Making Feedback Visible: Character-Level Error Localization in Handwritten CALL Input

Said Yasin **Torsten Zesch**
CATALPA
Center of Advanced Technology for Assisted Learning and Predictive Analytics
FernUniversität in Hagen, Germany

Abstract

Feedback in CALL should be as precisely localized as possible. For example, instead of merely stating that "there is a spelling error in your statement", it is more helpful to indicate the exact location of the error in the text. In fact, an even higher degree of precision would be to identify the specific characters responsible for the error. This localization of erroneous characters is challenging for handwriting input, a modality still widely used in primary and secondary education especially in language learning contexts. Our demo presents an interactive end-to-end prototype that captures the handwriting input, transcribes it, generates feedback, and visually grounds it in the handwriting on character-level granularity.

1 Introduction

Feedback is a central component of computer-assisted language learning (CALL). In particular, it is most useful when learners can relate it directly to their own production (Ferris and Roberts, 2001). For written language learning, this means that feedback should not only state that an error exists, but also indicate precisely where it occurs. Precise localization reduces the effort required to interpret feedback, supports self-correction, and makes automated feedback more transparent to learners and teachers.

While localized feedback is comparatively straightforward for digitized text, it remains challenging for handwritten input. This is an important limitation, because handwriting continues to play a significant role in primary and secondary education, including both first-language (L1) and second-language (L2) learning contexts. If CALL systems only support keyboard input, they may fail to reflect authentic classroom practices and may exclude younger learners whose digital writing skills are still developing.

For written corrective feedback, Ellis (2009) distinguishes (i) metalinguistic feedback (such as "there is a spelling error in your sentence") from (ii) direct feedback (providing the correct form), and (iii) indirect feedback (only visualizing that an error occurred, but not providing the correct form). In Figure 1, we provide examples for each type. Metalinguistic feedback is not grounded in the output, so it basically works the same for digitized input or handwriting. To ground direct and indirect feedback in the handwriting it becomes at least necessary to localize word boundaries, so that feedback can be precisely placed. The most challenging to implement is indirect feedback at character-level granularity as this requires locating the exact position of characters within a handwritten word. However, it is also the potentially most rewarding as indirect feedback is often argued to promote more active processing and self-correction than direct feedback (Lalande, 1982), although empirical findings are mixed (Chandler, 2003; Bitchener and Knoch, 2010; Van Beuningen et al., 2012) depending for example on the learner's proficiency level (Kang and Han, 2015).

Grounding localized feedback in handwriting requires solving several connected problems. First, the system must capture the handwritten production. Second, it must transcribe the handwriting into machine-readable text. Third, it must generate pedagogically meaningful feedback on the transcribed content. Finally, and crucially, the feedback must be mapped back onto the original handwriting. This final step is non-trivial: even if the transcription is correct, the system must know which region of the handwritten text corresponds to the word and character that triggered the feedback.

In this demo paper, we present an interactive

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Licence details: http://creativecommons.org/licenses/by/4.0/.

Proceedings of the 15th Workshop on Natural Language Processing for Computer Assisted Language Learning (NLP4CALL 2026)
pages 34–39, November 19–20, 2026, University of Tartu Library

