Transforming Handwriting into Language: A Sentence-Level Recognition Framework

Teng, Q. E.
Department of Computer Science
University of Toronto
Toronto, ON M5S 1A1
e.teng@mail.utoronto.ca

Tian, R.
Department of Computer Science
University of Toronto
Toronto, ON M5S 1A1
ricky.tian@mail.utoronto.ca

Wang, S.
Department of Computer Science
University of Toronto
Toronto, ON M5S 1A1
shunqi.wang@mail.utoronto.ca

Zhou, Z.
Department of Computer Science
University of Toronto
Toronto, ON M5S 1A1
ziheng.zhou@mail.utoronto.ca

Abstract

Transformer architectures have become effective for sequence-to-sequence tasks due to their global self-attention mechanisms. We present a full-sentence English handwriting recognition framework based on the pre-trained TrOCR model, composed of a Vision Transformer (ViT) encoder and an auto-regressive Transformer decoder. Fine-tuned on the GNHK dataset, the model adapts to handwriting-specific challenges such as style variation, slant, and noise factors that make sentence-level recognition difficult due to high variability in human writing. To enhance long coherence, we additionally incorporate a lightweight sentence reconstruction module. Evaluation using Character Error Rate (CER) shows that vision-language pre-training provides a strong foundation for robust sentence-level handwriting recognition, extending Transformer-based OCR toward broader multimodal document understanding.

1 Introduction

Automatic handwriting recognition (HWR) aims to convert handwritten text into machine-readable sequences, enabling applications such as digital archiving, educational assessment, and form processing. While earlier research has focused on recognizing isolated characters or words, full-sentence handwriting recognition is a new approach that aims towards more complex format recognition.

Beyond simple character identification, sentence-level recognition requires the model to capture long-range contextual dependencies, linguistic coherence, and the hierarchical structure of language. Variations in handwriting style, spacing, and punctuation further complicate this task, making robust generalization across writers and writing conditions an ongoing challenge.

Traditional HWR systems, such as Convolutional Neural Networks (CNNs) or Convolutional Recurrent Neural Networks (CRNNs), achieve reasonable performance on short, constrained text. However, their ability to maintain coherence across full sentences is limited by a reliance on localized features and recurrent mechanisms, which often struggle to capture long-distance dependencies effectively. Moreover, these architectures frequently depend on handcrafted preprocessing steps, which can introduce cascading errors and restrict adaptability to new handwriting styles.

The emergence of Transformer architectures has revolutionized sequence modeling by introducing self-attention, enabling models to capture relationships between all elements in a sequence simultane-

ously. The TrOCR (Transformer-based Optical Character Recognition) model extends this capability to image-to-text tasks by combining a Vision Transformer (ViT) encoder and a Transformer decoder in a unified sequence-to-sequence framework. This architecture eliminates explicit segmentation and enables direct end-to-end mapping from images to text sequences, making it particularly well-suited for sentence-level HWR.

Our approach leverages this Transformer-based framework to address the full-sentence recognition problem. The encoder, composed of multiple multi-head self-attention and feed-forward layers, processes a handwriting image by dividing it into small patches and embedding them into positional vectors. These visual tokens provide the decoder with the visual semantics necessary for text generation. The decoder then generates the transcription sequentially through masked self-attention and cross-attention layers, ensuring each token depends only on previously produced outputs. This structure aligns visual and linguistic information while preserving the natural flow of written text.

The contribution of the paper is as follows:

1. We developed a full-sentence HWR framework based on the TrOCR encoder–decoder architecture.
2. We provide a detailed architectural analysis of how multi-head self-attention and cross-attention jointly enable sentence-level transcription.
3. We demonstrate that fine-tuning a pre-trained vision-language model on the GNHK dataset significantly improves recognition accuracy and contextual coherence compared to conventional ViT models.

