Bachelorthesis

Comparison of language-specific HTR models

“Does the language of the training corpus affect the performance of a handwritten text recognition (HTR) model on crosslingual settings?”

submitted on 10.07.2025
by Sophie Zach, 2622574

First examiner: Prof. Dr. Tessa Taefi Touridocht
Second examiner: Prof. Dr. Sabine Schumann

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN HAMBURG
Department Medientechnik
Finkenau 35
20081 Hamburg

Abstract

This thesis investigates the influence of training language on the performance of handwriting text recognition (HTR) models. Two separate Vision Transformer-based models were trained using datasets in different languages, one with English data (IAM dataset), and another with German data (fhswf/german_handwriting). Both models were evaluated on their native test sets as well as on a cross-lingual test set to assess generalization and linguistic robustness.

Quantitative evaluation using Character Error Rate (CER) and Word Error Rate (WER) shows a clear degradation in recognition performance when models are tested on a language different from their training set. This highlights the sensitivity of HTR models to language-specific features, even when based on language-agnostic decoding mechanisms like Connectionist Temporal Classification (CTC). A qualitative error analysis was conducted to illustrate how specific types of language-dependent character sequences contribute to recognition failures. Furthermore, a pipeline for n-gram-based error attribution on character level was implemented to explore whether misrecognitions correlate with language-dominant character patterns.

Although the n-gram analysis could not be fully utilized due to insufficient cross-lingual performance, the results were discussed in the Appendix and the implemented tools remain available for future experimentation. The findings underscore the need for either multilingual training strategies or language-specific adaptation in practical HTR systems.

The code is publicly available at: https://github.com/Mir0da/HTR-VT_Bachelor

The german trained model is available at: https://huggingface.co/Mir0da/HTR-VT-german

The english trained model is available at: https://huggingface.co/Mir0da/HTR-VT-english

Contents

List of abbreviations

BLSTM	Bidirectional Long Short-Term Memory
CER	Character Error Rate
CNN	Convolutional Neural Networks
CRNN	Convolutional Recurrent Neural Networks
CSV	Comma-separated values
CTC	Connectionist Temporal Classification
GMMs	Gaussian Mixture Models
HMMs	Hidden Markov Models
HTR	Handwritten Text Recognition
LM	Language Model
LSTM	Long short-term memory
NLP	Natural Language Processing
OCR	Optical Character Recognition
RNN	Recurrent Neural Networks
TrOCR	Transformer-based Optical Character Recognition
ViT	Vision Transformer
ViT-CTC	Vision Transformers with Connectionist Temporal Classification
WER	Word Error Rate

List of figures

Unless otherwise stated, the images are based on own data and recordings.

List of tables

List of formulas

Note on the Use of AI Tools

During the preparation of this thesis, generative AI tools were used to support various aspects of the work. In particular, large language models (LLMs) were employed to assist with code development and debugging, as well as with the formulation and refinement of English academic prose. All conceptual decisions, experimental designs, and evaluations were made independently by the author.

1 Introduction

Handwritten Text Recognition (HTR) has undergone major progress in recent years, largely due to advances in deep learning architectures and the availability of large-scale labeled datasets. Models based on Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and more recently Transformers have significantly improved recognition performance in both printed and handwritten text scenarios (Graves *et al.*, 2009; Kang *et al.*, 2020; Diaz *et al.*, 2021; Ansari *et al.*, 2022; Fujitake, 2023; Li *et al.*, 2025).
These methods are widely applied in domains such as digital archiving, historical document analysis to make them more accessible for research and education, personal assistance, and automation of administrative processes and have the goal to develop universal architectures that are suitable for different use cases (Neat *et al.*, 2019; Diaz *et al.*, 2021; Dasari and Mehta, 2023; Koch *et al.*, 2023).

Despite these advances, HTR systems still face challenges when applied across different languages. Most state-of-the-art models are trained on monolingual datasets and implicitly encode language-specific patterns during training (Baek *et al.*, 2019; Diaz *et al.*, 2021). These patterns include statistical distributions of characters, frequent n-gram sequences, and orthographic conventions, all of which differ considerably between languages such as English and German. For instance, German exhibits a higher frequency of compound words and character trigrams such as "sch", whereas English has dominant bigrams like "th" as seen in later analysis (Hecht, Riedler and Backfried, 2002).

This thesis investigates how the language of the training data affects the performance of HTR models when applied to monolingual test sets. Specifically, it compares two Transformer-based models trained on different corpora, one on German, one on English, using a shared architecture inspired by the ViT-CTC pipeline proposed by Li *et al.*, 2025, which combines a Vision Transformer with a Connectionist Temporal Classification decoder. Both models are evaluated on fixed test sets in German and English, to assess their recognition accuracy in both in-language and cross-language scenarios.

Prior research has shown that without appropriate adaptation, cross-lingual transfer of HTR models is often limited, and that usage of a more diverse training data can be more efficient than expanding the number of training examples (Baek *et al.*, 2019; Diaz *et al.*, 2021).

Although a structured error analysis focusing on language-dominant n-gram sequences was initially planned, this component was deprioritized due to insufficient model performance in the cross-lingual setting. However, the complete implementation is included in the Appendix for future reuse or replication.

1.1 General Context and Motivation

A central motivation for this thesis emerged during the early stages of a practical software project that required integrating an HTR model into a text extraction pipeline. While numerous pretrained models for English handwriting were readily available, high-quality open source models for German were comparatively scarce. This raised a fundamental question: does it make a difference which language an HTR model was trained on, assuming both languages use the same script?

At first glance, the question may appear marginal. Since both German and English use the Latin alphabet, it might seem sufficient to fine-tune an English model to account for language-specific characters such as Umlauts. However, the issue quickly revealed itself to be more complex. Discussions with supervising instructors and practical observations suggested that linguistic factors, such as frequent n-gram patterns and writing conventions, could influence how individuals from different language backgrounds form characters.

From these observations emerged a more structured hypothesis: frequent exposure to specific character sequences, so-called n-grams, may shape handwriting patterns over time. These writing patterns, when encoded in training corpora, may affect how well a model generalizes to sequences in other languages. For example, someone accustomed to writing “th” in English may develop ligature-like structures that rarely appear in German handwriting. A model trained solely on German data may therefore struggle to recognize “th” in English inputs and vice versa.

Linguistic characteristics such as dominant n-gram patterns, compounding behavior, and character-level statistics are known to differ substantially between languages, even within the same script family (Hecht, Riedler and Backfried, 2002; Baek *et al.*, 2019). These differences might influence both, the way characters are written and how well models generalize to unseen language-specific sequences.

In many state-of-the-art systems, language modeling plays a crucial role in prediction accuracy. Many deep learning-based models implicitly learn the statistical regularities of a language, including typical character transitions and orthographic norms during training (Fujitake, 2023).
As a result, trained models often capture language-specific visual and sequential priors, even without explicit language modeling components such as lexicons or dictionaries (Diaz *et al.*, 2021).
This becomes particularly relevant in scenarios where little training data is available in the target language. For example, while high-quality English handwriting datasets are widely accessible (e.g., IAM (Marti and Bunke, 2002)), comparable publicly available datasets for contemporary German handwriting are less common. Practitioners thus often resort to models trained on English corpora, hoping for cross-lingual transfer.

However, a model trained in one language might struggle to recognize typical structures of another, even if characters overlap. This presents practical consequences for users working in multilingual or resource-limited contexts.

This project is part of a Bachelor thesis in the Media Informatics program at HAW Hamburg and will be presented in the form of a final documentation. The experimental code and data from analyses is hosted on GitHub and the trained models on Hugging Face Hub for reproducibility.

1.2 Goal and Scope of Work

This thesis aims to examine how the training language of an HTR model affects its ability to generalize across languages, initially with a particular focus on language-dominant n-gram patterns. For this purpose, two structurally identical HTR models are trained based on Vision Transformers with Connectionist Temporal Classification (ViT-CTC), following the approach introduced by (Li *et al.*, 2025). The official training implementation provided by the authors is adapted for this comparative experiment and to handle both datasets.
The two models are trained separately on:

- the IAM dataset (available at https://fki.tic.heia-fr.ch) for English
- an the fhswf/german_handwriting dataset (available at Hugging Face) for German

Each dataset is divided into a training, validation and test split. This setup enabled not only the evaluation of each model on its respective language but also a cross-lingual evaluation, wherein each model is tested on the other language's test set.

The core hypothesis is that frequent exposure to language-specific n-grams during training influences the model's ability to generalize across languages. For example, n-grams such as "th" are frequent in English but rare in German, whereas combinations like "sc" are dominant in German. To underline this claim, extracts of the n-gram analysis, further declared in 3.1.1, of the data sets are shown in Table 1.1.

n-gram	count_ german	count_ english	rel_ german	rel_ english	ratio	diff	category	absolute count
th	596	11692	~0.00153	~0.03024	~0.05072	11096	english-dominant	12288
sc	3524	786	~0.00907	~0.00203	~4.44090	2738	german-dominant	4310

Table 1.1: example of dominant n-grams in German and English
Source: own analysis

The intuition is that unfamiliar n-grams in the target language may lead to elevated recognition errors if they were underrepresented in the training language. This hypothesis is supported by prior research highlighting the role of language models in enhancing recognition accuracy by integrating linguistic priors into the decoding process (Graves *et al.*, 2009; Ströbel *et al.*, 2022; Fujitake, 2023). Hecht, Riedler and Backfried (2002) further emphasize that German orthography exhibits a high degree of morphological complexity and compounding, which complicates statistical modeling of character sequences. If a model trained exclusively on German struggles to accurately predict "th" sequences in English test

data, it may indicate that n-gram frequency plays a role in handwriting generalization, potentially due to how such combinations are formed in writing.
While language modeling has long been recognized as a key factor in text recognition systems, the impact of language-specific character patterns on handwriting formation has rarely been isolated as a variable to the best of the author's knowledge. Anecdotal and practical experience, such as variations in the handwritten form of the letter “t” when doubled (e.g., “tt”), shown in Figure 1.1, suggest that production frequency of character sequences may influence visual legibility and model learnability. To explore this hypothesis, both training corpora are analyzed for n-gram frequency distributions, and their representativeness for natural language is verified. During evaluation, recognition errors are categorized and correlated with n-gram categories (English-dominant, German-dominant, or language-neutral) based on their relative frequencies in the training corpora.

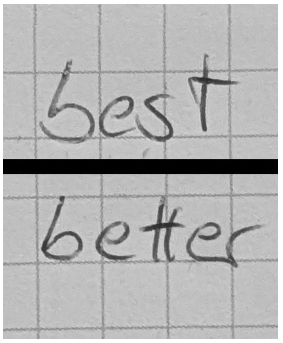

Figure 1.1 different “t” shapes in style of the author

This analysis aims to identify whether cross-lingual performance drops are disproportionately associated with unfamiliar or infrequent n-gram patterns from the model’s training language. The goal is to determine whether such n-gram-driven errors provide measurable evidence of a model's linguistic bias and to what extent this affects its robustness when faced with handwritten inputs in a different language.
Ultimately, this thesis contributes to understanding the linguistic robustness and transferability of HTR systems and provides evidence on whether language-specific training remains necessary in light of shared model architectures.

1.3 Research Problem

While deep learning-based Handwritten Text Recognition (HTR) systems have achieved remarkable performance on monolingual benchmarks, the majority of available datasets and pretrained models are based on English handwriting (Baek *et al.*, 2019; Diaz *et al.*, 2021; Fujitake, 2023). This monolingual bias limits our understanding of how language-specific features, particularly statistical patterns such as character n-grams, influence recognition performance when models are applied to different but related languages that share the same script.

Previous research has acknowledged the role of language modeling in improving HTR accuracy (Graves *et al.*, 2009; Diaz *et al.*, 2021), yet comparatively little is known about the implicit linguistic priors learned during training and how these may affect cross-lingual generalization. In particular, it remains unclear whether the frequency distribution of character combinations in the training corpus has a measurable effect on a model’s ability to recognize similar sequences in another language.

Despite its plausibility, this issue remains largely underexplored in HTR research. One reason may be the predominance of monolingual benchmarks and the scarcity of comparable multilingual handwriting datasets. Another challenge lies in the methodological complexity of isolating the effect of linguistic patterns like n-gram frequency from confounding variables such as handwriting style, image quality, or dataset composition. This

thesis does not aim to solve these problems comprehensively, but instead proposes a focused comparative study designed to highlight observable differences in recognition performance that may correlate with language-dominant n-gram structures.

The goal is not to generalize across all multilingual HTR scenarios, but to test a narrowly scoped hypothesis: Does the linguistic distribution of n-grams in the training data influence the model's ability to recognize those patterns when they appear in another language? If such effects can be empirically observed, even in a limited experimental setup, this could inform future efforts in data collection, multilingual model design, and cross-lingual transfer learning.

1.4 Research Question

This thesis investigates how the linguistic composition of training data affects the recognition performance of handwritten text recognition (HTR) models across languages. The work is situated at the intersection of multilingual machine learning and statistical language modeling, and it focuses on the following two **research questions**:

> **(RQ1)** Does the language of the training corpus affect the performance of a handwritten text recognition (HTR) model on crosslingual settings?
>
> **(RQ2)** How do language-dominant n-grams affect the performance and generalization of HTR models across German and English?

The first question takes a broader perspective and asks whether contemporary model architectures, such as ViT-CTC, exhibit inherent robustness in cross-lingual transfer settings, even in the absence of language-specific tuning or adaptation.

The second question examines whether differences in character n-gram frequency distributions between languages systematically impact recognition performance. This is assessed by training two structurally identical HTR models, one on English, one on German, and evaluating both on the other language's test set. Recognition errors are then analyzed in relation to the linguistic dominance of specific bigrams and trigrams.

Together, these questions aim to clarify whether language-specific features in the training data introduce measurable biases in HTR model performance, and whether current state-of-the-art architectures generalize reliably across closely related languages sharing the same script.

1.5 Structure of the Thesis

This thesis is organized into seven chapters. Following this introduction, **Chapter 2** presents the state of the art in handwritten text recognition, focusing on the development of relevant model architectures and the role of training data in sequence recognition tasks. The review includes recent transformer-based approaches as well as research addressing language-specific factors in HTR systems.

Chapter 3 outlines the methodology employed in this study. It describes the datasets used for training and evaluation, the model architecture adapted from Li et al. (2025), and the experimental setup, including preprocessing and training procedures. While a detailed n-gram analysis was originally planned, it is methodically documented but excluded from the main evaluation due to insufficient recognition performance in the cross-lingual scenario.

Chapter 4 presents the results of the training and evaluation experiments. The focus lies on standard character- and word-level error metrics, with special attention to differences in model performance when applied to test sets in the training vs. non-training language.

