Master's Thesis

Automated Feedback Generation for Second Language Handwriting Exercises

Anne Koch

February 2024 – August 2024

Matriculation Id: 3217159
Course of Study: Practical Computer Science

Reviewer:
Professor Dr.-Ing. Torsten Zesch
Professor Dr. Christian Beecks

FernUniversität in Hagen
Center of Advanced Technology for Assisted Learning and Predictive Analytics
Research Professorship Computational Linguistics
58097 Hagen

Abstract

Learning handwriting is a long and tedious process and adults learning a second script often do not have as regular access to human teacher feedback as children in primary school do. For this reason, this thesis explores an approach to automatically generate feedback for handwriting exercises.

This thesis studies feedback generation for single German letters provided as offline pixel images. In contrast to existing feedback approaches that provide feedback for tracing exercises or in comparison with a strictly defined standard letter, this approach explores a method to accept a greater variety of freehand letters and only give feedback where legibility of the letter is compromised.

The studied feedback algorithm is a three-step process. First, the letter is assessed for its quality. If feedback is required, the next step uses a Variational Autoencoder to transform the letter into a more typical letter form and in the last step a segmentation model is used for visually marking the letter features that were written well as well as those that still need improvement.

The analysis shows that the implemented proof-of-concept solution performs better than the baseline solution but also shows the limitations of the current implementation. Recommendations are proposed for future studies to collect actual learners' handwriting samples and study feedback generation based on online data.

In conclusion, this thesis proposes a new approach to generating visual handwriting feedback.

d

Contents

Glossary

cursive a type of handwriting that fluently connects all letters, also called joined script. 36

grapheme the smallest units in a writing system are its graphemes, or written symbols. (Bassetti and Cook 2005). 10, 11

manuscript a type of handwriting where each letter is written separately, also called block or print. 36

writing system a type of handwriting where each letter is written separately, also called block or print. 10

Acronyms

CNN Convolutional Neural Network. 41

L1 First language. 10, 36

L2 Second language. 49

VAE Variational Autoencoder. 37, 39

Chapter 1

Introduction

A large number of immigrants and refugees arriving in Germany come from countries with different languages and scripts used than in Germany. So, while they are literate in their first language, they need to learn the new script. However, while some literacy courses do exist, most beginners' language courses assume literacy and do not provide for literacy training (Piccinin and Dal Maso 2021).

This is the motivation to develop an application that supports learning to write and thus helping to fill this gap of not enough literacy training. There are apps in which letters can be *drawn* on screen via finger movements. This on-screen writing is already difficult for those who know how to write. Learning completely new letter forms with this medium seems almost impossible. So the envisioned application is to contain a component with which handwritten writing exercises performed on paper can be recorded as a photo and evaluated in the app.

The aim of this master's thesis is to develop approaches for generating automated feedback on the exercises. The intention is to not only accept letters as correct that correspond to a somewhat narrow definition of a model letter but follow a more open approach to allow for writers' variability and writing style to show itself and accept letters that can reasonably be read as the intended letter.

- first examples

1.1 Research question

How can feedback for single letter handwriting exercises in L2 language tuition be generated automatically?

1.2 Hypotheses

- The results of a handwriting recognition model can be used to assess the quality of a letter.
- A letter can be transformed in the latent space of a Variational Autoencoder to create a similar but correct version of the same letter.
- An object detection model can be used to mark letter features in a letter.

- Eingrenzung bzw. Abgrenzung von anderen Arbeiten
- Vorgehensweise, Methode oder Aufbau der Argumentation
- Ausblick auf Ergebnisse

Chapter 2

Didactic background

Danna and Velay (2015) describe handwriting as a complex perceptual-motor skill encompassing a blend of visual-motor coordination abilities, motor planning, cognitive, and perceptual skills, as well as tactile and kinaesthetic sensitivities that require thousands of hours of practice to master.

Depending on the group of people (see fig. 2.1) different tuition is required as they have different prerequisites and needs.