The findings suggest that large-scale Transformer pre-training provides an effective foundation for handwriting recognition that extends beyond isolated words to sentence-level understanding, paving the way for future work in complex document analysis and historical text restoration.

2 Background and Related Work

2.1 Early Handwriting Recognition Approaches

Early HWR relied on handcrafted features and statistical models such as Hidden Markov Models (HMMs) [4], which modeled characters as probabilistic state sequences. While effective on small datasets, HMM-based systems struggled with complex spatial variations, cursive writing, and writer-specific distortions.

2.2 CNN-RNN Architectures and the CTC Paradigm

CNNs shifted the field toward learned visual representations, capturing local structures such as edges and curves [5]. To model temporal dependencies, hybrid CNN-HMM and CNN-RNN frameworks emerged. CRNNs [6] combined CNN feature extraction with LSTM-based sequence modeling [7], enabling end-to-end recognition. However, their sequential recurrence limits scalability to full sentences, leading to error accumulation and poor modeling of long-range context. These limitations motivate architectures with global attention for sentence-level HWR.

2.3 Transformer Architectures in Vision and Text

Transformers [8] introduced self-attention, enabling each token to attend to all others and eliminating recurrence. This capacity to capture global dependencies makes Transformers better suited than RNNs for long, variable handwriting sequences—an essential property leveraged by our model.

2.4 Transformer-Based OCR and Handwriting Recognition

Transformer-based OCR systems extend self-attention to image-to-text tasks. TrOCR [3] combines a ViT encoder with a Transformer decoder and benefits from vision-language pre-training, improving robustness to stylistic variation, noise, and ambiguous handwriting. ViT patch embeddings, however, may struggle with curved or irregular text layouts, a limitation we mitigate through handwriting-specific fine-tuning.

2.5 Full-Sentence Handwriting Recognition

Recent Transformer-based HWR systems, such as TRBA [9], integrate visual and linguistic cues but often remain word-level or require custom modifications. By fine-tuning TrOCR on the GNHK dataset, our approach extends Transformer OCR to full-sentence transcription, aligning visual and linguistic information while handling noise, slant, and handwriting variability.

3 Dataset Exploration

3.1 Dataset Selection and Justification

Recognizing handwritten study notes poses challenges not reflected in standard datasets. Collections like the IAM Handwriting Database [2] consist of clean, expert-written samples that serve well for benchmarking but fail to capture the rushed, inconsistent handwriting typical in real notes. As a result, models trained solely on IAM often generalize poorly to authentic educational settings.

To bridge this gap, we use the **GNHK Handwritten Notes Dataset** [1], which contains diverse, unconstrained handwriting from multiple regions and page layouts. Its variability more closely matches real student notes, making it better suited for our task.

3.2 Dataset Statistics and Written Text Visualization

Table 1 presents the statistics of characters, texts, and lines for each region. These statistics are summarized from the paper introducing the **GNHK Handwritten Notes Dataset** [1]. Figure 1 provides an example of handwriting images captured under unconstrained conditions (After preprocessing). As shown in the figure, the words appear irregular, and some symbols are not recognizable.

Table 1: Statistics of characters, texts, and lines for each region.

Region	# Characters	# Texts	# Lines
Europe (EU)	58,982	13,592	3,306
North America (NA)	47,361	10,967	2,099
Asia (AS)	39,593	8,586	2,780
Africa (AF)	27,000	5,881	1,178
Total	**172,936**	**39,026**	**9,363**

Figure 1: Example of the handwriting images and their label. (After preprocessing).