Chapter 5 discusses these findings in light of the second research question, namely the general cross-lingual robustness of ViT-CTC-based HTR models. It reflects on the implications of language mismatch in real-world scenarios and outlines possible limitations and confounding factors.

Chapter 6 summarizes the key contributions of the work, revisits the two research questions, and provides conclusions and directions for future research.

Chapter 7 (Appendix) documents the implementation and results of the extended n-gram error analysis. It includes the tagging and alignment scripts, categorized frequency tables, and two complementary evaluations: error distributions by n-gram category and per-n-gram error rates relative to their frequency. While not central to the main evaluation, these results offer additional insight into potential language-specific effects in cross-lingual HTR.

1.6 Relevance and Research Gap

Recent studies have begun to explore language adaptation effects in HTR, for instance through post-processing with language models or the use of domain-specific training data. Ströbel et al. (2022) demonstrate that language-specific language models improve HTR output quality in historical corpora, while Koch et al. (2023) show that tailored Latin HTR systems outperform general-purpose OCR engines, highlighting the relevance of linguistic alignment. However, these studies typically focus on post-hoc correction or lexicon-based evaluation and do not consider how language-specific character-level statistics might impact learning during training.
To date, no HTR study has systematically explored the role of character-level n-gram distributions and their language dominance in cross-lingual settings. While statistical n-gram models are well-established for tasks like spelling correction and speech recognition (Hecht, Riedler and Backfried, 2002), their integration into HTR on character level has remained superficial or confined to decoding stages. Moreover, n-gram frequencies are typically studied at the word level, not on the character level, which is the primary input domain for modern HTR models (Graves *et al.*, 2009; Baek *et al.*, 2019).

Analysis of state-of-the-art models has shown that even modern HTR models are difficult to generalize, but the more diverse the training datasets, the better HTR models recognize even "foreign" handwriting (as to see in 2. State of the Art). However, no papers were found investigating whether such recognition difficulties, especially between languages, can be specifically attributed to n-gram distributions.

This thesis addresses this gap by evaluating how models trained on different languages perform when applied to unseen handwriting in a related but distinct language. In doing so, it examines whether statistical underrepresentation of specific n-gram patterns in the training language affects recognition performance on the target language. While the n-gram-based hypothesis could not be fully validated due to performance limitations, the study provides empirical evidence on the general cross-lingual robustness of ViT-CTC architectures.

As AI-based handwriting recognition systems are increasingly deployed in multilingual educational, archival, and administrative contexts, understanding the limitations of cross-lingual generalization becomes practically relevant. This work contributes to the design of more robust HTR systems by clarifying whether and how language-specific training remains necessary, even in architectures that share visual structure across tasks.

2 State Of The Art

2.1 Historical and Modern Approaches to Handwritten Text Recognition

Early research in handwritten text recognition (HTR) dates back to the 1950s and 1960s, with pioneering work by researchers like Mermelstein and Eyden (1964) who explored automatic interpretation of handwritten words using rule-based methods and statistical heuristics (Caesar *et al.*, 1994; Kim and Govindaraju, 1997; Garrido-Munoz, Rios-Vila and Calvo-Zaragoza, 2025). Early systems predominantly relied on hand-crafted features and statistical models such as Hidden Markov Models (HMMs) and Gaussian Mixture Models (GMMs) (He, Chen and Kundu, 1992; Guillevic and Suen, 1997; Graves *et al.*, 2009).

As Alex Graves noted in his thesis, even after decades of research, building a reliable general-purpose HTR system for unconstrained handwriting remained an open challenge. HMM-based systems required extensive parameter tuning and struggled when faced with new writers, unusual glyph shapes, or varying writing instruments.

The introduction of deep learning marked a paradigm shift. Hidden Markov Model (HMM) based HTR systems could learn from labeled data and were capable of recognizing entire words or lines without requiring individual character segmentation, a significant improvement over purely rule-based methods (Garrido-Munoz, Rios-Vila and Calvo-Zaragoza, 2025).

In the 2000s Neural network architectures began to replace or augment the traditional HMM pipelines. Convolutional Neural Networks (CNNs) were introduced to automatically learn visual features from images of text lines, removing the need for manual feature engineering. CNNs and Recurrent Neural Networks (RNNs), particularly Bidirectional Long Short-Term Memory (BLSTM) networks combined with Connectionist Temporal Classification (CTC) enabled end-to-end training without explicit character segmentation (Graves and Schmidhuber, 2008; Graves *et al.*, 2009; LeCun, Bengio and Hinton, 2015; Ansari *et al.*, 2022).

The CTC loss function proved particularly suitable for unsegmented sequence data such as handwriting, offering a powerful alternative to traditional HMM-based systems. Ansari et al. (2022) note that CTC can be seen as a conceptual successor to HMMs, streamlining the training of neural networks for text recognition by directly optimizing for the most probable sequence alignment.

By the late 2010s, some systems began to incorporate attention mechanisms in encoder-decoder frameworks, enabling the model to focus on different parts of the image when generating each output character. For example,Bluche, Louradour and Messina (2017) and Kang *et al.* (2020) explored attention-based RNN decoders for HTR, while more recent models like TrOCR adopted a pure Transformer-based encoder-decoder design. Transformer networks have now attained top-tier performance in HTR by leveraging self-attention to capture long-range dependencies without recurrent connections. Notable examples include TrOCR (Li *et al.*, 2022) and related approaches, which use a Vision Transformer (ViT) as the image encoder and a pretrained Transformer decoder for language modeling. These models take advantage of large-scale pretraining (both on printed text and

natural language) to boost recognition performance. Another line of work integrates Transformers into the traditional pipeline: for instance, Hernandez *Diaz* et al. (2021) found that a CNN+Transformer encoder with a CTC decoder (plus a language model) was the most effective architecture for text line recognition across diverse domains. This hybrid approach essentially replaces the recurrent layers with a self-attention encoder while still using CTC for alignment, combining the strengths of both paradigms. Researchers have also proposed purely transformer-based encoder-only models with CTC (e.g. *HTR-ViT* by Li et al. (2025)) as well as "*Transformer in Convolutional Neural Networks (TransCNN)*" - Liu *et al.* (2021).

Nonetheless, critics such as Baek *et al.* (2019) and Diaz *et al.* (2021) point out that many benchmark results overlook important aspects such as modular interpretability, dataset bias, or the role of pretraining. They advocate for more transparent evaluation pipelines and challenge the assumption that high accuracy on standard benchmarks necessarily translates to general-purpose applicability.

2.2 The Role of Data in HTR Training

The composition of training data is a key factor in the accuracy and generalizability of HTR models. While many systems perform well on standard benchmarks, they often fail on out-of-distribution inputs due to narrow training domains. A recent multilingual study showed that linguistic mismatch, such as unseen languages or vocabularies, has a greater impact on recognition errors than visual handwriting variation. This underscores the importance of training on diverse textual domains and languages to ensure robust performance. Broad domain coverage across scripts, writers, and layouts is essential for HTR systems to generalize reliably (Garrido-Munoz and Calvo-Zaragoza, 2025).
In practical terms, an HTR engine trained on a single writer or a very uniform style may achieve low error on similar inputs but fail dramatically on different styles. Hodel *et al.* (2021) illustrate this effect: when they trained a model on a large number of pages all written by the *same few scribes*, its error rates on those specific hands were low, but performance dropped on even slight variations of handwriting. In other words, the model overfits the particular writers. They found that adding more variety, combining material from many scribes and sources, yielded a much more generalizable model.
Visual and layout diversity in training data is equally crucial for robust HTR performance. Models trained only on clean, segmented lines risk overfitting and often fail when faced with complex page structures such as overlapping text, marginalia, or multi-column layouts. Including varied document formats and writing conditions helps the model focus on content rather than superficial layout cues. Datasets like the Belfort register illustrate how exposure to such variation improves transcription accuracy on real-world documents. Broader layout coverage in training enhances resilience to structural and visual noise (AlKendi *et al.*, 2024).

In low-resource settings, synthetic data has become a valuable means of enriching HTR training corpora, especially for rare characters, n-grams, or underrepresented languages. Generative models like VATr or ScrabbleGAN allow targeted creation of handwriting samples that improve linguistic and visual coverage. These synthetic examples help the model learn character forms and sequences not sufficiently present in real data. Studies show that such data can significantly boost recognition accuracy, particularly when it closely resembles the

real domain. Modern generation techniques enable precise control over style and content, making synthetic data a powerful supplement to authentic handwriting collections (Pippi, Cascianelli and Cucchiara, 2023; de Sousa Neto *et al.*, 2024; Garrido-Munoz and Calvo-Zaragoza, 2025).

In this thesis, the two corpora used, IAM (English) and fhswf/german_handwriting (German), differ not only in language but also in vocabulary, average line length, and scanning modalities. The German dataset includes compound nouns and diacritics absent from the English set, making direct model comparisons non-trivial.

2.3 Multilingual and Crosslingual HTR and OCR

Optical Character Recognition (OCR) and Handwritten Text Recognition both aim to transcribe text from images, but they differ fundamentally in the nature of the text they handle. OCR typically deals with machine-printed or typeset text, where characters follow consistent font patterns and layouts. HTR, in contrast, must cope with the highly variable and personalized strokes of handwriting. This difference in visual complexity means that methods successful in one domain may not transfer directly to the other. For example, printed text recognition has become very accurate and is often considered a solved problem for clean prints. Many studies have examined the unique obstacles that arise when multiple languages or scripts must be handled in one system, making the topic well researched (Peng *et al.*, 2013; Sharma *et al.*, 2015; Mathew, Singh and Jawahar, 2016; Bušta, Patel and Matas, 2019; Namysl and Konya, 2019; Drobac and Lindén, 2020; Biró *et al.*, 2023).

Handwriting, however, remained far more difficult; only with the advent of deep learning (e.g. LSTM networks) did HTR reach levels of accuracy that made it broadly feasible (Hodel *et al.*, 2021).
As Hodel et al. note, even when an HTR model achieves “excellent” character error rates below 2.5% on one script, the irregularities of truly different handwriting styles can prevent such low error rates on other scripts or hands.

Multilingual handwritten text recognition presents unique challenges due to differences in alphabets, ligatures, character frequencies, and orthographic conventions between languages. Many existing HTR systems are implicitly designed for monolingual settings or specialized alphabets, using language models to achieve better results, making cross-lingual generalization difficult (Diaz *et al.*, 2021; Koch *et al.*, 2023).
Transformer-based architectures such as TrOCR support multilingual decoding by incorporating language models pre-trained on diverse corpora (Li et al., 2022). However, most vision-based HTR pipelines, including Vision Transformer with CTC decoders, are typically trained and evaluated monolingually. This limits our understanding of their ability to generalize to unseen languages, especially those with low resource availability or distinct morphological patterns.
True cross-script HTR (e.g., one model handling Latin, Cyrillic, and Chinese together) is an ongoing research challenge, often requiring massive models or explicit multilingual training strategies (Keysers *et al.*, 2017).

Several contemporary studies have explicitly tested cross-script and cross-language generalization in HTR systems. Dash *et al.* (2024) (in preparation) focus on CRNN-based HTR and report on how well such models trained on one script/language can recognize another. Their findings indicate a significant drop in performance on unseen scripts, for instance, a model trained on Latin script handwriting has difficulty with a different script like Cyrillic or Arabic, often misidentifying characters or failing entirely. Even for unseen languages in the same script, accuracy declines if the language has unique character frequencies or combinations not seen during training. This aligns with intuitive expectations: an English-trained CRNN might partially decipher Spanish handwriting (shared alphabet aside from ñ), but would fare poorly on Greek or Tamil. Dash et al. emphasize the importance of cross-script training or adaptation: incorporating multiple scripts in the training data, or using transfer learning to fine-tune the model on a new script, dramatically improves results compared to zero-shot transfer. They also note that certain architectural choices can aid generalization, such as using bidirectional LSTMs and CNN features that capture low-level strokes which might be common across scripts. Still, their experiments reinforce that without explicit multi-script exposure, an HTR model's knowledge does not generalize far beyond its training distribution (unlike some OCR scenarios where basic font shapes overlap between languages). This underlines a key design principle: to build a truly multilingual HTR, one should either train on a multilingual dataset or employ domain adaptation techniques when moving to a new language.

To address the challenge of adapting HTR models to new languages efficiently, Chang & Li (2024) propose transformer-based HTR models with efficient multilingual fine-tuning strategies. Transformers have revolutionized many NLP and vision tasks by enabling powerful sequence modeling, but a full transformer model has a huge number of parameters, which makes training one model on many languages data-hungry. Chang & Li explore approaches like pretraining a transformer on a large pool of handwriting data (possibly in a high-resource language) and then fine-tuning it on target languages with limited data. Chang & Li report that such methods retain the performance on the original language while gaining proficiency in the new one, effectively yielding a single model that can handle multiple languages by conditioning on language-specific parameters. An advantage of transformer models in this context is their global attention mechanism, which can learn relationships between characters and context that are beneficial in any language (e.g., the model might learn a generic concept of "word spacing" or stroke order that transfers across alphabets). However, they also find that one must account for language-specific quirks, for instance, Arabic has joining characters and requires context for shape disambiguation, while Latin languages may have more stable isolated characters but larger vocabularies. Efficient fine-tuning can address this by allowing a small degree of language-specific customization without losing the shared representational power of the core model.

Across these studies, several common themes and promising strategies emerge for multilingual and crosslingual HTR:

- **Script Identification and Language Tagging:** As noted, determining the script of a text image (either via a separate classifier (Sharma *et al.*, 2015) or implicitly within a multi-task model) is often essential. A multilingual HTR system might first run a script

identifier on each line or word. Alternatively, a single model can be trained with a one-hot language-ID input so that it “knows” which language’s character set to output, effectively conditioning the model on the script. This prevents confusion between similar-looking characters in different alphabets and lets the model limit its predictions to the appropriate set. Script identification is particularly crucial when dealing with mixed-script documents (e.g., a bilingual archive where notes in English and French are intermingled): it might be wise to route each line to the corresponding language model to avoid misclassification. Sharma et al.’s work clearly showed the benefit of a dedicated script-ID stage in improving overall accuracy on multilingual video text

- **Shared Feature Learning:** Despite differences, many scripts share low-level features: strokes, curves, junctions. A CNN or transformer backbone can learn a rich representation of strokes that is *reusable* across languages. Keysers et al. leveraged this by using one backbone for 22 scripts, essentially learning a universal handwriting feature space (Keysers *et al.*, 2017). For offline HTR, researchers increasingly attempt a single encoder (CNN or ViT) to extract features, followed by multiple decoders or output layers for different scripts. This way, most of the model’s capacity is devoted to generic shape recognition, while only the final classification adapts to specific alphabets. Such shared architecture approaches make adding a new language cheaper, one only needs to train a new output head or small extension for the new script, as Chang & Li (2024) implement with fine-tuning strategies. The challenge is ensuring the shared features are truly general; sometimes pretraining on a diverse dataset (including synthetic data for low-resource scripts) is needed to achieve this.