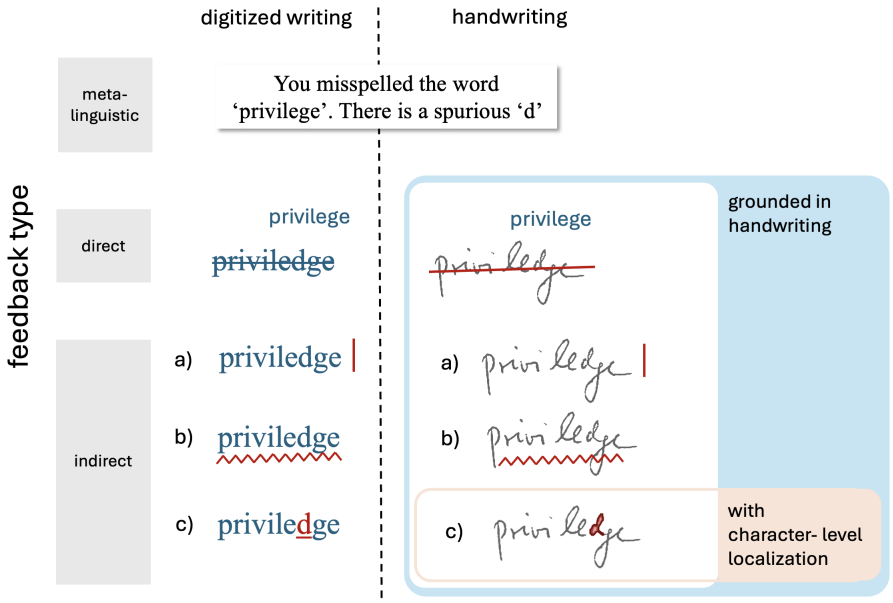


Figure 1: Examples of feedback types in digitized text and when grounded in handwriting.

end-to-end prototype for visually grounded feedback on handwritten learner input. The prototype is implemented as a mobile application for smartphones and tablets and runs fully offline on the device. Figure 2 provides a screenshot of the demo where corrective spelling feedback (as one example from a wider range of possible applications including morphology, syntax, lexical choice, etc.) is grounded in the handwriting and localized on character level.[1]

By keeping the learner's original handwritten production visible, the system preserves the modality in which the task was completed. At the same time, it connects this production to computational processing steps that are typically available only for typed input.

With our demo system, we aim to stimulate discussion about the role of handwriting in CALL, the design of localized feedback for young learners, and the technical requirements for grounding language feedback in multi-modal learner data. The groundink mobile application and its source code are available at https://github.com/said702/groundink.

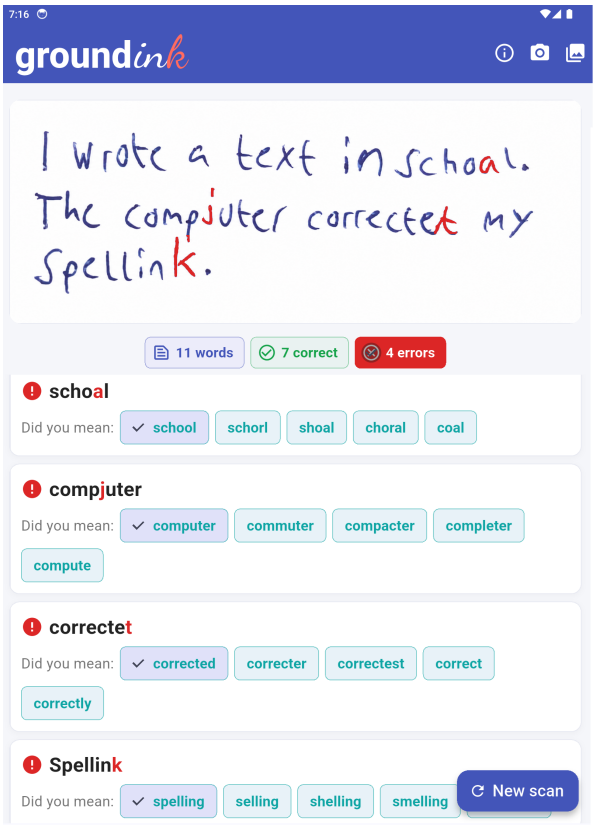


Figure 2: Character-level spelling feedback as implemented in the demo.

2 System Architecture

The prototype is implemented as an on-device mobile application. Its processing pipeline consists of

[1] We call our system *groundink*, a portmanteau of 'grounding' and 'ink'. At the same time, a misspelling of 'grounding' in need of localized feedback.

four main stages: word detection, character recognition and localization, feedback generation, and spatial grounding of the feedback in the original handwriting.

The application is implemented in Flutter, a cross-platform framework for mobile application development, using the Dart programming language. This allows us to use a single codebase for both Android and iOS.

Handwriting input is captured either through the device camera or from the local photo gallery using the Flutter image picker. All recognition and feedback components are executed locally on the device to support the privacy requirements of in-school applications. Therefore, the models are stored in the platform-independent Open Neural Network Exchange (ONNX) format and executed using ONNX Runtime, allowing the application to operate without an external server connection.