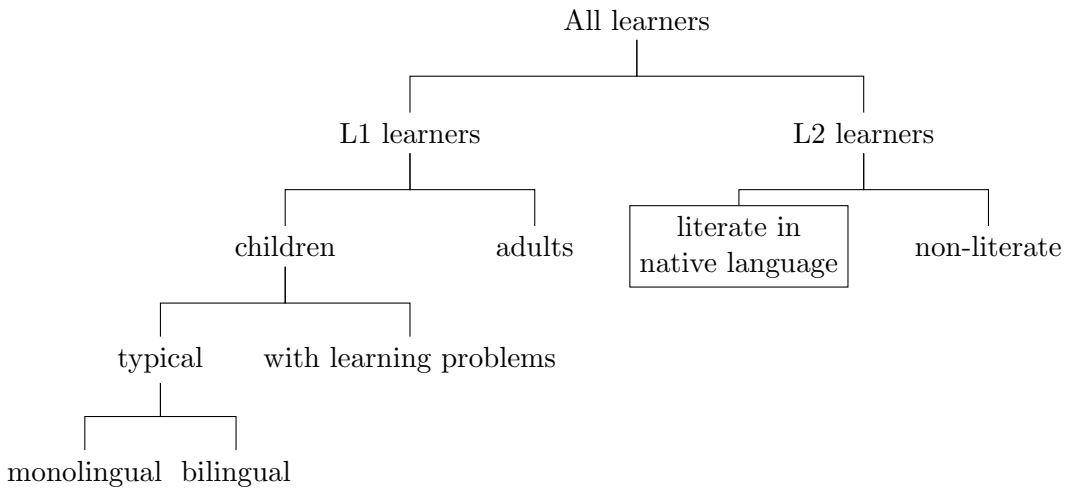


Figure 2.1: Groups of learners

2.1 First language handwriting acquisition

Most research on writing acquisition is done on children learning to write their first language as they are the by far biggest group of handwriting learners (Piccinin and Dal Maso 2021). Various models of writing acquisition have been proposed for example by Frith (1985), Ehri (1995), Feldgus and Cardonick (2002) and Berninger et al. (2006) for children growing up in literate communities and undergoing kindergarten and primary school education in their native language.

The models include phases of

1. Scribbling in early childhood: the child imitates writing
2. Emergent writing: the child combines simple shapes to form imaginary letters
3. Transitional writing: the child learns some letters and can write important words like their own name
4. Fluent writing: the child knows all letters and can write them

While the first three phases can occur naturally in a literate environment, the last phase is the transition to formal education. So when children start preschool or school they already have a background of oral language knowledge and first writing skills to build upon. However, while these stages describe the development of writing skills before and when formal education starts, they do not differentiate between different skill levels during the formal writing tuition phase.

A special case of children are bilingual children who receive formal education in both languages and scripts in primary school. Berkemeier (1997) describes positive interdependencies of starting learning both scripts within a short period of time and a high awareness of letter forms and differences between both scripts among this group of children.

Children with learning problems and adult First language (L1) learners Even though considerable research is done on children with learning problems, especially from a psychological and educational point of view, this group of learners as well as illiterate adult L1 learners are beyond the scope of this work.

2.2 Learning a second script

The research on handwriting acquisition focuses predominantly on children learning to write their native language. Research on other groups of learners is rarer. And while there is research on L2 writing acquisition in general it focusses more on spelling, orthography and text production than handwriting.

Describing the general second language literacy acquisition process, Marschke (2022) states that persons who already know a first language script already know how language is represented in a writing system (technical literacy) and already have experience in using writing (functional literacy). Therefore, they can use their existing knowledge when learning the second language and script. As they already learned another writing system, they already have the fine-motor-skills for writing and have an inventory of written symbols and grapheme-phoneme correspondences from their first language that they can put in relation to the new writing system.