- **Language-Specific Variation (Morphology and Orthography):** Languages differ in more than just characters. Morphologically rich languages can produce very long words (agglutinative languages) or have compound words that strain a character-sequence model that was only trained on shorter words. For instance, a German handwritten word can be exceptionally long due to compounding, which might be challenging if a model was mostly trained on short English words. Crosslingual HTR models have to cope with these differences in word length and structure. A possible solution is incorporating a character-level language model or at least allowing the model to output longer sequences than it saw in training. Orthographic differences, such as the use of accent marks or special characters, also need attention. If a multilingual model uses a unified character set (say Unicode), it should include all necessary characters during training; otherwise, it might simply omit or confuse an unseen accent. One observed limitation in some crosslingual OCR is that error rates rise on languages with characters absent from training (Crosilla, Klic and Colavizza, 2025). In HTR, this could mean misrecognizing ö as o, or ñ as n, if the model wasn’t trained on Spanish or German data. To mitigate this, one can include at least some data for each language to teach the model those distinctions, or use multilingual OCR results as pseudo-labels to guide HTR for low-resource languages (though caution is needed, since OCR errors could mislead HTR).

In conclusion, the research to date makes it clear that designing multilingual or crosslingual HTR systems requires careful consideration of both vision and language factors. One must not assume that success on a multilingual OCR benchmark (where printed fonts might be easier to distinguish) guarantees success on handwritten text. OCR benchmarks may overestimate generalization because printed characters are more consistent; HTR models face greater variability and thus need more robust training regimes. The key findings from Sharma et al. (2015) (for multi-script pipelines), Namysl & Konya (2019) (for synthetic training across languages), Dash et al. (2024) (for the limits of cross-script CRNN generalization), Chang & Li (2024) (for transformer fine-tuning across languages), Hodel et al. (2021) (for general vs. specific model performance), and Keysers et al. (2017) (for scaling to dozens of scripts with shared architecture collectively inform a roadmap. That roadmap includes using script-aware components, sharing learned features across languages, pretraining on abundant or generated data, and injecting language-specific knowledge when needed.

2.4 Evaluation Metrics and Diagnostic Tools

Handwritten text recognition systems are typically evaluated using error rates that quantify the discrepancy between the predicted transcription and the ground truth. The two most common metrics are the Character Error Rate (CER) and Word Error Rate (WER). These are defined as the normalized Levenshtein edit distance between the recognized text and the reference text, i.e. the minimum number of single-character insertions, deletions, or substitutions required to transform one into the other. CER is computed at the character level, while WER is computed at the word level (Neto, Bezerra and Toselli, 2020). In practice, a CER of X% means that X percent of characters are wrong (on average), and analogously for WER with whole words. Lower CER/WER values indicate more accurate recognition (AlKendi et al., 2024). Namysl and Konya (2019) explicitly describe using the Levenshtein distance to measure CER in their OCR experiments, and earlier works (e.g. Sharma et al. (2015)) likewise employ edit-distance-based evaluation methods in HTR pipelines.

CER and WER serve complementary roles in evaluating HTR output. CER, being a fine-grained character-level metric, is useful for detailed analysis of recognition errors. It captures individual character misrecognitions and is language-agnostic, making it especially insightful for examining which letters or symbols are confused by the model. WER, operating at the word level, provides a higher-level view of overall readability, but is often more sensitive to compounding errors and segmentation issues in handwritten text. In HTR, a single incorrect character will cause an entire word to be counted as wrong under WER (Sánchez *et al.*, 2019). Consequently, WER can sometimes paint a pessimistic picture, for example, a word with one letter misread is 0% correct by WER even if most of its characters were recognized correctly. Moreover, handwritten scripts may lack clear inter-word spaces or have variable spacing, leading to segmentation ambiguities. If the system merges or splits words improperly (for instance, inserting or missing a space), this directly impacts WER even when character recognition is otherwise correct.

Beyond these quantitative metrics, HTR and OCR evaluation is often complemented by qualitative diagnostic tools to pinpoint error patterns and guide system improvements (e.g. confusion matrix) (Drobac and Lindén, 2020).
Error analysis can also be broken down by n-gram category on word or character level or context, for instance, grouping errors by linguistic context such as common prefixes, suffixes, or specific three-letter sequences. This kind of n-gram analysis reveals whether certain character or word combinations (e.g. “th”, “sw”) or particular word fragments are often misread, which might indicate shortcomings in the model’s context modeling or in the language model (if used) (Evershed and Fitch, 2014).
Nguyen *et al.* (2019) already introduced character n-gram analysis for post OCR error detection. However, this was related to OCR not HTR.
In this work, the evaluation includes not only CER and WER but also an optional n-gram-based error categorization strategy. This method relates recognition errors to the linguistic patterns found in the training corpora and can reveal systematic weaknesses in cross-lingual generalization.

2.5 Technical Implementation

The implementation follows recent trends in HTR research. All models are trained using PyTorch and adapted from the official HTR-VT repository Li et al. (2025). Model weights and tokenizer configurations are shared via GitHub for reproducibility and remote access.
Image preprocessing and augmentation follow standard practices using scikit-image and torchvision. Training experiments were conducted on GPU desktops with RTX 4090 cards and logged using TensorBoard. YAML configuration files and GitHub version control ensured experiment traceability throughout development. The trained models used for the evaluation are available on the Hugging Face Hub.

2.6 Summary and Positioning

The field of HTR has progressed from statistical to deep learning-based and now to Transformer-based approaches. Transformer models with CTC decoding show strong recognition performance in monolingual settings, yet their behavior in cross-language tasks remains underexplored.
This thesis investigates the extent to which the language of the training data affects model performance on a fixed test set. By holding the model architecture constant and varying only the training language (English vs. German), it seeks to isolate linguistic influences in HTR performance. The following chapters describe the data preprocessing, model configuration, and evaluation protocol used in this controlled experiment.

3 Methods and Material

3.1 Dataset

Training Sets

Two monolingual datasets were selected for training the two HTR models:

- **English**: The IAM Handwriting Dataset (Marti and Bunke, 2002) is a widely used benchmark dataset for HTR, consisting of approximately 10,000 lines of handwritten text. The texts are based on the Lancaster-IBM Corpus and are all in English. Annotations include line, word, and character levels. The data was extracted locally and saved in a proprietary format (.jsonl) with image paths and transcriptions.

- **German**: The freely available dataset fhswf/german_handwriting was used for the German model. This contains approximately 11,000 lines of text from various samples of modern German handwriting.. Both datasets were manually preprocessed and partitioned to enable fair and comparable experiments.

English Dataset – IAM Handwriting Database

The IAM Handwriting Database is a widely adopted benchmark for line-level handwritten text recognition in English. It contains scanned forms of handwritten English text along with corresponding transcriptions. The subset used in this study is based on the `lines.txt` file, which provides a clean mapping between each text line and its corresponding ground truth. This dataset has become a de facto standard in the field and has been employed in numerous studies, with over 1000 citations.

The dataset was randomly split into training (80%), evaluation (10%), and test (10%) sets. Each sample was converted into a format compatible with the training architecture: individual image files with matching `.txt` files containing the transcription. The IAM dataset was chosen not only because of its accessibility and compatibility, but also because it reflects the statistical structure of natural English (see 3.1.2). Alternative English datasets, such as CENSUS-HWR (Joshi *et al.*, 2023), BRUSH (Kotani, Tellex and Tompkin, 2020) and IMGUR5K ('IMGUR5K-Handwriting-Dataset', 2025), were excluded due to limited linguistic diversity, unnatural calligraphic nature, or cultural bias.

German Dataset – fhswf/german_handwriting

The German training data is based on the fhswf/german_handwriting dataset, which was collected at the University of Applied Sciences Südwestfalen. It consists of 10,000 samples of handwritten text produced by 15 individuals and is freely available via Hugging Face. The dataset was manually downloaded, extracted, and converted into the same structure as the IAM dataset: images and corresponding `.txt` files. The same 80/10/10 split was applied.

German datasets suitable for HTR training are scarce. Many open-access collections focus on historical scripts such as Kurrent, Sütterlin, or Fraktur (e.g., READ (Toselli *et al.*, 2018),

Konzil, Schwerin,etc (*Papers with Code - Machine Learning Datasets*)), which are not representative of modern German. The fhswf dataset was selected because it contains contemporary handwriting samples and avoids overly domain-specific content.

Due to limited space in the repository, the datasets can't be uploaded within the project and must be downloaded and prepared separately. The splits are documented in the respective folders under train.ln, test.ln and val.ln.

Preprocessing and Data Integrity

No advanced preprocessing was performed. All data was used as provided, except for structural formatting to match the model requirements. The main goal was to retain the natural distribution of character sequences to enable later analysis of linguistic patterns such as n-gram frequencies.

3.1.1 Language Dominant N-Gram Classification

To investigate whether specific character sequences influence cross-lingual HTR performance, a structured analysis of bi- and trigram distributions was conducted. The goal was to identify n-grams that appear disproportionately often, in either German or English and assess whether such language-dominant patterns correlate with systematic recognition errors in cross language evaluation.

N-Gram Extraction

Bigrams and trigrams were extracted from the line-level transcriptions of both the IAM dataset (English) and fhswf/german_handwriting (German) training datasets.
The following steps were performed to prepare and analyze the datasets:

- extract the entire text from the datasets into an own `.txt` to make it easily accessible for the analysis.
- tokenize each line into overlapping character bigrams and trigrams, then compute absolute and relative frequencies for each n-gram within its respective corpus.

A minimum occurrence threshold is currently not enforced, but future iterations may incorporate one to filter out low-frequency artifacts and improve classification robustness.

Normalization and Frequency Ratio

To enable frequency comparisons despite differences in dataset size, all n-gram frequencies were normalized by dividing the absolute frequency $f_{g,L}$ of each n-gram g by total number of n-grams N_L in that corpus, getting the relative frequency $relF_{g,L}$:

$$relF_{g,L} = \frac{f_{g,L}}{N_L} \tag{3.1}$$

Here, $L \in \{EN, DE\}$ denotes the language corpus. To classify language dominance, the ratio between the German and English relative frequencies was calculated for each n-gram:

$$r_g = \frac{relF_{g,DE}}{relF_{g,EN} + \varepsilon} \tag{3.2}$$

where $\varepsilon = 10^{-5}$ is a small constant added to avoid division by zero. This measure quantifies how much more prevalent an n-gram is in one language relative to the other, independent of corpus size.

Classification Thresholds

The classification was based on a symmetric heuristic using the threshold $r_g = 3.0$, following logarithmic reasoning. The interpretation is as follows:

- German-dominant: $r_g > 3.0$
- English-dominant: $r_g < \frac{1}{3} \approx 0.33$
- Language-neutral: $0.33 \leq r_g \leq 3.0$

This threshold ensures that only n-grams with substantial frequency disparities are labeled as language-dominant. The value of 3.0 reflects the idea that an n-gram must appear at least three times more often in one language to be considered dominant. Several scientific sources from linguistics and statistics explicitly use or discuss a factor-3 threshold to determine dominant elements, because it marks a clear distance that goes beyond random fluctuations and thus indicates a significant dominance both practically and statistically (Martinger, 2013; Moss, 2015).

Output Format and CSV Export

The final result of the analysis was stored in two separate CSV files:

- `preparation_validation/output/ngram_analysis_2gram.csv`
- `preparation_validation/output/ngram_analysis_3gram.csv`

which include:

- the raw count of each n-gram,
- its relative frequency in both corpora,
- the calculated frequency ratio r_g
- the assigned dominance label.

These files form the basis for further analysis in the evaluation chapter, where error patterns of both HTR models are correlated with the dominance class of misrecognized n-grams.

Visualization

To further illustrate the distribution of n-grams within the established language-dominance categories, two bar charts were created, one for bigrams and one for trigrams (see Figures 3.1 and 3.2). Each chart presents the top ten n-gram (divided in bi- and trigrams). Each n-gram is coloured due to its category (english dominant, german dominant and neutral).

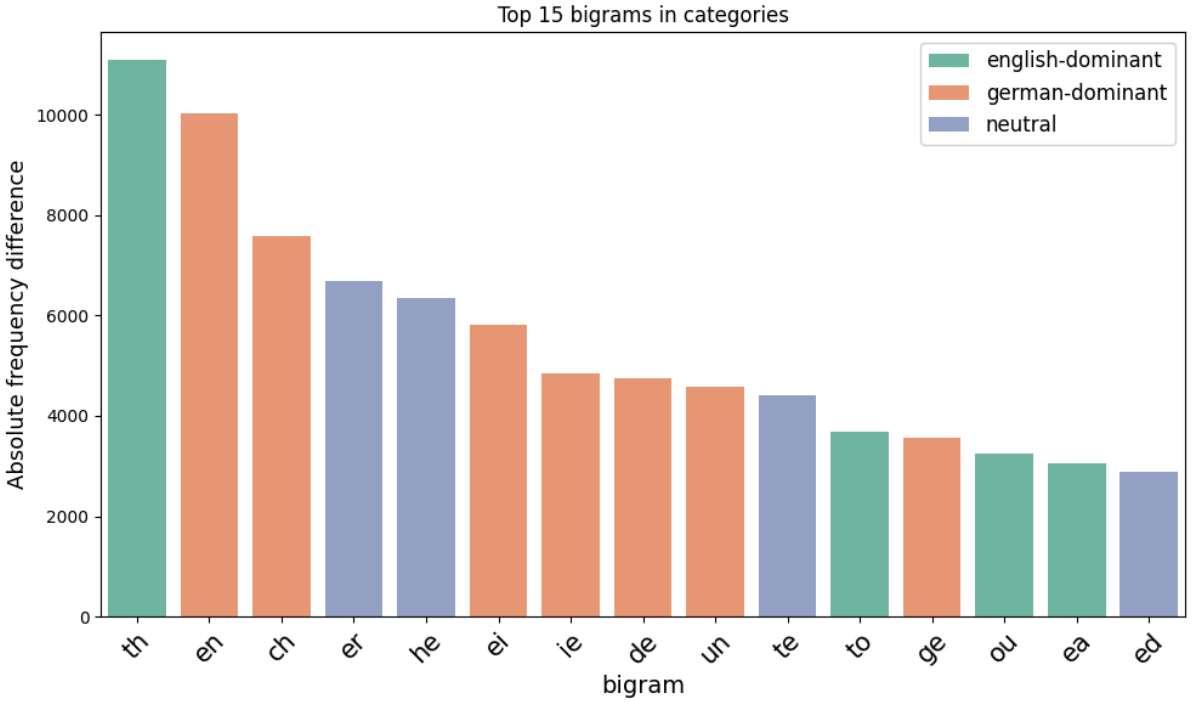

Figure 3.1: Top 15 Bigrams, divided in english dominant, german dominant and neutral