3.3 Dataset Structure

In this section, we describe the structure of the **GNHK Handwritten Notes Dataset** [1]. Each image in the training and testing set is paired with a JSON file that provides detailed annotations of the text regions in the image. For example, the image `eng_AF_004.jpg` has a corresponding file `eng_AF_004.json`. Each JSON file contains a list of annotation objects, where each object represents one continuous text region detected in the image. The main fields in each object are:

- **"text"** – The text for each region is stored without spaces and mostly uses standard ASCII characters. When a character cannot be represented in ASCII, a special token is used instead.
- **"polygon"** – defines the four corner points of the bounding box around the text, given by $\{x_0, y_0;\ x_1, y_1;\ x_2, y_2;\ x_3, y_3\}$

Here is an example of a JSON annotation used in the dataset:


```
{"text":"ownership","polygon":{"x0":1347,"y0":1606,"x1":2238,"y1":1574,"x2":2170,"y2
    ":1884,"x3":1300,"y3":1747}, "line_idx":4,"type":"H"}
```


3.4 Data Pre-processing

We first split each note image into smaller word crops, then pair each cropped region with its ground-truth label—the correct word shown in that crop. Figure 1 shows an example of how the dataset appears after the pre-processing step.

Using the JSON annotations described in the previous section, this process becomes straightforward. For every text object in the JSON file, we use the `polygon` coordinates to crop the corresponding region from the image and pair it with the text content stored in the `"text"` field.

After the word crop enters the model, it is first resized to the expected ViT input resolution, producing a tensor of shape $3 \times 224 \times 224$ before patching. The encoder then divides this tensor into $14 \times 14 = 196$ patches, flattens each patch, and projects it to a 768-dimensional embedding. After adding the CLS token, the ViT processes a total of 197 tokens, each of size 768, giving an encoder input and output shape of $(197, 768)$ for every image.

4 Model Architecture

A detailed overview of the model architecture is presented in Figure 2.

4.1 Architecture Selection and Fine-Tuning

Training a vision–language Transformer from scratch for this task is highly computationally expensive. To reduce both time and resource requirements, we fine-tune a pre-trained TrOCR model instead. The model is based on `trocr-small-handwritten` [3], and we fine-tune it on the **GNHK Handwritten Notes Dataset** [1]. The TrOCR paper notes that the model is primarily trained and tested on cropped text-line images and performs less effectively on complex inputs. Therefore, fine-tuning TrOCR on the GNHK dataset, which contains handwritten notes captured under unconstrained conditions, provides a meaningful opportunity to improve its performance on real-world handwriting.

During fine-tuning, the pre-trained TrOCR model is loaded into a sequence-to-sequence setup in which each cropped ViT encodes a word image, and the decoder generates the corresponding text transcription.

4.2 Word-to-Sentence Concatenation

The GNHK dataset is primarily designed for *word-level* recognition. Each image is accompanied by a JSON annotation file where every text object corresponds to a single word. To extend this setup for *sentence-level* recognition, we reconstruct the reading order of the words based on their spatial arrangement in the image. Each annotated word includes its bounding polygon. We order the words by sorting their top-left corner coordinates, first by y and then by x, which follows the natural reading order from top to bottom and left to right. After concatenating the ordered tokens, we use a Large Language Model (LLM) to refine the generated text. We choose Phi-3.5 Mini [10] because it is lightweight and well-suited for local deployment. All models used in this report, including the baselines, rely on Phi-3.5 Mini for text refinement. The LLM's role is to insert sentence boundaries and improve readability. In this way, we transform isolated word-level predictions into paragraphs.

4.3 Training Details

The main hyperparameters we tune are the learning rate, the number of fine-tuning epochs, and the number of frozen encoder blocks in the ViT encoder. Due to GPU memory constraints on our local hardware (an RTX 4080 Laptop GPU with 12 GB of VRAM), we fix the batch size to 24 for all experiments. All models are trained using the Hugging Face `Seq2SeqTrainer` with the default AdamW optimizer and a linear learning-rate schedule.

Our final model uses the `microsoft/trocr-small-handwritten` checkpoint as the initialization and fine-tunes it for 3 epochs with a learning rate of 3×10^{-5} and batch size 24. We freeze the first 4 encoder blocks of the ViT backbone to preserve low-level visual features while allowing higher layers and the decoder to adapt to the GNHK domain. Unless otherwise stated, all other Transformer hyperparameters (hidden size, number of heads, feed-forward dimensions) follow the default configuration of the pre-trained checkpoint.