Because of these existing knowledge and skills, in Germany the Federal Office for Migration and Refugees (BAMF) has been offering special integration courses for second script learners since 2018. According to BAMF (2018), this course starts with an intensive introduction to the Latin writing system with a duration of 300 lesson units

	Russian	Tigrinya	Arabic	German
writing system	alphabetic	Abugida	Abjad	alphabetic
size	32 letters	248 symbols	28 letters	26 letters
lowercase and uppercase letters	yes	no	no	yes
text writing direction	left to right	left to right	right to left	left to right
letter writing direction	start variable, anti-clockwise turns	strictly from top left to bottom right	start variable, clockwise turns	start variable, anti-clockwise turns
letter position	above and below line	strictly above line	in Ruq'a handwriting above line	above and below line
pen lifts within letters	few lifts in some letters	repeated lifts	repeated lifts	few lifts in some letters

Table 2.1: Comparison of script characteristics of Russian, Tigrinya, Arabic and German (Berkemeier (1997), Calle (2022) and Leupolz-Oebel (2020))

which precedes the actual language course. This approach is possible for second script learners because they are already literate in a non-Latin writing system and are familiar with writing as a system, which is why - unlike primary and functional illiterates - they are able to acquire the Latin script within a shorter period of time.

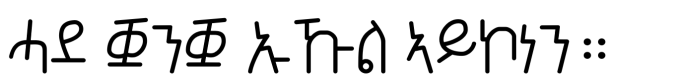
Taking the existing knowledge into account, Minuz et al. (2022) and Marschke (2022) advocate for *contrastive* language tuition that respect the migrants' first languages and if possible use them to build new skills in the second language in comparison with them. Depending on the similarities and differences between the languages and scripts involved, there might be different transfer effects.

To exemplify some of the issues that can occur when learning to write a new script, the following paragraphs list some languages that are currently of relevance in Germany with regard to migrants' language learning as well as some of the transfer effects resulting from them. For an overview of some characteristics of the listed scripts see table 2.1 and handwriting examples of these scripts 2.2.

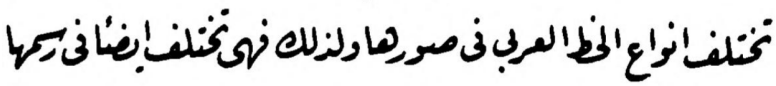
Russian Similar to Latin based scripts, the Cyrillic Russian script differentiates between upper and lower case letters and is written from left to right. While Latin based and Cyrillic scripts are structurally similar, Berkemeier (1997) lists the types of grapheme phoneme correspondences or differences that occur between letters of the two scripts as follows:

- matching graphemes with matching phoneme correspondence
- matching graphemes with different phoneme correspondence

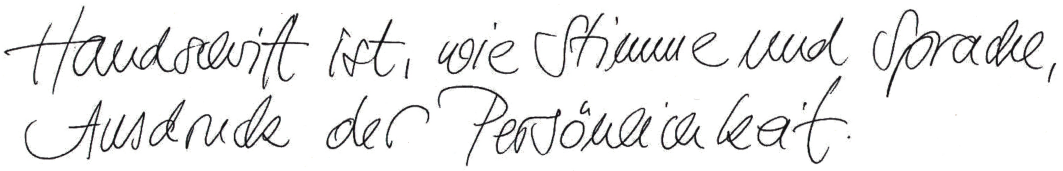
(a) Russian, written with a handwriting font



(b) Tigrinya, written with a handwriting font



(c) Arabic (Al-Khattat 1986)



(d) German (Pieper 2014)

Figure 2.2: Handwriting samples in the four example scripts

- different graphemes with matching phoneme correspondence and
- graphemes that only exist in one of the two alphabets.

Addressing these correspondences and differences when teaching the new script can help students gain awareness about their writing.