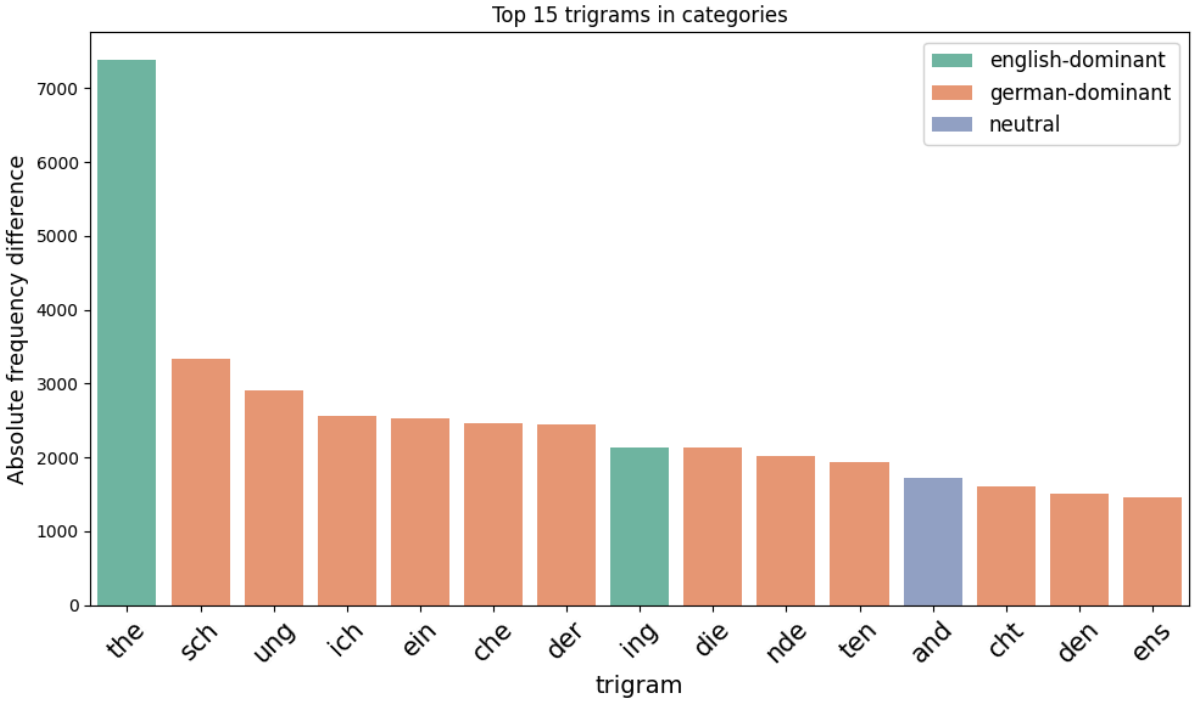

Figure 3.2: Top 15 Trigrams, divided in english dominant, german dominant and neutral

The diagrams reveal distinct distribution patterns between the categories. This visualization supports the assumption that the classification procedure effectively captures language-specific patterns and justifies its use as the basis for subsequent error analysis.

By aggregating relative frequencies across the entire corpus and not focusing solely on the most common n-grams, the data provide a more holistic perspective on the internal structure of each category. This view is essential for validating that the classification is not only driven by extreme outliers, such as umlauts or rare character combinations, but reflects broader statistical trends in both corpora.
For further data find the corresponding `.csv` files in the linked GitHub repository.

3.1.2 Validation of Language Representativeness

To ensure that the results of this study are not based on language-specific outliers or highly specialized corpora, both training datasets, English (IAM) and German (fhswf/german_handwriting), were analyzed for their representativeness of natural language. The central aim was to verify that their character sequences reflect common linguistic patterns found in general English and German usage.
For this purpose, the 100 most frequent bi- and trigrams were extracted from both corpora and compared against statistical n-gram distributions published by the Leipzig Corpora Collection (*Leipzig Corpora Collection - Wortschatz Deutsch*, Website) via Practical Cryptography (*Practical Cryptography*, Website). These reference frequencies are based on approximately 4.5 billion characters of natural text collected from web and media sources, ensuring a broad and balanced linguistic base.
Each n-gram was assigned a rank based on its frequency in the training data and then matched with the corresponding reference rank.

While exact positional rankings occasionally differed by one or two places, the overall distributions in high-frequency n-grams aligned remarkably well between the datasets and their respective reference corpora. Common high-frequency trigrams such as "the", "and", or "ing" in English and "der", "die", or "sch" in German appeared in both the expected frequency range and order. Except for the trigram “ung” that is significantly less common in the German reference corpora and “hat” that is significantly less common in the English reference corpora, the difference doesn't exceed a distance of 5 ranks for the top 10 n-grams. This qualitative similarity suggests that the datasets capture typical orthographic patterns of their respective languages.
A representative excerpt of the comparisons can be found in Tables 3.1, 3.2, 3.3 and 3.4.

bigrams German			
n-gram	own rank	reference rank	difference
en	1	2	-1
er	2	1	1
ch	3	3	0
te	4	6	-2
ei	5	5	0
de	6	4	2
nd	7	8	-1
in	8	7	1
ie	9	9	0
un	10	15	-5

Table 3.1: Top 10 German-dominant bigrams in the training data compared to their frequency rank in the Leipzig Corpora Collection.

trigrams German			
n-gram	own rank	reference rank	difference
sch	1	3	-2
ein	2	2	0
ung	3	18	-15
ich	4	4	0
der	5	1	4
che	6	7	-1
nde	7	5	2
die	8	6	2
ten	9	9	0
und	10	10	0

Table 3.2: Top 10 German-dominant trigrams in the training data compared to their frequency rank in the Leipzig Corpora Collection.

bigrams English			
n-gram	own rank	reference rank	difference
th	1	1	0
he	2	2	0
in	3	3	0
er	4	4	0
an	5	5	0
re	6	6	0
es	7	7	0
en	8	11	-3
st	9	9	0
ed	10	13	-3

Table 3.3: Top 10 English-dominant bigrams in the training data compared to their frequency rank in the Leipzig Corpora Collection.

trigrams English			
n-gram	own rank	reference rank	difference
the	1	1	0
ing	2	3	-1
and	3	2	1
ent	4	4	0
her	5	6	-1
tha	6	8	-2
ere	7	11	-4
ion	8	5	3
nth	9	9	0
hat	10	17	-7

Table 3.4: Top 10 English-dominant trigrams in the training data compared to their frequency rank in the Leipzig Corpora Collection.

This validation step was particularly important given the potential for distortion in small or domain-specific corpora. For instance, one portion of the German dataset contains handwritten university notes, which could have skewed the distribution toward technical terms. Without this comparative analysis, such imbalances might have gone unnoticed. The external reference allowed a reliable evaluation without the need for manual text inspection, which would have introduced subjectivity and been infeasible at scale.
Based on the observed correspondence between corpus-internal and external n-gram distributions, both datasets are considered sufficiently representative of modern English and German handwriting. This ensures that the results of the following experiments can be

interpreted beyond the narrow scope of the selected datasets and are generalizable to the broader written language.
All the described code that was used for the preparations and validations for the dataset and the its out `.txt` and `.csv` files can be found as followed:

```
preparation and validation/
├── Extract_text_from_datasets.py      # extract entire text from the datasets
├── ngram_frequency_categorization.py  # compute frequencies for each n-gram and
├── seperate_top_ngrams.py             # extracts top 100 n-gram of each dataset
├── top_ngram_ranking.py               # compare the top n-gram for validation
├── output/
│   ├── german_handwriting_transcriptions.txt  #extracted from german dataset
│   ├── ngram_analysis_2gram.csv     #extracted brigrams sorted and categorized
│   ├── ranking_bigramm_diff_de.csv  #ranking of most common bigrams across the
german training corpora and the reference corpora
│   ├── top100_bigrams_de.txt        #top 100 bigrams in the german training set
│   └── ...
├── practicalCryptographyData          # top n-gram lists of Leipzig Corpora
Collection
```

3.2 Model Architecture

The model architecture used in this thesis follows the ViT-CTC approach proposed by Li et al. (2025), which combines a CNN-based feature extractor with a transformer encoder and Connectionist Temporal Classification (CTC) loss for sequence alignment. The Pipeline is visualized in Figure 3.3.

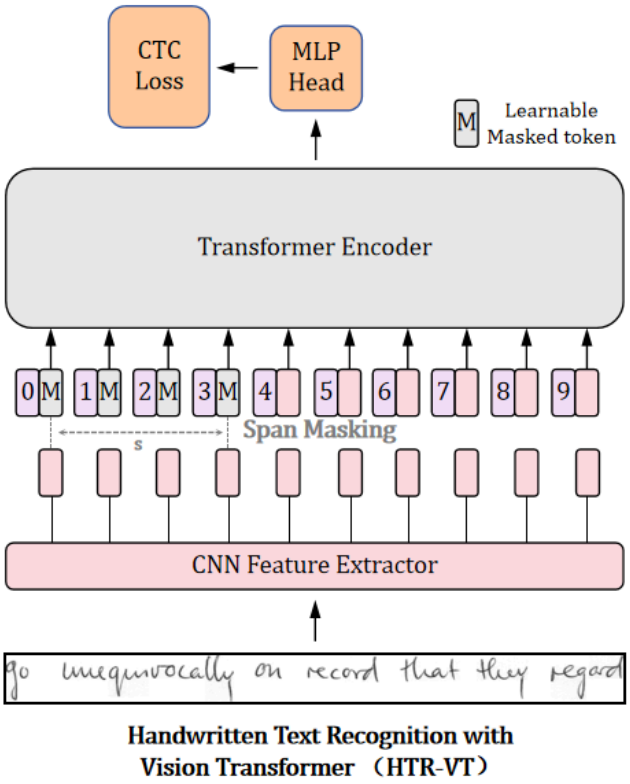

Figure 3.3: ViT-CNC Pipeline (Source: Li et al 2025)

Input images are first processed by an CNN extractor, which converts handwritten text lines into spatially structured feature sequences with a fixed length, representing non-overlapping image patches. These feature maps are then flattened and passed to a Vision Transformer (ViT) encoder, which takes the features as input tokens and gives character predictions as output.

To improve robustness and generalization in the context of handwritten input, the architecture incorporates several enhancements:

- a span-masking strategy during training, where random sequences of visual tokens are replaced by learnable mask tokens to improve contextual learning;
- sinusoidal positional encodings to preserve the order of visual tokens; and
- Sharpness-Aware Minimization (SAM) as an optimizer, encouraging the model to converge towards flatter minima for better generalization.

For decoding, the architecture employs CTC loss, a method widely used in sequence-to-sequence tasks without explicit alignment between input and output. The CTC decoder enables the model to output sequences of variable lengths, without requiring character-level segmentation during training. This makes it well-suited for line-level handwriting recognition where character boundaries are often ambiguous or overlapping (Li et al., 2025).

3.3 Training Protocol

To ensure a valid comparison between the two language-specific models, all training procedures were carried out under controlled and reproducible conditions. The training was based on the publicly available HTR-VT architecture by Li et al. (2025) (Yuting, 2025), which employs a ViT-CTC model optimized for handwritten text line recognition. The training pipeline was adapted to handle both the IAM dataset (English) and the fhswf/german_handwriting dataset (German), while keeping the model architecture and core procedures identical.

Training Environment

All model training was conducted on dedicated GPU workstations located at the university, each equipped with an NVIDIA GeForce RTX 4090 and CUDA 11.8 support. Preliminary preprocessing and validation of the codebase were performed locally on a similar system to ensure consistency across environments.

The full project was managed via a unified GitHub repository (`HTR-VT_Bachelor`) and the training process was extended with automated model uploading to the Hugging Face Hub for persistence and later evaluation. TensorBoard was used to monitor training and evaluation metrics, logging data are also available in the respective Hugging face repositories.

Training Configuration

The original implementation (train.py) provided by the authors of the HTR-VT model was adopted and extended to support both German and English training data. Specific adaptations include:

- **Preprocessing**: The English dataset (IAM) as well as the German dataset were reformatted into a `.ln` structure (`train.ln`, `val.ln`, `test.ln`), consistent with the input format expected by the training pipeline.
- **Environment Setup**: A custom requirements.txt file was created to ensure compatibility with Windows environments, streamlining setup across multiple machines (local PC and university GPU-desktops).
- **Model Configuration**: The core training script train.py and the options.py module were extended to dynamically load and support the German dataset. In addition, a Hugging Face integration was added to automatically upload model checkpoints after each evaluation iteration.

- **Training Parameters**: Following the original paper's training configuration, the hyperparameters were adopted without modification for the English model. Both models work with AdamW Optimizer. For the German model, minor changes were required due to image size differences:
    - --img-size was set to 1024×64 (vs. 512×64) and
    - --patch-size to 4×16 instead of the default to ensure compatibility with the resolution of the handwritten text lines.

Both models used the same training duration (98k iterations) and regular evaluation checkpoints every 500 steps.

The complete set of training arguments used for both models is summarized in Table 3.5 below:

Parameter	English Model 1 & 2	German Model
--max-lr	1e-3	1e-3
--train-bs / val-bs	64 / 8	64 / 8
--img-size	512×64	1024×64
--patch-size	4×16	4×16
--weight-decay	0.5	0.5
--total-iter	98000	98000
--mask-ratio	0.4	0.4
--attn-mask-ratio	0.1	0.1
--max-span-length	8	8
--proj	8	8
--dila-ero-max-kernel	2	2
--dila-ero-iter	1	1
--proba	0.5	0.5
--alpha	1	1

Table 3.5: Training arguments for English and German model

The models were trained from scratch, without relying on pre-trained weights, to ensure language-specific adaptation from the outset. Given the reported training efficiency in the original work, where convergence was achieved within 16 hours,the full training cycle could be reproduced within reasonable computational constraints. The english was trained for approximately 16h and the german model was trained for ~30h.
Note: Due to an upload failure the german model interrupted training at 98.000 iterations, while 100.000 iterations were planned. Since the model did not achieve better CER and WER results for a big amount of pre completed iterations and only limited time were left for the model training, the german model wasn't retrained and the english model, due to its faster training results, got retrained with an adjusted total-iter value of 98000.
The loss function and optimization strategy were adopted without modification.

