Declan Urbaniak-Dornstauder
DVU072 / 11284744
CMPT 481

Project Proposal

Interaction problem

The problem addressed by this project relates to the potential limitations of conventional keyboard-based input devices when it comes to entering digital text. Whether it be due to physical constraints or a lack of technical proficiency, these input methods may be impractical for certain user groups. The objective of this project is to develop and explore an alternative, user-friendly method for text input that accommodates a diverse range of users.

Interaction technique

The proposed interaction technique aims to recreate the simplicity of pen-and-paper handwriting in a digital environment. Users can directly write on the screen of a device using a stylus or their fingers, and this input is then translated into digital text. By offering a seamless and intuitive method for users to input text, this technique strives to combine the familiarity of traditional writing with the convenience of digital platforms.

Tools and resources

Hardware

The implementation of this interaction technique should simulate an external input device used peripherally with another device, similar to how a keyboard is used with a desktop computer. This can be accomplished using a mobile device with touchscreen capabilities, such as a phone or tablet. The device will run software to enable capturing handwritten text on-screen, translating the written text into digital text, and transmitting the digital text externally to another device. For the sake of compatibility and development familiarity, this software will be developed for Android devices only.

Languages, libraries, and technologies

As mentioned previously, the software enabling the interaction technique will take form of an Android application. It will be developed primarily in Kotlin, and will feature a simple user interface built with the Jetpack Compose toolkit. The back-end will be implemented in multiple layers - Tensorflow Lite will be used to develop the machine learning model required for handwriting recognition, and Kotlin will be used for general back-end functionality.

Datasets

As the core functionality of the software lies in the ability to recognize handwritten text, the learning model must be trained on a diverse set of handwriting samples. The exact method of sampling is yet to determined, though the most important aspects top take into consideration are the diversity of the data and the overall accuracy of the model. Only characters and symbols available from the US keyboard layout will be considered.

Outline of activities

1. *Research:* Further research is required to ensure appropriate technologies, libraries, etc. are used. Due to its importance, a special emphasis should be placed on researching the implementation of the machine learning model.

2. *Architecture design:* Establish the application design pattern (likely MVVM), and determine the appropriate structure to organize modules of the application.

3. *Handwriting capture modules:* Implement both the front-end and back-end modules that enable writing on-screen and recording of written input. Implement additional UI features as necessary.

4. *Learning model dataset:* Create a dataset of handwriting samples to train the learning model. Special consideration should be taken to ensure the dataset is as diverse as possible, while still maintaining a high accuracy.

5. *Handwriting recognition modules:* Develop the back-end module implementing the handwriting recognition learning model.

6. *Data output module:* Develop the back-end module that enables the transmission of translated text to external devices (low priority).

Milestone 1 (March 8)

Complete activities 1, 2, and 3. Make significant progress towards activities 4 and 5.

Milestone 2 (March 22)

Complete activities 4, 5, and (if possible given time constraints) 6.

Risks

1. *Learning model development experience:* This will be my first time developing a machine learning model. Due to its critical role in the project, my inexperience is a potential risk. From what I understand, text recognition is considered to be an entry level problem in the world of machine learning. Given this, I should have access to plenty of resources to support me in development of my model. Worst case-scenario, I have the option of resorting to OCR frameworks like Tesseract to for handwriting recognition in this project.

2. *Project scope:* Considering I am working independently, the scope of this project *may* not be feasible to complete alone. I believe this is dependent on how much time it takes me to complete activities 3 and 4, which I have already discussed in the previous point. Worst case-scenario, I believe it is realistic to end up with a ‘minimum viable product’ of sorts that is not quite as polished as it could be, but is still satisfies the and meets the base functionality of this interaction technique.

Evaluation

I have no expectations that the proposed interaction technique will ever be as efficient as a traditional keyboard for text input, so I will not be doing any sort of comparative analysis.
Instead, the evaluation will focus primarily on the functionality and correctness of the implementation of the interaction technique. For example, the correctness in recognizing written text will be considered. It is also important to consider the general usability of the interaction technique, such as making sure it is not inconvenient or frustrating to use. This will likely be self reported, though if possible, it would be ideal to have a sample of users to test the interaction technique and provide feedback of their experience.