Word Detection. The first stage uses the nano-sized YOLO26n model[2] to detect handwritten words in the input image and predict a bounding box for each word. We fine-tune this model on the **IAM Handwriting Database** (Marti and Bunke, 2002), which contains 1,539 scanned handwritten English forms from 657 writers and 115,320 word instances. The predicted bounding boxes are used to extract the corresponding word regions from the original image. For display purposes, the detected words are ordered by line from top to bottom and from left to right within each line. Since each word is processed independently, the reading order does not affect the evaluation. While the model was trained on English data, it mainly learns to visually separate words without the textual content. This makes the resulting model largely language-independent. On an Android emulator (API 34), the word detection model took approximately one second to process a page containing around 200 words.

Character Recognition and Localization. Each word crop is processed by a YOLO26n-seg instance segmentation model for joint character recognition and localization. We fine-tune this model on **CharLoc** (Yasin and Zesch, 2026), which contains 37,010 handwritten English word images with more than 219,000 characters annotated with character labels, bounding boxes, and precise character contours. For each detected character, the model predicts its class and a contour representing its spatial extent. The predicted characters are ordered from left to right to reconstruct the recognized word, while the contours retain their locations in the handwritten image. On the Android emulator, the character model required approximately 175 ms per word.

Feedback Generation. The recognized transcription is passed to a feedback component, which generates feedback on the learner's production and identifies the corresponding target in the recognized text. The feedback target is represented by the character position or span to which the feedback refers.

Feedback Grounding. Once a character or character span has been identified as the target of the feedback, it must be mapped to its location in the handwritten image. This cannot be reliably inferred from its position in the transcription, since handwritten characters vary in width, spacing, and slant and may touch or overlap.

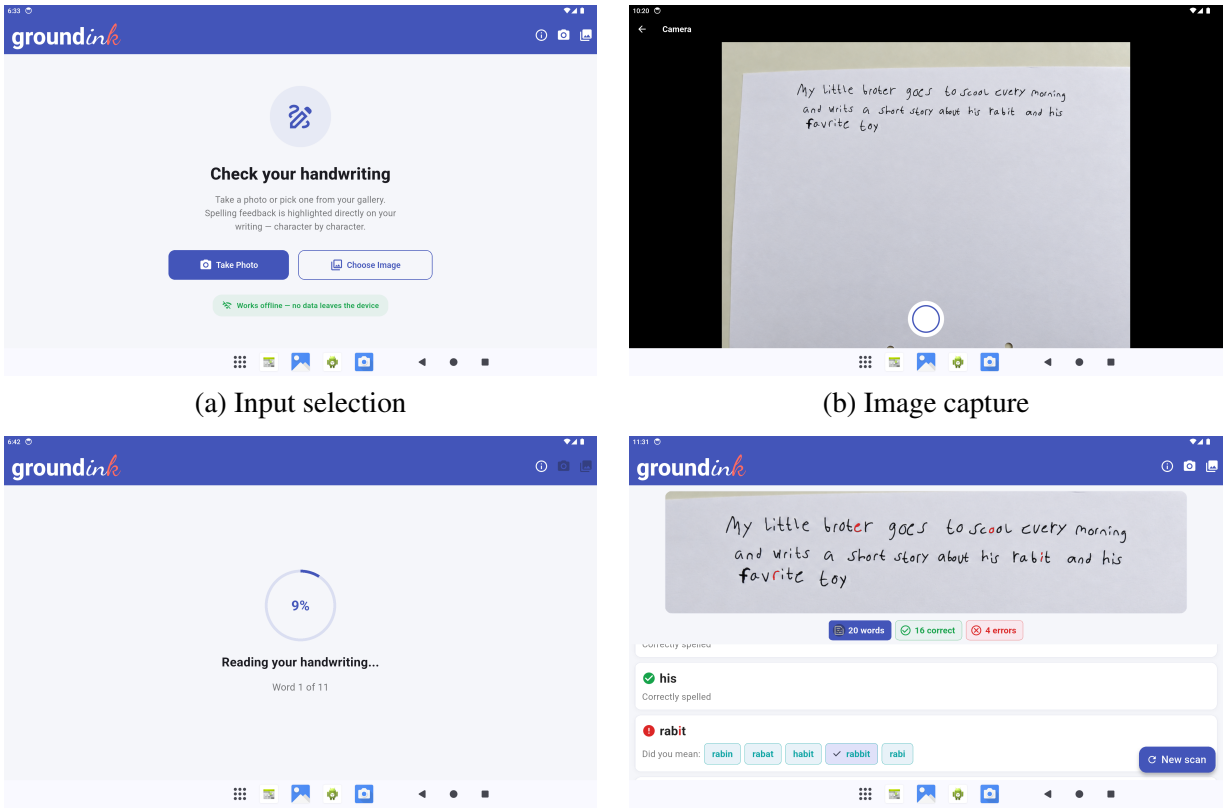
The identified character position is mapped to the corresponding predicted character contour. This contour is used as a mask to highlight the relevant character directly in the original handwritten image. Thus, the feedback is spatially grounded at character level rather than only at the level of the complete word.