Tigrinya Calle (2022) studied German text samples of writers whose first script was Tigrinya and found transfer problems that can be explained with the differences between the two scripts like all letters staying above the line without descenders as would be required in German but is not the case in Tigrinya, making pen lifts where the German letter usually would be written in one motion and using capital letters within words. All learners had some introductory tuition in the German script. But such differences would need to be addressed more and learners should be made aware of them as often as possible so that differences can be internalised and automatically performed correctly.

Arabic Leupolz-Oebel (2020) studied the handwriting of students with first written language Arabic who migrated to Germany where they had to learn a new writing system. She found four areas where those students had transfer problems from their first learned writing system, namely:

- transfers caused by different writing direction

- transfers of known letter forms to other letters
- transfers of familiar pen turning or movement directions (as seen in letters *o*, *a* and *g* in figure 2.3)
- transfers of the familiar writing position for letters (as seen in the letters with descenders in figure 2.3)

Fabiani et al. (2023) studied bilinguistic people writing both in French and Arabic and found in addition to the points above, that Arabic writing requires more pen-lifts compared to French.

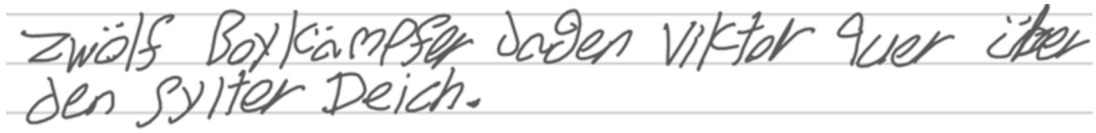


Figure 2.3: Transfer problems in a handwriting sample of a person with first written language Arabic (image from Leupolz-Oebel (2020))

Non-literate learners The other group of adult L2 learners are learners who are non-literate in their first language.

> Like all other adults, non-literate learners enter their second language classes with a wealth of life experiences and life skills, knowledge of the world, fluent communication skills in one or more languages and with well-developed skills to process meaningful information. In other words, in most domains of life and communication those who are non-literate share the skills that literate language learners employ, and clearly differ from young pre-school children. But research on non-literate second language learners that has been conducted during the last decades also clearly shows that some (cognitive) literacy-based skills that are usually presupposed in second language teaching for literate learners cannot be expected from them. (Minuz et al. 2022)

Even though a relevant part of migrants are amongst this group of people, they will not be considered in this work, as they require far more intensive human tuition than can be provided by a mobile phone app.

2.3 Writing exercises

There is little research on the type of writing exercises. Most research that is done on writing exercises focuses on small children like Florence and Nathalie (2021) assessing the effect of modifying the pen-traces of five-year-old-children or Chartrel and Vinter

(2008) studying the spatial and temporal constraints on handwriting movements in children of 5 to 7 years of age.

While there is little research on what writing exercises are used for adults and how effective they are, some information about existing writing exercises can be gained from textbooks for German literacy education. Two examples can be seen in figures 2.4 and 2.5. The books are intended both for illiterate persons and second script learners. As they are also intended for illiterate people who never learned to write before, they include exercises for basic pattern forming with a pen and tracing larger letters with a finger. The actual letter writing exercises consist mostly of copying and tracing sample letters. Surprisingly a print font is used in these examples and not a handwriting font, even though a handwriting font is used in other parts and exercises of the books.