Checkpointing and Interruption Recovery

The training script was extended to upload all saved checkpoints directly to the Hugging Face Hub. This ensured that training could be resumed at any point without local storage limitations and that the models are always available across multiple PCs. Model states, optimizer parameters, and iteration counters were included in the checkpoints, allowing for full recovery in case of interruption or crash.

Metric Logging and Evaluation

Loss curves and performance metrics were logged throughout training using TensorBoard. The primary metric for validation was Character Error Rate (CER), measured using greedy decoding. The Word Error Rate (WER) was computed post-hoc using saved predictions for both validation and test splits.

This tightly controlled training setup provides a reliable basis for analyzing how the linguistic properties of training data influence the model's performance when all other conditions are kept constant.

3.4 Evaluation Strategy

The evaluation strategy in this thesis was initially designed to investigate not only the overall recognition performance of each model but also to analyze how the statistical properties of the training corpora affect recognition errors. To this end, both quantitative metrics and a planned n-gram-sensitive error categorization were prepared.

Quantitative Metrics

The primary evaluation criteria are the Character Error Rate and Word Error Rate, which are commonly used in handwritten text recognition to measure edit distances between predicted sequences and the ground truth. These metrics are computed using the Levenshtein distance at the character and word level, respectively, and allow for a standardized comparison across different models and datasets.
Evaluation is performed once after each training run on the full held-out test set. During training, performance is periodically assessed on a validation set using the same metrics to identify the best-performing checkpoint based on CER.

Optional N-Gram Error Categorization

A secondary analysis was devised to study the linguistic origins of errors in more detail. The idea was to compare model performance on bigrams and trigrams that are statistically dominant in one language (e.g., "th", "ing" in English vs. "sch", "ung" in German) against more neutral patterns. This required a multi-step process:

1. **Frequency Analysis** of training corpora to rank all occurring bigrams and trigrams.
2. **Relative Frequency Classification** into German-dominant ($r > 3$), English-dominant ($r < 0.33$), and language-neutral categories.
3. **Prediction Alignment** using Levenshtein distance to identify error regions.

4. **Tagging** of erroneous n-grams based on their language classification.
5. **Normalization** of error counts by total occurrence of each n-gram category.

Although the computational setup and codebase for this analysis were fully prepared, the method was not included in the core evaluation. Preliminary testing revealed that recognition quality on the cross-lingual test sets was too poor to derive meaningful error category distributions. Especially for the models evaluated on the language they were not trained on, both CER and WER reached levels where individual character errors could not be reliably attributed to linguistic mismatch.
However, the full pipeline and source code for the planned n-gram categorization are provided in the project repository under `evaluation/ngram_analysis` and will be discussed further in the Appendix. This offers a foundation for future work using models with better cross-lingual generalization or alternative datasets that allow for a more balanced comparative analysis.

4 Results

This chapter presents the empirical results of both models, one trained on English handwriting data, the other on German, evaluated on both test sets. The results are structured into quantitative performance, qualitative examples.

The complete Data of the evaluation is to be found in the linked GitHub repository in `evaluation/` . To determine which file represents which evaluation, refer to Table 4.1.

trained on	tested on	File
English	English	predictions_vs_groundtruth_**EoE**.cs
English	German	predictions_vs_groundtruth_**EoG**.csv
German	German	predictions_vs_groundtruth_**GoG**.cs
German	English	predictions_vs_groundtruth_**GoE**.csv

Table 4.1: Principle of naming of files for different evaluation combinations

4.1 Quantitative Evaluation

To assess the transcription performance of the trained HTR models, the models were quantitatively evaluated using the Character Error Rate (CER) and Word Error Rate (WER) as the primary metrics. Both are standard in handwritten text recognition, with CER offering fine-grained insights into character-level mismatches, while WER accounts for higher-level tokenization errors, which are particularly relevant in practical downstream applications such as text search or indexing.

Four evaluations were conducted in total:

- The model trained on the IAM dataset (English) was evaluated on both the IAM and the German test set.
- The model trained on the German dataset was evaluated on both the German and the IAM test set.

The results for each model-dataset were computed using the official test.py script provided by the original implementation. This script applies the same preprocessing, decoding strategy, and loss function (CTC) as during training, ensuring a fair and consistent evaluation. To make the comparison more fair, the examples of the german testset that included script exclusive characters like "äöüßÄÖÜ” got filtered when using the testset to evaluate the English model. All models were tested on both their native test set and the respective foreign-language set. The results are summarized in Table 4.2.

Evaluation Metrics

Let **CER** be defined as the normalized Levenshtein distance between predicted and ground truth character sequences. **WER** is computed analogously at the word level, taking into account insertions, deletions, and substitutions.
The Table 4.2 shows the CER, WER and Loss of the both models, tested across both test sets.

Modell	CER	WER	Loss
English → English Testset	0.0221	0.0818	5.811
English → German Testset	0.4294	0.8685	84.393
German → German Testset	0.0146	0.0671	4.069
German → English Testset	0.1995	0.5318	46.255

Table 4.2: Evaluation results (CER/WER/Loss) across models and test sets.

CER and WER across different training–test combinations

The chart seen in Figure 4.1 visualizes the Character Error Rate (CER) and Word Error Rate (WER) for four evaluation scenarios, comparing models trained on English and German. While all models perform well on their native test set, a substantial drop in recognition quality is observed when tested on data written in a different language.

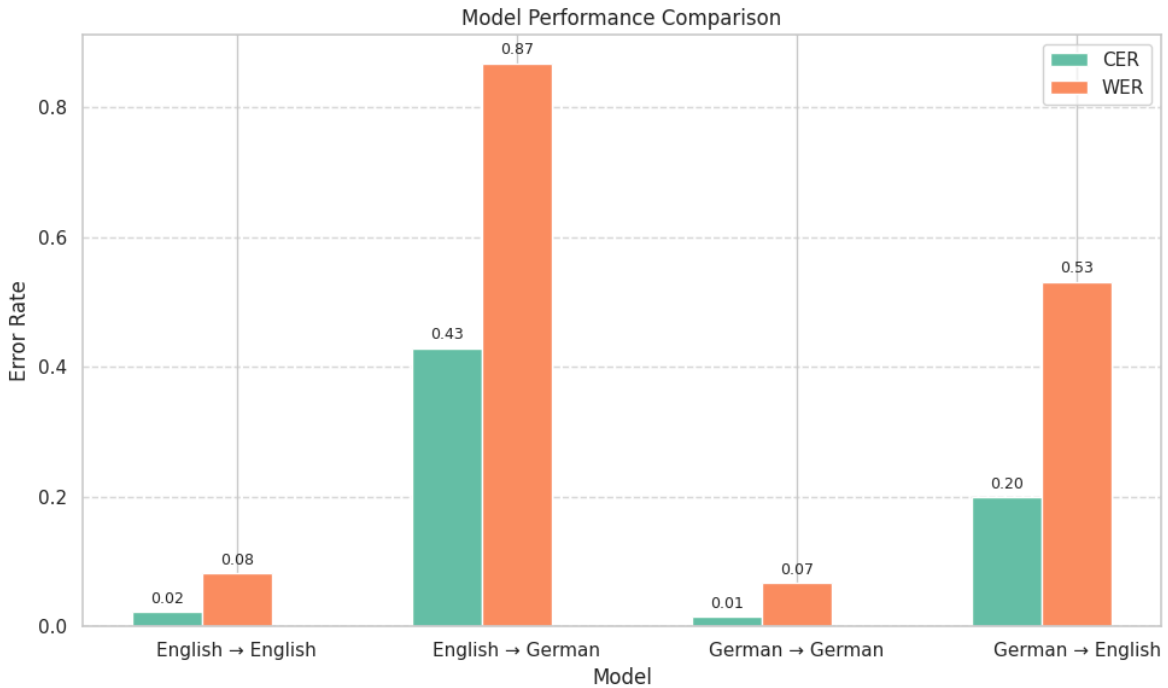

Figure 4.1: Visualisation of Evaluation results (CER/WER/Loss)

4.2 Error Analysis and Cross-Language Observations

The evaluation results reveal a substantial performance gap when each model is tested on a dataset written in a language it was not trained on. This effect is especially pronounced in the English-trained model evaluated on the German test set, where both Character Error Rate (CER = 0.4294) and Word Error Rate (WER = 0.8685) deteriorate significantly compared to in-language evaluation (CER = 0.0221, WER = 0.0818). A similar, though less extreme, drop is observed for the German-trained model when tested on English data (CER = 0.1995, WER = 0.5318).

These results suggest a consistent trend across all settings: models exhibit significantly higher error rates when evaluated on out-of-distribution language data. The implications and potential reasons for this behavior will be further discussed in Chapter 5.

To further support these findings, qualitative error examples are provided in Section 4.3.

4.3 Qualitative Error Examples

To complement the quantitative evaluation, a selection of prediction, reference pairs were inspected manually to gain insight into the nature of model errors.

The following tables, Table 4.3, Table 4.4, Table 4.5 and Table 4.6, are showing three representative examples of the two models evaluated on the different test sets.

English dataset, English test set

Type of Error	Prediction	Ground Truth
Minor	*an honours degrle course . Farming is Britain's*	*an honours degree course . Farming is Britain's*
Moderate	*on those bfeen miles of noustain roads , it disappeard from the earthly*	*on those fifteen miles of mountain roads , it disappeared from the earthly*
Major	*of VAETY ANO it's a artwule . " Niker was the Hospital's*	*of variety and it 's a change . " Nigel was the hospital's*

Table 4.3: Different Types of Errors of the English Model evaluated on the English testset

Minor error:
The model produces *“degrle”* instead of *“degree”*, indicating a minor substitution that likely stems from similar character embeddings. The impact is negligible for understanding but highlights the precision limit in fine-grained prediction.

Moderate error:
In *“bfeen miles of noustain roads”*, several phonetic approximations (*“bfeen”*, *“noustain”*) replace valid tokens. While the sentence retains structure and context, it risks miscommunication and demonstrates sensitivity to character-level noise.

Major error:
The phrase *“VAETY ANO it’s a artwule”* is an example of major degradation. It includes non-words, capitalization errors, and distorted named entities (*“Niker”* instead of *“Nigel”*). These errors suggest a local decoding failure or unstable attention.

English dataset, German test set

Type of Error	Prediction	Ground Truth
Minor	*Nationalbeweging*	*Nationalbewegung*
Moderate	*-specifzert einen Ausschitt and dem Verhatten einer Klasse*	*- spezifiziert einen Ausschnitt aus dem Verhalten einer Klasse*
Major	*Hthikent ds Bualtle Asgidt*	*deutlich auf den Punkt. Die Adressaten sind*

Table 4.4: Different Types of Errors of the English Model evaluated on the German testset

Minor error:
In the word *“Nationalbeweging”*, the model replaces the final *„u“* with an *„e“*. While this minimally affects readability, it still leads to a non-standard form in German, reducing accuracy in OCR-critical contexts.

Moderate error:
This example shows a string of mid-level inaccuracies, such as *„specifzert“* instead of

„spezifiziert“ or *„Verhatten“* instead of *„Verhalten“*. These distortions indicate partial familiarity with German morphology but insufficient generalization.

Major error:
“Hthikent ds Bualtle Asgidt” represents a complete breakdown in character alignment and linguistic plausibility. The sequence contains nonsensical words and suggests the model struggles to decode non-English scripts when unfamiliar with key token patterns.

German dataset, German test set

Type of Error	Prediction	Ground Truth
Minor	*…kommt. Bis, jetzt wurden*	*…kommt. Bis jetzt wurden*
Moderate	*Persistierung (wo): Fik oder Datenbank oder...*	*Persistierung (wo) : File oder Datenbank oder...*
Major	*Das Rutro Ränst auf des PAnutobatum.*	*Das Auto fährt auf der Autobahn.*

Table 4.5: Different Types of Errors of the German Model evaluated on the German testset

Minor error:
In this sentence, an unnecessary comma appears after the word *“Bis,”* which introduces a minor punctuation error. Such small mistakes may still impact syntactic parsing or readability in automated pipelines.

Moderate error:
This sample contains several middle-tier distortions: *“Fik”* instead of *“File”*, misplaced colons, and unnecessary ellipses. These introduce semantic ambiguities while still retaining some recognizable structure.

Major error:
“Das Rutro Ränst auf des PAnutobatum.” shows a severe degeneration of the original input. The output contains pseudo-words with phonetic resemblance but no syntactic or semantic validity. It highlights the model’s occasional instability even on known-language data.

German dataset, English test set

Type of Error	Prediction	Ground Truth
Minor	*government bi the Duke of Wellington.*	*government by the Duke of Wellington .*
Moderate	*blown up. He has wow Vevealed his full plans*	*blown up . He has now revealed his full plans*
Major	*an homower degne Gowese. Forwirg ir Blitain´s*	*an honours degree course . Farming is Britain's*

Table 4.6: Different Types of Errors of the German Model evaluated on the English testset

Minor error:
The model substitutes *"bi"* for *"by"* and omits the final period. These are negligible errors with little semantic impact, yet they reflect insufficient robustness in minor details.

Moderate error:
The words *"wow"* and *"Vevealed"* are corruptions of *"now"* and *"revealed"*. While still partially intelligible, such deviations may lead to misunderstandings in downstream processing.

Major error:
The prediction *"an homower degne Gowese…"* reflects a breakdown of phonetic approximations without semantic coherence. The original phrase refers to an educational qualification, but the model fails to convey even the general meaning.

4.4 Observations

Across all evaluations, a clear pattern emerges: models perform substantially better when tested on the same language they were trained on. The native-language scenarios (English→English and German→German) yield the lowest Character Error Rates (CER) and Word Error Rates (WER), both quantitatively and qualitatively. Notably, the German model achieves the lowest CER overall (0.0146), suggesting that the dataset and training pipeline were well-aligned.

In contrast, the performance of both models drops significantly when applied to cross-language test sets. The English-trained model fails to generalize to the German test data (CER: 0.4294), and the German model exhibits similarly high CER when tested on English data (0.1995). This finding highlights the importance of language-specific training in HTR, particularly due to differences in character sets, word structure, and stylistic conventions.

The qualitative analysis further supports this conclusion. While minor substitution errors were common in all models, cross-language predictions often exhibited breakdowns at the semantic or structural level, including phonetically distorted words and failed recognition of idiomatic phrases. Such errors are not adequately captured by CER or WER alone, reinforcing the importance of complementary qualitative evaluation methods.

In summary, the results confirm the strong language dependence of handwritten text recognition. They provide a quantitative and qualitative foundation for the subsequent discussion, where the implications of these findings for multilingual and cross-lingual HTR are explored in more detail.