3 Demo

To demonstrate character-level grounded feedback in a concrete application scenario, we use spelling feedback as an example. The feedback generation component computes the Levenshtein distance between each recognized word and the entries in an English word list bundled with the application and identifies several of the closest matches as correction candidates.[3] Among these candidates, the system automatically selects the closest match. If this does not correspond to the word intended by the student, they can choose from the remaining plausible alternatives. The selected word is then compared with the recognized word to identify and highlight the affected character in the orig-

[2] https://docs.ultralytics.com/models/yolo26

[3] Note that we are using English here for demonstration purposes. The application itself is not limited to English and has been successfully tested with various other languages that use the Latin alphabet. Highlighting is currently limited to the character set in CharLoc, i.e. the standard English alphabet without for example German umlauts (ä, ö, ü).



(a) Input selection (b) Image capture

(c) Local processing (d) Grounded feedback

Figure 3: Workflow of the mobile handwriting feedback system. The learner captures or selects an image of handwritten text, which is processed locally on the device before character-level grounded feedback and correction suggestions are presented.

inal handwriting. When a different candidate is selected, the highlighting is dynamically updated, allowing different error locations to be visualized in cases of ambiguous spelling errors.

Figure 3 shows how this character-level grounding is realized in the mobile demo. The learner first selects an image source, either by capturing a new image or choosing one from the gallery (a). The handwritten text is then captured through the camera interface (b), after which the application processes the image locally and displays the recognition progress (c). Finally, the relevant handwriting region is cropped and enlarged, and the interface presents it together with the recognized words, correction suggestions, and character-level feedback highlights (d). In this example, selecting "rabbit" as the correction for the recognized word "rabit" highlights the corresponding character in the original handwriting.

Although the current demo uses spelling errors as a concrete example, the grounding approach can also be widened to support other linguistic error types whenever the relevant feedback target

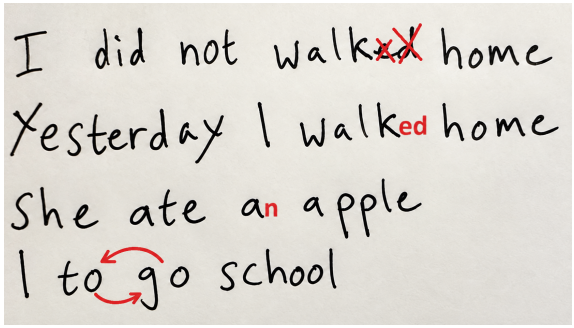


Figure 4: Illustrative visualizations of character- and word-level feedback beyond spelling.

can be mapped to the handwriting. Figure 4 illustrates such extensions: morphological or grammatical errors could be visualized by highlighting or crossing out superfluous characters, or by inserting missing characters or inflectional endings in a visually distinct typeset font. Word-order errors could be represented using arrows between word-level bounding boxes. Thus, the proposed approach can support a broader range of corrective feedback beyond spelling.

In future work, we plan to improve the robustness of the system and evaluate it in realistic classroom settings. Another promising direction is the integration of augmented reality.

4 Related Work

Previous systems for feedback on handwritten input either present the feedback separately from the handwriting or spatially ground it at the word level. We are not aware of any other research prototype that grounds feedback at character-level granularity in the original handwriting. Karlsson and Åkerman (2022) present Orthographer, which processes camera-captured handwriting and visually marks affected words together with correction suggestions. Bailey et al. (2020) developed a HoloLens-based application that analyzes handwritten words and provides immediate visual feedback in the user's field of view. Yasin and Zesch (2025) use a document camera to capture handwritten documents and spatially ground generated feedback using word-level bounding boxes.