For sequence generation, we use beam search with 4 beams, a maximum output length of 64 tokens, a no-repeat n-gram size of 3, and a length penalty of 2.0. We set the decoder start token to the tokenizer `[CLS]` token, and use the tokenizer `[EOS]` and `[PAD]` tokens as the end-of-sequence and padding tokens, respectively. Mixed-precision training with FP16 is enabled when a compatible GPU is available.

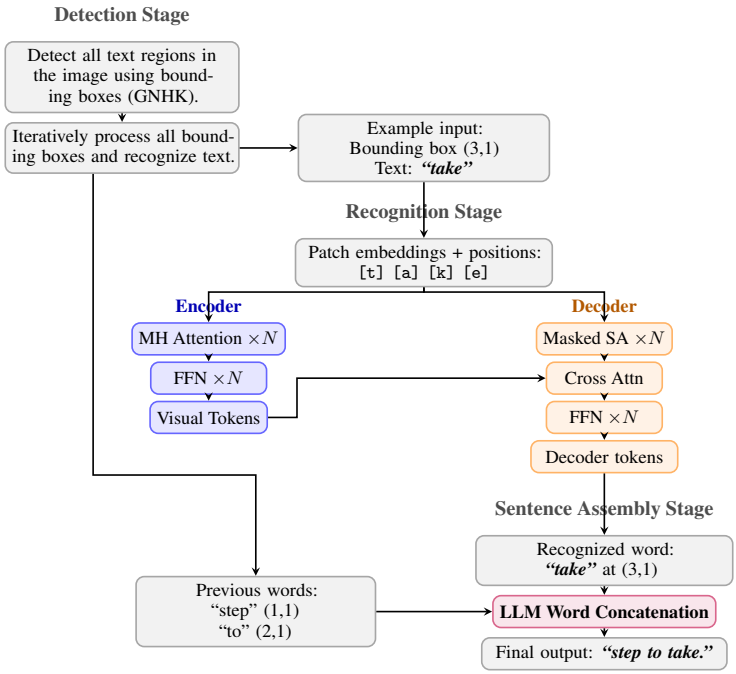

Figure 2: Pipeline overview. Each detected word is cropped and recognized by the Transformer encoder–decoder (example shown for ***"take"*** at top-left corner $(3, 1)$). Previously recognized words are passed to an LLM-based module that concatenates them into the final sentence ***"step to take."***.

5 Results

5.1 Evaluation on Word-Crop Recognition

In this section, we present the fine-tuned model's performance on cropped word recognition, corresponding to the "Recognition Stage" in Figure 2. The model's predicted word is compared with the ground-truth label for each crop, and the evaluation metric used is the Character Error Rate (CER). CER measures the character-level difference between a predicted string p and its ground-truth label g using the Levenshtein edit distance. Formally,

$$\text{CER} = \frac{S + D + I}{N},$$

where S, D, and I denote the number of substitutions, deletions, and insertions required to transform p into g, and N is the total number of characters in the ground-truth string. A CER of 0 indicates perfect transcription, while higher values reflect worse model performance.

We compare our model's performance against two baselines, `trocr-small-handwritten` and `trocr-base-handwritten`, both introduced in [3]. The base variant is larger and more expressive

than the small model, and both pretrained models are trained on the same collection of handwritten notes. For a fair comparison, all models—including our fine-tuned version—are evaluated on the held-out test split that none of them have seen during training.

Unless otherwise stated, the CER reported for our model in this section corresponds to the best-performing configuration identified in our hyperparameter sweep (Section 5.3): three epochs of fine-tuning with a learning rate of 3×10^{-5}, batch size 24, and freezing the first four encoder blocks. Table 2 reports the average CER computed across all cropped test images. The CER results show that our fine-tuned model substantially improves over both pretrained baselines on the GNHK word-crop recognition task.