5 Discussion

This chapter interprets the results presented in Chapter 4 in light of the central research question:

Does the language of the training corpus affect the performance of a handwritten text recognition (HTR) model on crosslingual settings?

5.1 Evaluation of Results

The results presented in Chapter 4 demonstrate substantial differences in performance between the evaluated HTR models, depending on the language of both the training and test sets. The most informative metrics for this assessment are the Character Error Rate (CER) and Word Error Rate (WER), which together offer a comprehensive picture of the models' recognition accuracy.

As expected, each model performs best on the test set that matches the language of its training data. The German-trained model achieves a CER of 1.46% and a WER of 6.71 on the German test set, while the English-trained model reaches a CER of 2.21% and a WER of 8.18% on the IAM (English) test set. These results confirm the general reliability of both models in monolingual scenarios and align with prior findings that HTR models can achieve low error rates when trained and evaluated within the same domain and language context.

However, a substantial drop in recognition accuracy was observed when both models were evaluated on out-of-domain, cross-lingual data. When applied to the German test set, the English-trained model showed a markedly higher WER and CER, and the same pattern occurred in reverse. These findings illustrate that the models failed to generalize to handwriting in a language they had not seen during training, even though both languages share the Latin script.

The difference between CER and WER is also notable across all experiments. While CER reflects low-level character mismatches, WER exposes higher-order structural issues such as incorrect word boundaries or substitutions that affect the semantic integrity of full words. In cross-lingual settings, WER values exceed 0.5 in both cases, indicating severe distortion at the word level and reinforcing the importance of language alignment in training data.

Interestingly, the magnitude of the performance drop was not symmetric: the English-trained model performed worse on German than the German-trained model did on English. While this result must be interpreted cautiously due to possible dataset differences, it may hint at asymmetries in script complexity, training diversity, or character set coverage. English contains more frequent ligatures and high-frequency bigrams such as “th”, which are rare in German, whereas the German script includes umlauts and more compound words. These structural differences may introduce unseen or underrepresented n-grams during cross-lingual inference, contributing to elevated error rates.

These results indicate that CTC-based models exhibit strong language dependence, likely due to their reliance on sequential token distributions and absence of explicit linguistic priors. A more detailed interpretation of these findings is presented in Section 5.4.
Overall, the results underline the limits of language-agnostic HTR assumptions. Although both models operate on visually similar scripts, their recognition ability appears strongly tied to the distributional properties of their training language. This reinforces the relevance of training language as a decisive variable in HTR performance.

5.2 Implications for HTR Training

The findings of this study offer several key insights for the training of handwritten text recognition (HTR) models, particularly in multilingual or cross-lingual deployment scenarios. The sharp performance degradation observed in cross-lingual evaluations highlights the language-specific nature of learned representations in modern HTR architectures. Although both models operated on visually similar scripts and shared the same architecture, they failed to generalize reliably across language boundaries. This suggests that exposure to language-specific distributions, such as frequent bigrams, diacritics, and morphological structures, can be crucial for achieving robust recognition performance.
These results indicate that training language-specific models remains the most reliable strategy for high-stakes HTR applications, especially where low error tolerance is required (e.g. archival digitization or legal document analysis). The relatively poor performance of otherwise well-trained models on unseen languages further supports the case for multilingual pretraining or domain-adaptive fine-tuning as necessary steps in cross-lingual settings.
For practitioners, this underscores a central limitation of current systems: script overlap alone does not guarantee transferability. An English-trained model cannot be expected to perform adequately on non-English handwriting without prior adaptation, regardless of shared alphabets. Representative examples of language-specific constructions must be present in the training corpus for the model to learn robust associations.
In multilingual or low-resource contexts, this emphasizes the need for either dedicated language-specific models or training strategies that incorporate diverse linguistic input. Possible approaches include multilingual pretraining, synthetic data augmentation with rare language patterns, or adaptive fine-tuning using small in-domain samples. While the present study does not generalize to all scripts or recognition settings, it demonstrates that language is not a neutral variable in HTR. Robust cross-lingual performance requires deliberate design choices in training data composition and model adaptation.
These observations stress the importance of dataset availability and diversity. Language-specific HTR systems can only reach competitive accuracy if appropriate annotated datasets are available. This underlines the need for more open-access, high-quality handwriting datasets in languages beyond English.

5.3 Limitations and Alternative Explanations

While the results offer clear trends in language-dependent model behavior, several confounding variables and methodological constraints may have influenced the observed

error patterns. This section outlines potential sources of bias in data, architecture, and evaluation.
Many of these issues are well-documented in prior research and were known at the time of experimental design. Some of these potential mitigations, including the integration of external language models or multi-task learning, were deliberately excluded in this study to preserve the interpretability of the planned n-gram-based error analysis. Since language models introduce strong linguistic priors during decoding, their use would likely have masked or distorted the relationship between training data and character-level error patterns. The following sections outline limitations relating to dataset design, architectural biases, and evaluation methodology, while also indicating where future work might address these issues without compromising linguistic transparency.

5.3.1 Dataset- and Label-Based Constraints

Several data-related inconsistencies may have contributed to the observed model performance patterns. First, the input images from the two datasets differ in format and preprocessing: while the German dataset uses RGB images, the IAM dataset is provided in grayscale. This variation in color channels may affect the low-level features extracted during training, despite normalization and resizing.
Second, the training and evaluation image resolutions (img_size) were not uniform across experiments. The German dataset was processed at a higher width (1024×64) compared to the standard size (512×64) used for IAM, which may lead to differences in the effective receptive field and resolution of character features.
Third, layout conventions such as spacing, punctuation, and line length differ between datasets. The German dataset exhibits more compact text lines and a larger variety of sentence-level structures, which may impact the sequence modeling capabilities of CTC-based architectures.
Lastly, it should be noted that the IAM dataset contains overlapping content across its train and test sets, that is, the same sentence may appear in both splits but written by different writers. This could lead to an overestimation of generalization ability when evaluated on the native test set.

5.3.2 Model Architecture Bias

The model architecture used in this thesis relies on Connectionist Temporal Classification decoding, which introduces certain inductive biases. CTC assumes monotonic alignment between input and output sequences, limiting its ability to handle reordering or strong contextual dependencies. Moreover, the decoder operates independently of language semantics, making it less sensitive to syntax or word-level plausibility. This raises the question whether decoding mechanisms with integrated language models may be more appropriate for cross-lingual HTR.

No external language model was integrated into the decoding pipeline. As a result, the model's output is purely driven by visual patterns and lacks the corrective influence of statistical or pretrained linguistic information. This may be especially disadvantageous in

cross-lingual scenarios, where out-of-domain data often requires linguistic generalization beyond the training context.

Furthermore, no architectural adaptation was introduced to handle language-specific characteristics such as character clusters, diacritics, or compound structures, factors that may be better modeled through hybrid or multi-head decoding approaches.

5.3.3 Evaluation Framework Limitations

The evaluation relied primarily on Character Error Rate and Word Error Rate, which are standard in HTR but offer limited insight into linguistic structure or semantic correctness. These metrics are string-level measures and do not distinguish between different error types, such as phonetic confusion, affix misalignment, or lexical disfluency.
While an n-gram analysis framework was developed and is available in the Appendix, its practical use was limited due to the poor cross-lingual performance of the models. As the error rates exceeded reasonable thresholds, a systematic analysis of error patterns (e.g., cross-lingual token shifts) would have been statistically unstable and is therefore not included in the main results.
The qualitative selection of error cases, while insightful, was not based on a fully randomized or representative sampling. It primarily illustrates specific types of failure but does not claim statistical significance.

5.3.4 Hypotheses Regarding Error Sources

Some error patterns observed in cross-lingual evaluations may stem from language-specific phenomena. German contains a higher proportion of compound nouns and inflectional morphology which were not present in the English training data. This could lead to segmentation errors or symbol mismatches during decoding.

The differences in token frequency distributions and character entropy between the two languages may also bias the decoder's output, particularly when using CTC without LM regularization.

Finally, both models were trained on limited handwriting samples from a small number of writers, which limits the model’s ability to learn style-invariant features. This lack of style diversity may further reduce the model’s robustness when applied to visually or linguistically dissimilar test data.

6 Conclusion and Outlook

6.1 Summary of Findings

This thesis investigated the influence of training language on the performance of handwritten text recognition models, with a focus on cross-lingual generalization. Two structurally identical ViT-CTC models were trained on English (IAM) and German (fhswf) datasets and evaluated on both monolingual and cross-lingual test sets.

The results show that both models perform well when evaluated on test data in their training language, achieving low character and word error rates (e.g., CER 0.0146 / WER 0.0671 for the German model). However, cross-lingual performance drops drastically, with the English model reaching a CER of 0.4294 on German data, and the German model reaching 0.1995 on English text. These findings confirm that even with shared scripts, training language plays a major role in recognition performance.

Qualitative analyses support this conclusion: cross-lingual errors often involve phonetic approximations or structurally implausible outputs, especially in morphologically rich or unfamiliar constructions. This suggests that modern HTR systems implicitly learn language-specific regularities, both visual and statistical, which are not easily transferred across linguistic domains.

Ultimately, the findings underscore that language is not a neutral variable in HTR. Robust recognition in multilingual settings requires either diversified training data, adaptive strategies, or explicit linguistic modeling.

6.2 Answer to the Research Question

The central research question of this thesis was whether the language of the training corpus affects the performance of an HTR model on monolingual test data. Based on the experimental results, the answer is clearly affirmative: the language used during training has a substantial impact on recognition accuracy.
Both models performed well when evaluated in-language, but failed to generalize reliably in cross-lingual settings. This performance drop cannot be explained by visual dissimilarity alone, as both languages share the Latin script and similar handwriting styles. Instead, the results suggest that linguistic factors, such as character distributions, diacritics, and morphological complexity, are implicitly learned during training and strongly influence model behavior.
Although a more granular linguistic error analysis was planned, the available results already indicate that language-specific features shape the model's internal representation of handwritten text. Effective cross-lingual HTR thus requires deliberate consideration of language during model design and training.

6.3 Contribution to the Field

This thesis contributes to the field of handwritten text recognition by empirically demonstrating that the language of the training data significantly affects model performance, even in state-to-the-art architectures like Vision Transformer with CTC. While language models and linguistic priors are well established in natural language processing, their implicit role in visual recognition pipelines remains underexplored.

By systematically isolating the training language as a variable and holding all architectural and procedural factors constant, this study reveals that modern visual HTR models develop language-specific biases that impair cross-lingual generalization. These findings support and extend previous claims in the literature that corpus-level alignment and linguistic structures have a hidden but measurable influence on recognition behavior, even in end-to-end models without explicit language components.

In addition, the work addresses a methodological blind spot in HTR research: evaluation strategies often assume that performance on monolingual test sets is indicative of general capability. This study demonstrates that such assumptions may lead to overestimating a model's robustness in multilingual or low-resource deployments. As such, the results inform not only model design but also evaluation protocol design in multilingual settings.
Finally, this work provides practical resources to the community by making the full experimental pipeline and trained models openly available on Hugging Face. These assets can serve as a basis for further cross-lingual benchmarking and reproducibility efforts.

6.4 Outlook and Future Work

The present study opens several avenues for further research that extend beyond its current scope. While the dual-model comparison revealed clear performance drops in cross-lingual settings, the mechanisms underlying these failures remain only partially understood. Future work should investigate the following directions in more depth:

Multilingual Pretraining and Fine-tuning

A logical extension would be to test whether multilingual pretraining, either on synthetic or real handwriting corpora, can mitigate the observed performance degradation. Architectures like TrOCR (Li et al., 2022), which include language-aware decoders, could be evaluated in direct comparison with decoder-free models to examine whether cross-lingual transfer improves with language modeling capacity.

Integration of External Language Models

The current ViT-CTC models operate without any linguistic prior. Integrating external language models at the decoding stage, such as n-gram or transformer-based models,could help disambiguate noisy visual inputs, particularly in morphologically rich or diacritic-heavy languages like German. This follows suggestions by España-Boquera *et al.* (2011), who showed performance gains through hybrid decoding approaches.

Systematic Evaluation on Aligned Multilingual Datasets

One limitation of this work was the difference in visual and lexical properties between the English and German corpora. Future experiments should be run on fully parallel or artificially aligned multilingual datasets to control for such variance. Alternatively, unsupervised methods could be used to generate pseudo-parallel corpora for evaluation.

Error Typology and Targeted Data Augmentation

This work showed that specific types of linguistic and visual mismatch lead to predictable failure patterns. These insights could be used to guide targeted data augmentation strategies or curriculum learning approaches. For example, models might benefit from contrastive training on confusing n-grams or similar handwriting styles across languages.

Benchmarking Cross-lingual HTR Transfer

Finally, the field would benefit from a standardized benchmark for cross-lingual HTR evaluation, similar to what exists for OCR or NLP (e.g., XTREME for multilingual NLP (Hu *et al.*, 2020)). This would make comparisons more reliable and highlight progress in architectures that generalize across languages.

Several directions emerge from this research:

- **Extension to other languages**: Including structurally different scripts (e.g., French, Turkish, or Slavic languages) could validate the generality of these findings.
- **Integration of synthetic data**: Controlled injection of language-dominant n-grams might further isolate their influence.
- **Multilingual training and fine-tuning**: Exploring mixed-language corpora and transfer learning approaches could help improve model robustness.
- **Decoder-level interventions**: Future research may explore integrating external n-gram models or fine-grained attention control to mitigate cross-lingual mismatch.

6.5 Methodological Supplement

While the core evaluation in this thesis focused on cross-lingual model performance as measured by CER and WER, the initial research design also included a more granular N-gram-based error analysis. The intention was to categorize recognition errors according to the language-dominance of character sequences (e.g., German-typical trigrams such as "sch", "ung", or "äu"). Due to unexpectedly poor model performance on cross-lingual test sets, this approach was not included in the main evaluation chapter, as the noisy outputs would have undermined the interpretability of fine-grained error categories.

Nevertheless, the technical implementation of the N-gram pipeline, as well as the methodology for future replications, are documented in the Appendix. These resources are

provided to support further research on linguistic transfer effects in handwriting recognition systems, especially in multilingual or low-resource contexts.