Handwriting-based feedback has also been explored in language-learning applications. Pocket English Master (Imasha et al., 2023) combines real-time handwritten-text scanning with grammar and spelling correction for English learners. More recently, Paris et al. (2026) introduced FRANKI, a shared classroom device that scans handwritten student texts and generates AI-based feedback, which is presented either through a projection or as a printed response. Across these systems, feedback is either presented separately from the handwritten input or spatially associated with complete words. One factor contributing to this gap has been the limited availability of character-level annotations that support the localization of both block and cursive handwriting. Yasin and Zesch (2026) address this limitation with CharLoc, enabling the joint recognition and localization of individual handwritten characters.

Limitations

The demo is a proof of concept and has not yet been evaluated in user studies.

The feedback component has not been independently evaluated, as we focus in the demo on the localized grounding of feedback in handwriting. In other words, we consider for the purpose of this demo the generated feedback to be correct and focus on the visualization.

AI Disclosure

☐ Not applicable (no AI tools used)
☐ Assistive use
☒ Editing
☐ Generation of minor elements
☐ Translation
☒ Compression/expansion
☐ Generation from prompt

Details: We used OpenAI's GPT-5.6 Luna model through the ChatGPT app to improve grammar, sentence structure, clarity, and academic writing style (Editing), as well as to shorten selected author-written passages (Compression/expansion). All AI-assisted modifications were reviewed and verified by the authors.

References

Christine Bailey, Jacqueline Deprey, Abdulfatai Fakoya, John Nolan, Ivan Quiles-Rodriguez, Cameron Soderberg, Jeong-Yoon Wu, and Richard Yu. 2020. Exploration of augmented reality as an assistive device for students with dyslexia. Gemstone Team ART.

John Bitchener and Ute Knoch. 2010. Raising the linguistic accuracy level of advanced L2 writers with written corrective feedback. *Journal of Second Language Writing*, 19(4):207–217.

Jean Chandler. 2003. The efficacy of various kinds of error feedback for improvement in the accuracy and fluency of L2 student writing. *Journal of Second Language Writing*, 12(3):267–296.

Rod Ellis. 2009. A typology of written corrective feedback types. *ELT Journal*, 63(2):97–107.

Dana Ferris and Barrie Roberts. 2001. Error feedback in L2 writing classes: How explicit does it need to be? *Journal of Second Language Writing*, 10(3):161–184.

Ashini Imasha, Kavindu Wimalaweera, Manohari Maddumage, Dilain Gunasekara, Kalpani Manathunga, and Devanshi Ganegoda. 2023. *Pocket English Master – Language Learning with Reinforcement Learning, Augmented Reality and Artificial Intelligence*, pages 74–85.

EunYoung Kang and Zhaohong Han. 2015. The Efficacy of Written Corrective Feedback in Improving L2 Written Accuracy: A Meta-Analysis. *The Modern Language Journal*, 99(1):1–18.

Viktor Karlsson and Aston Åkerman. 2022. Real-time detection of spelling mistakes in handwritten notes. Master's Thesis.

John F. Lalande. 1982. Reducing composition errors: An experiment. *The Modern Language Journal*, 66(2):140–149.

Urs-Viktor Marti and H. Bunke. 2002. The iam-database: An english sentence database for offline handwriting recognition. In *International Journal on Document Analysis and Recognition*, volume 5, pages 39–46.

Carla Paris, Valeria Herskovic, and Daniel Villaseñor. 2026. FRANKI: AI-Powered Feedback for Handwritten Assignments. In *Proceedings of the 25th Annual ACM Interaction Design and Children Conference*, IDC '26, page 1299–1302, New York, NY, USA. Association for Computing Machinery.

Catherine G. Van Beuningen, Nivja H. De Jong, and Folkert Kuiken. 2012. Evidence on the Effectiveness of Comprehensive Error Correction in Second Language Writing. *Language Learning*, 62(1):1–41.

Said Yasin and Torsten Zesch. 2025. Show, Don't Tell – Providing Visual Error Feedback for Handwritten Documents. In Zygmunt Vetulani, Patrick Paroubek, and Irakli Kardava, editors, *Human Language Technology as a Challenge for Computer Science and Linguistics – 2025*, pages 99–103. Adam Mickiewicz University Press, Poznań.

Said Yasin and Torsten Zesch. 2026. Accurate localization of characters in handwritten words. *International Journal on Document Analysis and Recognition (IJDAR)*.