Table 2: **CER performance evaluation across the test set**

Model	AVG Test CER
`trocr-small-handwritten`	1.0440
`trocr-base-handwritten`	0.5942
Fine-tuned Model (ours)	**0.2764**

5.2 Evaluation on Paragraph-Level Generation

Beyond word-level accuracy, real-world handwriting applications require reconstructing full paragraphs of notes. Since the GNHK dataset does not include paragraph-level annotations, we approximate a reference by concatenating all ground-truth word labels for each note and prompting the LLM to organize them into a coherent paragraph. This LLM-generated text serves as a consistent comparison target.

Both our fine-tuned model and the `trocr-base-handwritten` baseline are evaluated on this task. Each model predicts the cropped words for a note, assembles these predictions into a raw transcript, and passes it to the same LLM for paragraph reconstruction. We compute cosine similarity between each model's output and the reference paragraph. As shown in Table 3, our model achieves a substantially higher similarity score, indicating improved paragraph-level coherence and reconstruction quality.

Table 3: **Average cosine similarity with ground-truth paragraphs**

Model	Cosine Similarity
`trocr-base-handwritten`	0.5421
Fine-tuned Model (ours)	**0.7070**
Overall Improvement	**0.1649**

5.3 Hyperparameter Tuning Results and CER

To understand how different choices of training hyperparameters affect recognition quality, we ran a small sweep over the learning rate, number of fine-tuning epochs, and the number of frozen encoder blocks in the ViT. The batch size was fixed to 24 in all runs. Concretely, we varied:

- epochs: 1 or 3;
- learning rate: 3×10^{-5} or 5×10^{-5};
- number of frozen encoder blocks: 0, 2, or 4.

Table 4 summarizes five representative configurations and reports, for each, the best validation CER achieved during training and the final test CER measured on the held-out test split.

Several trends emerge from this sweep. First, comparing `E1` and `E2` shows that a smaller learning rate (3×10^{-5}) consistently reduces both validation and test CER for a fixed number of epochs and frozen blocks. Second, increasing the number of epochs from 1 to 3 while also freezing more encoder blocks (`E2` vs. `E4`) yields the best overall performance: configuration `E4_ep3_lr3e-5_fr4` achieves

Table 4: **Representative hyperparameter configurations and CER.** All runs use batch size 24. Validation CER is reported at the best epoch for that configuration.

ID	Epochs	LR	Frozen blocks	Val CER / Test CER
E1_ep1_lr5e-5_fr2	1	5×10^{-5}	2	0.316 / 0.2970
E2_ep1_lr3e-5_fr2	1	3×10^{-5}	2	0.308 / 0.2877
E3_ep3_lr5e-5_fr2	3	5×10^{-5}	2	0.313 / 0.3702
E4_ep3_lr3e-5_fr4	3	3×10^{-5}	4	**0.291 / 0.2764**
E5_ep1_lr5e-5_fr0	1	5×10^{-5}	0	0.299 / 0.2819

the lowest validation CER and the best test CER of 0.2764. In contrast, using a higher learning rate with three epochs (`E3`) leads to a noticeably worse test CER (0.3702), suggesting that the model starts to overfit and that optimization becomes less stable at 5×10^{-5}.

We also experimented with training for more than three epochs using similar settings. In these longer runs, the training loss continued to decrease monotonically, but the validation CER reached a minimum around the third epoch and then began to increase. In one extreme case, the validation CER briefly spiked above 8.0 after the sixth epoch, indicating a breakdown of the decoding behaviour. These loss and CER trajectories support our choice of three fine-tuning epochs and a learning rate of 3×10^{-5} with four frozen encoder blocks as a good trade-off between optimization and generalization for sentence-level handwriting recognition on GNHK.

For full reproducibility, the hyperparameter sweep script `experiments/run_experiments.py` and all logged results `experiments/log.txt` are included in the project repository.