7 Appendix

7.1 n-gram Analysis

This section provides a technical overview of the implementation pipeline for a character-level n-gram error analysis. This section provides a technical overview of the implementation used for the character-level n-gram error analysis. The methodological rationale for this analysis, including the motivation, assumptions, and theoretical categorization of n-gram dominance, was already outlined in Chapters 3.4 and 4 of the main thesis. For completeness and to ensure reproducibility, the key steps are briefly summarized here before presenting the technical pipeline for the evaluation in greater detail. Although the method was not included in the final evaluation due to the poor performance of the models on mismatched language data, it remains a valuable exploratory tool and is fully implemented in code.
The full implementation of the described pipeline includes scripts for frequency extraction, ratio-based categorization, error alignment, tagging logic, and summary generation and is available in the project repository under evaluation/ngram_analysis/ :

```
evaluation/ngram_analysis/
├── align_predictions.py   # Matches predictions to ground truth via
Levenshtein
├── tag_ngram_errors.py    # Labels affected n-grams with language categories
├── summarize_ngram_errors.py   # Aggregates error statistics by n-gram type
├── errorsrate_undominant-ngrams.py   #calculate relative error rate of each
ngram
├── output/
│   ├── aligned_predictions_EoG.csv
│   ├── tagged_bigram_errors_EoG.csv
│   ├── summary_bigram_errors_EoG.csv
│   ├── errorrate_bigram_EOG.csv
│   ├── errorrate_summary_bigram_EOG.csv
│   └── ...
```

7.1.1 N-Gram Frequency Extraction

Each training corpus (English and German) was analyzed to determine the most frequent character-level bigrams and trigrams.

To achieve this:

- The full set of transcriptions from each training dataset was tokenized into overlapping character bigrams and trigrams.

- Frequencies were calculated using collections.Counter, and the results were stored in sorted tables.

- The top 100 bigrams and trigrams were saved for further use in categorization and visualization.

7.1.2 Categorization by Language Dominance

N-grams were assigned to one of three categories based on their relative frequencies across the two training corpora, see Table 7.1.

Category	Condition	Interpretation
German-dominant	$\frac{f_{DE}}{f_{EN}} > 3.0$	Common in German, rare in English
English-dominant	$\frac{f_{DE}}{f_{EN}} < \frac{1}{3}$	Common in English, rare in German
Language-neutral	$\frac{1}{3} \leq \frac{f_{DE}}{f_{EN}} \leq 3.0$	Shared across both languages

Table 7.1: Classification of the n-grams based on a threshold ratio of 3.0

This classification relies on a threshold ratio of 3.0, following conventions in linguistic frequency analysis, to ensure that only substantially overrepresented n-grams are labeled as "dominant".

7.1.3 Alignment of Predictions and Ground Truth

To identify and localize recognition errors, each line of predicted text was aligned with its corresponding ground truth using character-level edit distance. This alignment was used to identify exactly which parts of a line differed, through insertions, deletions, or substitutions, and where these differences occurred. This step was implemented in the script align_predictions.py, which leverages Python's built-in difflib module to compute minimal edit operations between two sequences.

The alignment script processes a CSV file containing predictions and references. For each entry:

- The predicted string and the ground truth are read as character sequences.

- Differences between them are computed using a sequence-matching function.

- The result is stored as a list of operations, each defined by:

- the type of edit (insert, delete, or replace)
- the start and end indices in both the prediction and the reference string

The aligned data is written to a new CSV file with the structure shown in Table 7.2.

prediction	ground_truth	diff_ops
"governmnent bi tle Duke"	"government by the Duke"	"[('replace', 12, 13, 12, 13), ('replace', 15, 16, 15, 16)]"

Table 7.2: Example of Aligned prediction, reference, and character-level edit operations

This alignment information is crucial for subsequent tagging of character-level n-grams. It ensures that only those segments directly affected by recognition errors are extracted and analyzed in downstream steps. By localizing errors precisely within each prediction, the pipeline avoids overgeneralization and enables fine-grained categorization by n-gram context.

7.1.4 Error Tagging by N-Gram Category

In the next step, all aligned character-level errors were analyzed with respect to their linguistic context. The goal was to identify whether the errors occurred within character sequences (bigrams or trigrams) that are statistically dominant in one training language.

For each aligned error segment:

- Overlapping bigrams and trigrams were extracted from the corresponding region of the ground truth string.
- These n-grams were compared against precompiled frequency tables from the training corpora.
- Each n-gram was assigned to one of three categories:
  - **German-dominant**: occurs at least 3× more often in the German corpus than in the English corpus
  - **English-dominant**: occurs at least 3× more often in the English corpus
  - **Language-neutral**: occurs in both corpora with comparable frequency

The tagged n-grams were saved in a new CSV file, with the structure shown in Table 7.3. Each entry contains:

- the original prediction
- the ground truth
- the computed edit operations
- and a list of all error-associated n-grams, annotated by category

Example output format:

prediction	ground_truth	edit_ops	error_ngrams
"govermnent bi tle Duke"	"government by the Duke"	[('replace', ...), ...]	[(('th', 'english-dominant'), 1), (('by', 'language-neutral'), 1), …]

Table 7.3: Example of Tagged errors with prediction, reference, edits, and n-gram category labels.

By tagging each error with its local n-gram context, this step enables subsequent analysis of whether language-dominant character sequences are more prone to recognition errors, especially in cross-lingual settings. This contextualized view allows error rates to be interpreted in light of the linguistic distribution in the training data.

7.1.5 Category-Level Error Summary

The final step in the n-gram error analysis pipeline summarizes how frequently different types of language-dominant n-grams were involved in recognition errors. This is achieved by processing the outputs from the previous tagging step and computing category-level statistics across the dataset.

For each prediction sample, the script counts:

- how many error-associated n-grams belong to each category (German-dominant, English-dominant, or language-neutral), and
- how many test samples contain at least one error in that category.

Based on this, three metrics are reported per category:

- **Total Errors** – the absolute number of affected n-grams in each category

- **Affected Samples** – the number of test lines in which errors from this category occurred

- **Relative Frequency** – the percentage share of this category relative to all error-tagged n-grams

The results will be shown in the next Chapter.

This overview helps assess whether certain types of linguistic patterns are overrepresented among recognition errors, particularly in cross-lingual scenarios. Although not used for the main evaluation due to unstable decoding outputs on the test sets, the full implementation is available and can support more robust follow-up studies in future work.

7.1.6 Error Rate Aggregation by Category

To gain a more fine-grained perspective on which character sequences were most error-prone, a separate analysis was conducted that compared the number of recognition errors per n-gram to its overall frequency in the test set. This allows identification of sequences that, despite being rare, were disproportionately affected by misrecognition. The implementation processes the output from the previous tagging step and the full ground truth text line by line. For each line:

- The ground truth text is tokenized into overlapping n-grams (bigrams or trigrams),

- All n-grams found in error-marked regions are counted,

- The total number of occurrences for each n-gram is also recorded across the full test set,

- Using a precomputed frequency map, each n-gram is assigned to one of three categories (english-dominant, german-dominant, neutral),

- The relative error rate is calculated by dividing the number of misrecognized occurrences by the total number of times the n-gram appeared.

The result is a per-n-gram error table showing, for each sequence, how often it was seen, how often it caused an error, its resulting error rate, and its category. This analysis enables identification of specific patterns such as “th” or “sch” that may disproportionately contribute to model failures in cross-lingual settings.

A second function aggregates these values by category to calculate both the mean error rate and the median error rate for each group. While the mean gives an average tendency, the median highlights the typical error behavior and is more robust to outliers. This distinction helps identify whether high error rates are broadly distributed across a category or concentrated in a few outlier sequences.

Code Availability and Execution
The complete pipeline can be executed locally by running the following scripts in order:

1. `align_predictions.py` - aligns predicted vs. ground truth lines and stores edit metadata.

2. `tag_ngram_errors.py` - extracts all error-relevant n-grams and maps them to categories.

3. `summarize_ngram_errors.py` - produces aggregate statistics for comparative analysis.

4. `errorsrate_undominant-ngrams.py` - calculate relative error rate of each n-gram and calculate mean and median error rate

Each script is documented inline and expects CSV files as intermediate input/output. Paths and filenames can be adjusted via variables at the end of each script.

7.2 Results

7.2.1 Error Distribution by N-Gram Category

The following tables present the actual output of the n-gram error analysis, conducted on the cross-lingual predictions of both HTR models. Each table summarizes how often character-level errors occurred in the context of language-dominant or language-neutral bigrams and trigrams. The values represent:

- **total_errors**: number of erroneous n-grams of a given category,
- **affected_samples**: number of test lines in which such errors occurred,
- **relative_frequency (%)**: proportion of errors in this category relative to all errors.

English-trained model evaluated on German handwriting
The following tables, Table 7.4 and Table 7.5, show how frequently each n-gram category was involved in recognition errors when the English-trained model was applied to German handwriting.

Bigram error distribution			
n-gram category	total errors	affected samples	relative frequency (%)
english-dominant	165	126	02.40
german-dominant	1983	516	28.84
neutral	4727	578	68.76

Table 7.4: Error distribution by bigram category, English model on German handwriting

Trigram error distribution			
n-gram category	total errors	affected samples	relative frequency (%)
english-dominant	281	165	03.59
german-dominant	4093	544	52.35
neutral	3445	551	44.06

Table 7.5: Error distribution by trigram category, English model on German handwriting

German-trained model evaluated on English handwriting
The following tables, Table 7.6 and Table 7.7, show the error distribution by n-gram category when the German-trained model was applied to English handwriting.

Bigram error distribution			
n-gram category	total errors	affected samples	relative frequency (%)
english-dominant	2177	889	27.25
german-dominant	370	313	04.63
neutral	5443	1031	68.12

Table 7.6: Error distribution by bigram category, German model on English handwriting

Trigram error distribution			
n-gram category	total errors	affected samples	relative frequency (%)
english-dominant	4746	1029	54.43
german-dominant	427	293	04.90
neutral	3547	905	40.68

Table 7.7: Error distribution by trigram category, German model on English handwriting

The values shown are derived from the output of the `summarize_ngram_errors.py` script. All frequency tables and intermediate results are available in the project repository for further inspection.

7.2.2 N-Gram-Specific Error Rate

To complement the frequency-based error analysis, this section presents aggregated error rates by n-gram category. Based on the full set of n-grams in the test set, each sequence was classified as english-dominant, german-dominant, or language-neutral using a precomputed category map. The previously calculated per-n-gram error rates were then summarized across categories.
For each group, the mean error rate indicates the average recognition error probability for n-grams of that category. The median error rate serves as a robust indicator of the typical

error tendency, especially when individual outliers are present. The number of unique n-grams per category is included to reflect the sample size behind each metric.
This analysis provides a broader impression of whether certain linguistic groups were generally more error-prone, even beyond individual high-risk patterns.

English-trained model evaluated on German handwriting
Table 7.8 and Table 7.9 showing the mean error rate and the median error rate for the bigram and trigram analyses for the English-trained model evaluated on German handwriting.

Bigram error distribution			
n-gram category	mean error rate	median error rate	number of n-grams
english-dominant	0.4399	0.4907	54
german-dominant	0.5462	0.5385	73
neutral	0.4665	0.4553	250

Table 7.8: Aggregated bigram error rates by category, English-trained model evaluated on German handwriting

Trigram error distribution			
n-gram category	mean error rate	median error rate	number of n-grams
english-dominant	0.4976	0.5000	266
german-dominant	0.6417	0.6667	572
neutral	0.5775	0.5714	1017

Table 7.9: Aggregated trigram error rates by category, English-trained model evaluated on German handwriting

German-trained model evaluated on English handwriting

Table 7.10 and Table 7.11 showing the mean error rate and the median error rate for the bigram and trigram analyses for the German-trained model evaluated on English handwriting.

Bigram error distribution			
n-gram category	mean error rate	median error rate	number of n-grams
english-dominant	0.3383	0.3333	109
german-dominant	0.3300	0.2800	45
neutral	0.2962	0.2652	251

Table 7.10: Aggregated bigram error rates by category, German-trained model evaluated on English handwriting

Trigram error distribution			
n-gram category	mean error rate	median error rate	number of n-grams
english-dominant	0.4527	0.4412	1181
german-dominant	0.3965	0.3333	206
neutral	0.3889	0.3333	1033

Table 7.11: Aggregated bigram error rates by category, German-trained model evaluated on English handwriting

7.3 Discussion

7.3.1 Discussion of N-gram Analysis

The results of the n-gram error analysis suggest that language-dominant character sequences do account for a noticeable share of recognition errors in cross-lingual HTR. When the English-trained model was applied to German handwriting, over 50 % of trigram-level errors occurred within German-dominant sequences. Similarly, when the German-trained model processed English handwriting, more than half of the trigram errors involved English-dominant patterns.

At first glance, this supports the assumption that underrepresented linguistic patterns in the training data are more prone to recognition errors. However, a closer look reveals that the difference between language-dominant and language-neutral categories is not as stark as expected. In both evaluation directions, a substantial proportion of errors, between 40 % and 69 % depending on n-gram type, occurred in language-neutral sequences. This suggests that the model's performance is not solely determined by language-specific exposure, but also affected by general recognition limitations, handwriting variability, or token complexity.

Moreover, some language-dominant errors may reflect broader structural challenges in decoding, such as long compound words or overlapping character forms, rather than the absence of the sequence in the training data per se.

Taken together, the analysis confirms that linguistic frequency plays a role in error distribution, but does not fully explain recognition behavior. Language-neutral n-grams continue to account for the majority of observed errors in several conditions. This highlights the need for more nuanced diagnostic tools and evaluation strategies in cross-lingual HTR research.

7.3.2 Discussion of Category-Level Error Rates

The aggregated error rates by n-gram category provide further insight into the linguistic sensitivity of the evaluated HTR models. Across both bigram and trigram levels, the English-trained model showed the highest error rates on German-dominant n-grams, with a mean error rate of 0.55 for bigrams and 0.64 for trigrams.
Interestingly, language-neutral n-grams also exhibited relatively high error rates, indicating that recognition challenges are not limited to strictly language-specific patterns.

If language-specific n-grams were systematically harder to recognize in a cross-lingual setting, we would expect clearly higher error rates for the *foreign* dominant categories. However, the observed values do not support such a pattern.

Across both evaluation directions, the error rates for *english-dominant*, *german-dominant*, and *neutral* n-grams are relatively similar. In some cases, the median error rate is even higher for neutral sequences. This suggests that the linguistic dominance of a sequence alone does not account for recognition errors in a reliable or interpretable way.

But, as already discussed in the main part of the thesis, it is important to note that the overall character and word error rates in the cross-lingual experiments were extremely high. This strongly suggests that other factors likely had a more substantial influence on model failure. These include structural differences between the datasets, such as layout conventions, line segmentation methods, handwriting styles, or broader domain mismatch.

While n-gram categories remain a useful lens for error analysis, the current results do not provide strong evidence for a direct link between n-gram frequency and recognition robustness in cross-lingual HTR. Instead, they point to a more fundamental lack of generalization capacity, likely caused by domain divergence rather than linguistic bias alone. The n-gram-based error rates must therefore be interpreted in the context of these underlying limitations, which likely overshadowed finer-grained linguistic effects.

