Noisy Scanned Documents. In: 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW), pp 1–6, https://doi.org/10.1109/ICDARW.2019.10029

[24] Kalmbach T, Bast L, Wagner A (2026) HGAF: Handwritten German Administrative Forms. https://doi.org/10.5281/zenodo.22934701

[25] Kass D, Vats E (2022) AttentionHTR: Handwritten Text Recognition Based on Attention Encoder-Decoder Networks. In: Uchida S, Barney E, Eglin V (eds) Document Analysis Systems, vol 13237. Springer International Publishing, Cham, p 507–522, https://doi.org/10.1007/978-3-031-06555-2_34, series Title: Lecture Notes in Computer Science

[26] Lauar F, Laurent V (2024) Spanish TrOCR: Leveraging Transfer Learning for Language Adaptation. https://doi.org/10.48550/arXiv.2407.06950

[27] Levenshtein VI (1966) Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. Soviet Physics Doklady 10(8):707–710. Translated from Doklady Akademii Nauk SSSR, 163(4):845–848, 1965

[28] Li M, Lv T, Chen J, et al (2023) TrOCR: Transformer-Based Optical Character Recognition with Pre-trained Models. Proceedings of the AAAI Conference on Artificial Intelligence 37(11):13094–13102. https://doi.org/10.1609/aaai.v37i11.26538

[29] Likforman-Sulem L, Zahour A, Taconet B (2007) Text line segmentation of historical documents: a survey. International Journal of Document Analysis and Recognition (IJDAR) 9(2):123–138. https://doi.org/10.1007/s10032-006-0023-z

[30] Liu Y, Li Z, Huang M, et al (2024) OCRBench: on the hidden mystery of OCR in large multimodal models. Science China Information Sciences 67(12):220102. https://doi.org/10.1007/s11432-024-4235-6

[31] Marti UV, Bunke H (2002) The IAM-database: an English sentence database for offline handwriting recognition. International Journal on Document Analysis and Recognition 5(1):39–46. https://doi.org/10.1007/s100320200071

[32] Nguyen TTH, Jatowt A, Coustaty M, et al (2021) Survey of Post-OCR Processing Approaches. ACM Computing Surveys (CSUR) 54(6):124:1–124:37. https://doi.org/10.1145/3453476

[33] Pather N, Fouché J, Mundia S, et al (2026) From Handwriting to Structured Data: Benchmarking AI Digitisation of Handwritten Forms. https://doi.org/10.48550/arXiv.2604.16504

[34] Retsinas G, Sfikas G, Gatos B, et al (2022) Best Practices for a Handwritten Text Recognition System. In: Uchida S, Barney E, Eglin V (eds) Document Analysis Systems, vol 13237. Springer International Publishing, Cham, p 247–259, https://doi.org/10.1007/978-3-031-06555-2_17, series Title: Lecture Notes in Computer Science

[35] Ströbel PB, Hodel T, Boente W, et al (2023) The Adaptability of a Transformer-Based OCR Model for Historical Documents. In: Coustaty M, Fornés A (eds) Document Analysis and Recognition – ICDAR 2023 Workshops, vol 14193. Springer Nature Switzerland, Cham, p 34–48, https://doi.org/10.1007/978-3-031-41498-5_3, series Title: Lecture Notes in Computer Science

[36] Tikhonov A, Loew L, Matić-Chalkitis M, et al (2022) Multilingual Handwritten Text Recognition (MultiHTR) or Reading Your Grandma's Old Letters in German, Russian, Serbian, and Ottoman Turkish with Artificial Intelligence. In: Schwan A, Thomson T (eds) The Palgrave Handbook of Digital and Public Humanities. Springer International Publishing, Cham, p 215–233, https://doi.org/10.1007/978-3-031-11886-9_12

[37] Wei H, Sun Y, Li Y (2026) DeepSeek-OCR 2: Visual Causal Flow. https://doi.org/10.48550/arXiv.2601.20552

[38] Xu Y, Lv T, Cui L, et al (2022) XFUND: A Benchmark Dataset for Multilingual Visually Rich Form Understanding. In: Findings of the Association for Computational Linguistics: ACL 2022. Association for Computational Linguistics, Dublin, Ireland, pp 3214–3224, https://doi.org/10.18653/v1/2022.findings-acl.253

[39] Yousef M, Hussain KF, Mohammed US (2020) Accurate, data-efficient, unconstrained text recognition with convolutional neural networks. Pattern Recognition 108:107482. https://doi.org/10.1016/j.patcog.2020.107482

[40] Yu B, Jain A (1996) A generic system for form dropout. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(11):1127–1134. https://doi.org/10.1109/34.544084

[41] Zottin S, De Nardin A, Piciarelli C, et al (2026) U-DIADS-TL: a novel dataset for text line segmentation in historical manuscripts. International Journal on Document Analysis and Recognition (IJDAR) 29(2):647–658. https://doi.org/10.1007/s10032-026-00585-7

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- hgafijdarSupplementaryInformation.pdf