6 Discussion

In this section, we discuss how, after fine-tuning on the GNHK dataset, our model achieves improved performance in recognizing handwritten notes.

6.1 Performance of Word-Crop Recognition

The model's word-level recognition capability is reflected by the CER metric. As shown in Table 2, our fine-tuned model achieves a substantially lower average CER, indicating that its predictions for individual word crops more closely match the ground-truth labels from the GNHK dataset compared to the non-fine-tuned baselines. Notably, although our model is fine-tuned starting from `trocr-small-handwritten`, it still outperforms the larger and more complex `trocr-base-handwritten` model after fine-tuning, demonstrating that our fine-tuning strategy effectively adapts the model to the specific handwriting patterns present in student note images.

6.2 Performance of Paragraph-Level Generation

This task is closer to real-world applications and is more challenging, as the quality of the generated paragraph depends directly on the raw notes (the list of predicted words extracted from each note) fed into the LLM, and therefore ultimately on the accuracy of the individual word predictions. Therefore, we include only `trocr-base-handwritten` as the baseline model, since it provides reasonable performance on the word-level recognition task. As shown in Table 3, our fine-tuned model achieves a substantially higher cosine similarity than the baseline, demonstrating that the paragraphs it produces more closely match those generated from the ground-truth word labels.

6.3 Impact of Hyperparameter Settings

The hyperparameter sweep in Section 5.3 (Table 4) shows that the learning rate has the largest impact on generalization. For a fixed number of frozen encoder blocks, using a smaller learning rate of 3×10^{-5} consistently improves both validation and test CER compared to 5×10^{-5}. For example, configuration `E2` achieves a lower test CER than `E1` despite identical architectural settings. In contrast, when the learning rate is kept high (5×10^{-5}), increasing the number of epochs from 1 to 3 (as in `E3`) reduces the training loss but noticeably worsens the test CER, indicating overfitting.

The number of frozen encoder blocks also plays a meaningful role. Comparing `E2` and `E4` shows that freezing more early ViT layers improves validation stability and generalization by preserving strong pretrained priors. This prevents overspecialization in the specific handwriting domain. Conversely, updating all encoder blocks (as in `E5`) produces slightly worse CER despite similar training loss, suggesting reduced robustness.

Taken together, the results indicate that a moderate learning rate, a small number of fine-tuning epochs, and partially freezing the encoder provide the best balance between optimization and generalization. This rationale supports our final choice of three epochs, a learning rate of 3×10^{-5}, and freezing the first 4 encoder layers as the default configuration for our final model. These hyperparameters consistently produce stable training dynamics, lower validation loss, and the best test CER among all tested settings. Overall, the ablation results highlight that careful control of learning rate and encoder freezing is crucial for effective and robust fine-tuning of TcOCR on the GNHK dataset.

7 Limitations

Missing Word-Level Localization:
Our system currently relies on GNHK's ground-truth bounding boxes and lacks an integrated word detection module. Real-world application, however, requires detecting words within cluttered, free-form layouts. Integrating a robust detector for unconstrained handwriting remains future work.
Ground Truth Instability:
Since GNHK lacks paragraph annotations, we approximate ground truth by organizing word labels via an LLM. However, LLM introduces variability in these references, limiting metric reproducibility. Reliable evaluation will require human-annotated paragraph data.