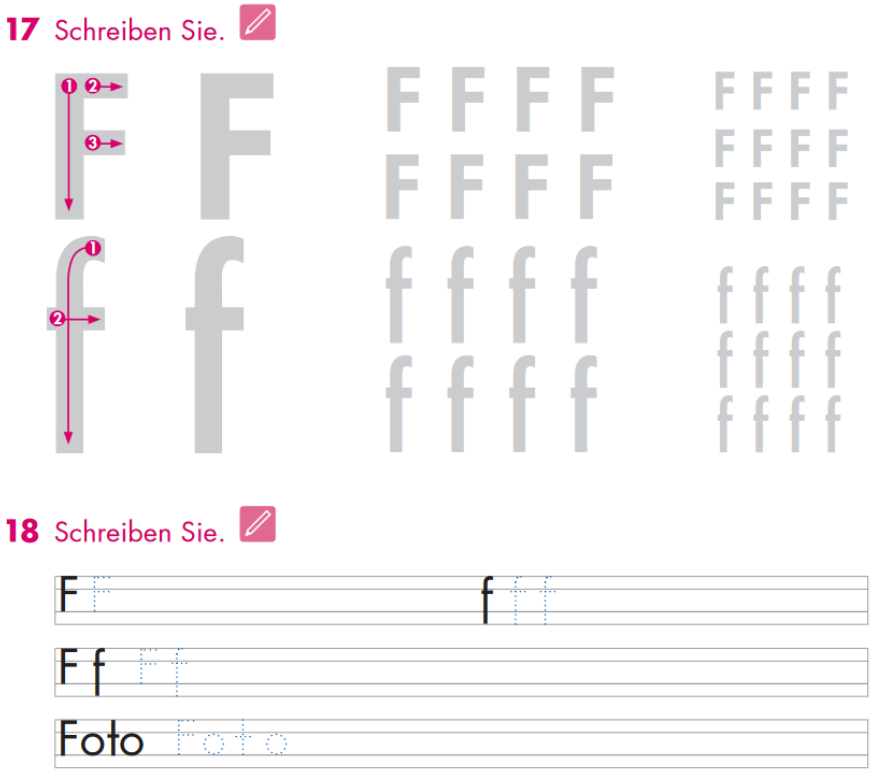


Figure 2.4: Handwriting exercises for the letter *F* from Böttinger (2011)

2.4 Assessment of writing exercises

If feedback is to be given on writing exercises then they need to be graded first. The discussion about how to grade or assess handwriting is old, see reviews by Graham (1986) and Rosenblum et al. (2003) who generally see legibility or readability as the basis for assessing handwriting. The latter describe that as a consensus about which

Figure 2.5: Handwriting exercises for the letter *E* from Böttinger and Wiebel (2018)

criteria constitute the critical components of handwriting readability, most researchers accept the criteria of size (height, width); slant; spacing (spaces between letters/words); the degree of line-straightness; shape (letter form and shape); and the general merit of the writing. Lichtsteiner describes legibility and fluency as the main criteria for grading handwriting.

grading systems

2.5 Types of feedback

Limbu et al. (2019),

Loup-Escande et al. (2017b)

Patchan and Puranik (2016)

Feedback is essential for learning any new activity. Danna and Velay (2015) divide feedback for handwriting exercises into basic feedback and supplementary feedback and while they discuss studies done on children, some of the effects might be the same for adults learning a second script.

Basic feedback (also called primary or intrinsic feedback) is described as sensory feedback that is used naturally by a writer to control their handwriting, one being visual

feedback that gives the writer spatial information about how and where they write, and the other being proprioceptive feedback, which is created by muscles, tendons and joint receptors creates information about the location and movements of arm, hand and fingers and allows for controlling movement execution. When students learn a new script they use visual control extensively at the beginning, but later, their body learns the motor program for the script and a mix of visual and proprioceptive control is used.

Supplementary feedback (also called extrinsic feedback) is described as additional sensory information provided to a writer in complement to natural sensory feedback. Supplementary feedback can be provided in real-time, i.e. during execution of a writing task, or after the performance of the writing task. Real-time feedback has advantages as it is very directly connected to what the person is doing at that moment. In contrast to that is the situation that often occurs in language tuition that a student hands in some homework and receives the corrected homework back some days later. This is far from optimal as the feedback is not received when it would be needed, i.e. during execution of the task, but results from the fact that there are not enough teachers available. This situation could be alleviated with feedback provided by a smartphone app.

According to Danna and Velay (2015), the supplementary feedback can be provided in the same modes as basic feedback: proprioceptive or visual or in a different mode as auditory feedback. Proprioceptive feedback provided with sensors and vibrating devices requires complex and costly devices and is therefore not taken into account here for the purpose of use with a smartphone app.