7.4 Concluding Remarks

The analysis pipeline documented in this appendix was designed to support a more granular understanding of character-level recognition errors in HTR, with a particular focus on language-dominant patterns. Although the main study did not rely on these results for its core evaluation, the full implementation offers a reusable foundation for future error diagnostics and cross-lingual benchmarking.

The modular structure of the scripts and the use of interoperable CSV formats make the pipeline adaptable to other models, languages, or n-gram schemes. For future research, the tagging strategy could be extended to include additional linguistic features such as syllable boundaries, morphemes, or positional statistics within words.

In summary, while this implementation was not used as a primary evaluation method due to model instability on out-of-language data, it represents a methodologically consistent and replicable framework that can inform more robust HTR error analysis in subsequent studies.

References

AlKendi, W. *et al.* (2024) ‘Advancements and Challenges in Handwritten Text Recognition: A Comprehensive Survey’, *Journal of Imaging*, 10(1), p. 18. Available at: https://doi.org/10.3390/jimaging10010018.

Ansari, A. *et al.* (2022) ‘Handwritten Text Recognition using Deep Learning Algorithms’, in *2022 4th International Conference on Artificial Intelligence and Speech Technology (AIST)*. *2022 4th International Conference on Artificial Intelligence and Speech Technology (AIST)*, pp. 1–6. Available at: https://doi.org/10.1109/AIST55798.2022.10065348.

Baek, J. *et al.* (2019) ‘What Is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis’, in. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4715–4723. Available at: https://openaccess.thecvf.com/content_ICCV_2019/html/Baek_What_Is_Wrong_With_Scene_Text_Recognition_Model_Comparisons_Dataset_ICCV_2019_paper.html (Accessed: 28 May 2025).

Biró, A. *et al.* (2023) ‘Synthetized Multilanguage OCR Using CRNN and SVTR Models for Realtime Collaborative Tools’, *Applied Sciences*, 13(7), p. 4419. Available at: https://doi.org/10.3390/app13074419.

Bluche, T., Louradour, J. and Messina, R. (2017) ‘Scan, Attend and Read: End-to-End Handwritten Paragraph Recognition with MDLSTM Attention’, in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1050–1055. Available at: https://doi.org/10.1109/ICDAR.2017.174.

Bušta, M., Patel, Y. and Matas, J. (2019) ‘E2E-MLT - An Unconstrained End-to-End Method for Multi-language Scene Text’, in G. Carneiro and S. You (eds) *Computer Vision – ACCV 2018 Workshops*. Cham: Springer International Publishing, pp. 127–143. Available at: https://doi.org/10.1007/978-3-030-21074-8_11.

Caesar, T. *et al.* (1994) ‘Handwritten word recognition using statistics’, in *IEE European Workshop on Handwriting Analysis and Recognition: A European Perspective*. *IEE European Workshop on Handwriting Analysis and Recognition: A European Perspective*, p. 5/1-5/7. Available at: https://ieeexplore.ieee.org/abstract/document/383965 (Accessed: 4 July 2025).

Chang, D. and Li, Y. (2024) *Mixed Text Recognition with Efficient Parameter Fine-Tuning and Transformer*, *arXiv.org*. Available at: https://arxiv.org/abs/2404.12734v4 (Accessed: 9 June 2025).

Crosilla, G., Klic, L. and Colavizza, G. (2025) ‘Benchmarking Large Language Models for Handwritten Text Recognition’. arXiv. Available at: https://doi.org/10.48550/arXiv.2503.15195.

Dasari, S.K. and Mehta, S. (2023) ‘(PDF) Text detection and recognition through deep learning-based fusion neural network’, *ResearchGate* [Preprint]. Available at: https://doi.org/10.11591/ijai.v12.i3.pp1396-1406.

Dash, S.K. *et al.* (2024) ‘Multi-Lingual Handwritten Recognition Using Convolutional Recurrent Neural Networks’, in *2024 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*. *2024 International*

Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), pp. 1–7. Available at: https://doi.org/10.1109/ICSES63760.2024.10910392.

Diaz, D.H. *et al.* (2021) 'Rethinking Text Line Recognition Models'. arXiv. Available at: https://doi.org/10.48550/arXiv.2104.07787.

Drobac, S. and Lindén, K. (2020) 'Optical character recognition with neural networks and post-correction with finite state methods', *International Journal on Document Analysis and Recognition (IJDAR)*, 23(4), pp. 279–295. Available at: https://doi.org/10.1007/s10032-020-00359-9.

España-Boquera, S. *et al.* (2011) 'Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(4), pp. 767–779. Available at: https://doi.org/10.1109/TPAMI.2010.141.

Evershed, J. and Fitch, K. (2014) 'Correcting noisy OCR: context beats confusion', in *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*. New York, NY, USA: Association for Computing Machinery (DATeCH '14), pp. 45–51. Available at: https://doi.org/10.1145/2595188.2595200.

'IMGUR5K-Handwriting-Dataset' (2025). Meta Research. Available at: https://github.com/facebookresearch/IMGUR5K-Handwriting-Dataset (Accessed: 5 July 2025).

Fujitake, M. (2023) 'DTrOCR: Decoder-only Transformer for Optical Character Recognition'. arXiv. Available at: https://doi.org/10.48550/arXiv.2308.15996.

Garrido-Munoz, C. and Calvo-Zaragoza, J. (2025) 'On the Generalization of Handwritten Text Recognition Models'. arXiv. Available at: https://doi.org/10.48550/arXiv.2411.17332.

Garrido-Munoz, C., Rios-Vila, A. and Calvo-Zaragoza, J. (2025) 'Handwritten Text Recognition: A Survey'. arXiv. Available at: https://doi.org/10.48550/arXiv.2502.08417.

Graves, A. *et al.* (2009) 'A Novel Connectionist System for Unconstrained Handwriting Recognition', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5), pp. 855–868. Available at: https://doi.org/10.1109/TPAMI.2008.137.

Graves, A. and Schmidhuber, J. (2008) 'Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks', in *Advances in Neural Information Processing Systems*. Curran Associates, Inc. Available at: https://proceedings.neurips.cc/paper/2008/hash/66368270ffd51418ec58bd793f2d9b1b-Abstract.html (Accessed: 28 May 2025).

Guillevic, D. and Suen, C.Y. (1997) 'HMM word recognition engine', in *Proceedings of the Fourth International Conference on Document Analysis and Recognition*. *the Fourth International Conference on Document Analysis and Recognition*, pp. 544–547 vol.2. Available at: https://doi.org/10.1109/ICDAR.1997.620559.

He, Y., Chen, M.-Y. and Kundu, A. (1992) 'Handwritten word recognition using HMM with adaptive length Viterbi algorithm', in *[Proceedings] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*. *[] ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 153–156 vol.3. Available at: https://doi.org/10.1109/ICASSP.1992.226253.

Hecht, R., Riedler, J. and Backfried, G. (2002) ‘Fitting German into N-Gram Language Models’, in P. Sojka, I. Kopeček, and K. Pala (eds) *Text, Speech and Dialogue*. Berlin, Heidelberg: Springer, pp. 341–346. Available at: https://doi.org/10.1007/3-540-46154-X_49.

Hodel, T. *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(13), pp. 1–10.

Hu, J. *et al.* (2020) ‘XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalization’. arXiv. Available at: https://doi.org/10.48550/arXiv.2003.11080.

Joshi, C. *et al.* (2023) ‘CENSUS-HWR: a large training dataset for offline handwriting recognition’. arXiv. Available at: https://doi.org/10.48550/arXiv.2305.16275.

Kang, L. *et al.* (2020) ‘Pay Attention to What You Read: Non-recurrent Handwritten Text-Line Recognition’. arXiv. Available at: https://doi.org/10.48550/arXiv.2005.13044.

Keysers, D. *et al.* (2017) ‘Multi-Language Online Handwriting Recognition’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), pp. 1180–1194. Available at: https://doi.org/10.1109/TPAMI.2016.2572693.

Kim, G. and Govindaraju, V. (1997) ‘A lexicon driven approach to handwritten word recognition for real-time applications’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4), pp. 366–379. Available at: https://doi.org/10.1109/34.588017.

Koch, P. *et al.* (2023) ‘A tailored Handwritten-Text-Recognition System for Medieval Latin’. arXiv. Available at: https://doi.org/10.48550/arXiv.2308.09368.

Kotani, A., Tellex, S. and Tompkin, J. (2020) ‘Generating Handwriting via Decoupled Style Descriptors’, in A. Vedaldi et al. (eds) *Computer Vision – ECCV 2020*. Cham: Springer International Publishing, pp. 764–780. Available at: https://doi.org/10.1007/978-3-030-58610-2_45.

LeCun, Y., Bengio, Y. and Hinton, G. (2015) ‘Deep learning’, *Nature*, 521(7553), pp. 436–444. Available at: https://doi.org/10.1038/nature14539.

Leipzig Corpora Collection - Wortschatz Deutsch (no date). Available at: https://corpora.uni-leipzig.de/de?corpusId=deu_news_2024 (Accessed: 10 July 2025).

Li, M. *et al.* (2022) ‘TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models’. arXiv. Available at: https://doi.org/10.48550/arXiv.2109.10282.

Li, Y. *et al.* (2025) ‘HTR-VT: Handwritten Text Recognition with Vision Transformer’, *Pattern Recognition*, 158, p. 110967. Available at: https://doi.org/10.1016/j.patcog.2024.110967.

Liu, Y. *et al.* (2021) ‘Transformer in Convolutional Neural Networks’. arXiv. Available at: https://doi.org/10.48550/arXiv.2106.03180.

Marti, U.-V. and Bunke, H. (2002) ‘The IAM-database: an English sentence database for offline handwriting recognition’, *International Journal on Document Analysis and Recognition*, 5(1), pp. 39–46. Available at: https://doi.org/10.1007/s100320200071.

Martinger, H. (2013) *Terms of endearment in American Soap Operas : A corpus study of honey, sweetheart and darling*. Available at: https://urn.kb.se/resolve?urn=urn:nbn:se:lnu:diva-24299 (Accessed: 5 July 2025).

Mathew, M., Singh, A.K. and Jawahar, C.V. (2016) ‘Multilingual OCR for Indic Scripts’, in *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*. *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, pp. 186–191. Available at: https://doi.org/10.1109/DAS.2016.68.

Mermelstein, P. and Eyden, M. (1964) ‘A system for automatic recognition of handwritten words’, in *Proceedings of the October 27-29, 1964, fall joint computer conference, part I*. New York, NY, USA: Association for Computing Machinery (AFIPS ’64 (Fall, part I)), pp. 333–342. Available at: https://doi.org/10.1145/1464052.1464081.

Moss, L. (2015) *Corpus Stylistics and Henry James’s Syntax*, *Doctoral thesis, UCL (University College London)*. Doctoral. UCL (University College London). Available at: https://discovery.ucl.ac.uk/id/eprint/1461029/ (Accessed: 5 July 2025).

Namysl, M. and Konya, I. (2019) ‘Efficient, Lexicon-Free OCR using Deep Learning’. arXiv. Available at: https://doi.org/10.48550/arXiv.1906.01969.

Neat, L. *et al.* (2019) ‘Scene text access: a comparison of mobile OCR modalities for blind users’, in *Proceedings of the 24th International Conference on Intelligent User Interfaces*. New York, NY, USA: Association for Computing Machinery (IUI ’19), pp. 197–207. Available at: https://doi.org/10.1145/3301275.3302271.

Neto, A.F. de S., Bezerra, B.L.D. and Toselli, A.H. (2020) ‘Towards the Natural Language Processing as Spelling Correction for Offline Handwritten Text Recognition Systems’, *Applied Sciences*, 10(21), p. 7711. Available at: https://doi.org/10.3390/app10217711.

Nguyen, T.-T.-H. *et al.* (2019) ‘Post-OCR Error Detection by Generating Plausible Candidates’, in *2019 International Conference on Document Analysis and Recognition (ICDAR)*. *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 876–881. Available at: https://doi.org/10.1109/ICDAR.2019.00145.

Papers with Code - Machine Learning Datasets (no date). Available at: https://paperswithcode.com/datasets?q=&v=lst&o=newest&task=handwriting-recognition&lang=german&mod=texts (Accessed: 10 July 2025).

Peng, X. *et al.* (2013) ‘Multilingual OCR research and applications: an overview’, in *Proceedings of the 4th International Workshop on Multilingual OCR*. New York, NY, USA: Association for Computing Machinery (MOCR ’13), pp. 1–8. Available at: https://doi.org/10.1145/2505377.2509977.

Pippi, V., Cascianelli, S. and Cucchiara, R. (2023) ‘Handwritten Text Generation from Visual Archetypes’, in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22458–22467. Available at: https://doi.org/10.1109/CVPR52729.2023.02151.

Practical Cryptography (no date). Available at: http://practicalcryptography.com/cryptanalysis/letter-frequencies-various-languages/ (Accessed: 10 July 2025).

Sánchez, J.A. *et al.* (2019) ‘A set of benchmarks for Handwritten Text Recognition on historical documents’, *Pattern Recognition*, 94, pp. 122–134. Available at: https://doi.org/10.1016/j.patcog.2019.05.025.

Sharma, N. *et al.* (2015) ‘Multi-lingual text recognition from video frames’, in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. *2015 13th*

International Conference on Document Analysis and Recognition (ICDAR), pp. 951–955. Available at: https://doi.org/10.1109/ICDAR.2015.7333902.

de Sousa Neto, A.F. *et al.* (2024) 'Data Augmentation for Offline Handwritten Text Recognition: A Systematic Literature Review', *SN Computer Science*, 5(2), p. 258. Available at: https://doi.org/10.1007/s42979-023-02583-6.

Ströbel, P.B. *et al.* (2022) 'Evaluation of HTR models without Ground Truth Material'. arXiv. Available at: https://doi.org/10.48550/arXiv.2201.06170.

Toselli, A.H. *et al.* (2018) 'HTR Dataset ICFHR 2016'. Zenodo. Available at: https://zenodo.org/records/1297399 (Accessed: 5 July 2025).

Yuting (2025) 'YutingLi0606/HTR-VT'. Available at: https://github.com/YutingLi0606/HTR-VT (Accessed: 10 July 2025

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich das vorliegende Dokument selbständig und nur mit den angegebenen Hilfsmitteln verfasst habe. Alle Passagen, die ich wörtlich aus der Literatur oder aus anderen Quellen wie z. B. Internetseiten übernommen habe, habe ich deutlich als Zitat mit Angabe der Quelle kenntlich gemacht.

10.07.2025, Hamburg	
Datum	Unterschrift