8 Ethical Considerations

Data Privacy and Algorithmic Bias:
Because the GNHK dataset contains "in the wild" handwriting, it may include sensitive PII, and models trained on such data could facilitate privacy violations if applied to personal journals or notes. Demographic imbalance (Table 1) also risks systematic bias, particularly for underrepresented regions. In settings such as education or form assessment, this may produce allocative harm for individuals with atypical handwriting patterns or disabilities.
Authenticity and Reliability:
Our reliance on an LLM (Phi-3.5 Mini) introduces dual risks regarding textual integrity and factual reliability. First, the model's design risks transforming transcription into content modification. This threatens the authenticity of historical archives and invalidates educational assessments. Also, this generative capacity creates a risk of hallucination, where the model invents plausible but factually incorrect text (e.g., altering medical dosages) from ambiguous handwriting. The system lacks uncertainty quantification to flag these potential errors. Consequently, users may succumb to automation bias, blindly trusting the coherent, "corrected" output over the messy original truth.
Misuse and Societal Impact:
Automated handwriting recognition reduces barriers that once limited large-scale analysis of handwritten materials, potentially enabling mass surveillance or intrusive indexing by institutional actors. In operational settings such as loan or medical processing, errors and demographic biases can translate directly into unequal access to services for groups with non-standard handwriting.

9 Conclusion

In this work, we present a complete pipeline for paragraph-level recognition of messy, unconstrained student handwriting. By fine-tuning the TrOCR model on the GNHK dataset, we adapt a vision–language Transformer to the unique challenges of real-world handwritten notes. The fine-tuned model achieves strong word-level performance, substantially reducing Character Error Rate compared to pretrained baselines. Building on these predictions, we introduce an LLM-based reconstruction module that restores sentence structure and improves overall paragraph coherence. Together, these components yield state-of-the-art results in both character-level transcription and paragraph-level similarity, demonstrating the effectiveness of combining domain-specific fine-tuning with lightweight language modeling for complex handwriting recognition.

References

1. Lee, A.W.C., Chung, J. & Lee, M. (2021) GNHK: A Dataset for English Handwriting in the Wild. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*.
2. Marti, U.-V. & Bunke, H. (2002) The IAM-database: An English sentence database for offline handwriting recognition. In *International Journal on Document Analysis and Recognition*, 5(1), pp. 39–46. DOI: 10.1007/s100320200071.
3. Li, M., Lv, T., Cui, L., Lu, Y., Florencio, D., Zhang, C., Shafai, F., Wang, H., Chang, M., & Wei, F. (2021) TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models. *arXiv preprint arXiv:2109.10282*.
4. Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE, 77(2), 257–286.*
5. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. In *Proceedings of the IEEE, 86(11), 2278–2324.*
6. Shi, B., Bai, X., & Yao, C. (2016). An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 39(11), 2298–2304.*
7. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. In *Neural Computation, 9(8), 1735–1780.*
8. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS), 6000–6010.*
9. Baek, J., Kim, G., Baek, S., Park, S., & Lee, H. (2019). What is wrong with scene text recognition model comparisons? Dataset and model analysis. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV), 4715–4723.*
10. Abdin, M., Abhishek, A., Awadalla, A., Bakshy, E., et al. (2024). Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. *arXiv:2404.14219*.

Revision Notes (Based on TA's comments)

- Revised the Abstract to explicitly mention the challenges of unconstrained handwriting (noise, slant, variability), the inclusion of a sentence reconstruction pipeline, and the use of CER and WER as evaluation metrics.
- Strengthened the "Background and Related Work" section by explicitly linking each architecture to our proposed model; specifically, we addressed the scalability limitations of CRNNs, justified TrOCR's robustness for unconstrained handwriting, and noted the challenges ViT encoders face with curved or irregular text.
- Added the shape of the preprocessed input image and the final tensor dimensions in Section 3.4 (*Preprocessing the Data*).
- Included GPU constraints and all hyperparameters used during training, as well as details on whether early ViT encoder layers were frozen. These additions appear in a new subsection titled *Training Details* within the Model section.
- Added Model Hallucination and Uncertainty Quantification into "Authenticity and Reliability" part in section 8.
- Refined paragraph structure for readability, fixed minor grammar issues, ensured consistent acronym usage, and verified citations for all statements.
- Revised work division table available in README.md on GitHub repository.

Source Code

Source code available on GitHub repository: https://github.com/dwz92/Transforming-Handwriting-into-Language-A-Sentence-Level-Recognition-Framework