When providing real-time feedback, it is preferable not to provide the feedback in the same sensory mode as is used in the activity to be learned as this would risk overloading the cognitive capacities and leading to degradation of the movement. As writing requires visual control when learning, it would therefore not be helpful to use real-time visual feedback. When visual feedback is displayed after the writing movement, however, it can have its place, both for informing similar to basic feedback about spatial features like the correctness of the letters, their position on the paper and position relative to each other or giving supplementary feedback about the velocity and smoothness of the movement.

The third option for supplementary feedback, auditory feedback is argued to have several advantages: it is less likely to overload the cognitive process than additional visual FB, it is suitable for informing about the movement kinematics and it is easy to apply.

2.6 Writing materials

When designing a handwriting app, a decision has to be made whether it should be planned in such a way that the student will write with a stylus or even just a finger on the phone screen or whether the phone is to be used to photograph or film the written product or writing process of the student. As both the writing surface and the writing

instrument influence writing, this decision might influence both the usability of the app and the learning outcome.

Gerth et al. (2016) did a comparison of handwriting with pen and paper and tablet and stylus between three groups of people: preschoolers, second-graders and adult persons. While it was found that writing with a stylus on a tablet leads to higher demands on motor control due to the lower friction of the writing surface, the adults could adapt easily to the new medium, because they could fine-tune their existing automated motor programs to the lower friction and higher writing velocity.

It remains unclear, however, whether this ease of adaptation would also apply to adults learning a new script as they would not have the automated motor programs for that new script even though they have a good general control of the writing instrument.

For that reason and because the intended users of the app are assumed to own a smartphone but not necessarily a larger tablet that would allow for writing more letters on the screen and increase the letter size, the app will be planned for use with pen and paper instead of the phone monitor.

Lineature Traditionally, in primary schools, German writing is taught on a line system with four guidelines per line and a spacing of five millimetres between the lines. The intended purpose of the lines is to delimit the writing and provide a visual clue for orientation. However, Lindsay (2011) studied the effect of unlined compared to lined paper on the legiblitiy and creativity of primary school childrens' handwriting and found that younger children (mean age 6 years 7 months) still need to learn basic writing skills like spacing and straightness of line and therefore lined paper might act as an interference rather than an aid. However, older children of about age 9 have already gained more familiarity with writing and benefit from lined paper.

For adult learners of writing who can already write in their first language, writing on a single line is the normal case and thus probably sufficient when learning a new script.

Chapter 3

Related work

Relevant for this work is the research in the areas of generating feedback for handwriting exercises as well as assessing handwriting. In general, all previous work focusses on tracing exercises or in some other way on comparing learners' writing to a standard model of writing. Many of the studies used some kind of expert grading or comparison with expert solutions as the basis for their assessment and feedback.

3.1 Comparable approaches for assessing handwriting and generating feedback

Some studies were done exploring approaches on how to generate visual, haptic and verbal feedback for handwriting exercises.

3.1.1 Visual feedback

Lili and Zhengwei (2021) studied ways to assess children's handwriting of digits using image processing methods. With the improved Hilditch algorithm for skeletonisation they removed unnecessary pixel points from the images and then went on to compare the resulting digit image with a given writing specification using a number of relevant features like strokes, starting point position, ending point position, stroke order, number of straight lines, number of circular lines, corner point position, overall position and pixel ratio. While this approach seems promising and useful for gaining information for feedback generation, some points remain unclear in the paper. First, they are collecting information from pixel images, but it is not clear how they could gain temporal information about start and end points and stroke order from skeletonised pixel images. Second, while they claim that they fed feedback about deficiencies that did not meet the specifications back to the user, it is not clear what kind of feedback they created and how they created it.

Some research has been done on training systems for Chinese calligraphy. Even though Chinese brush calligraphy is far more difficult to learn than writing letters with a pencil or pen as it requires very precise control of the brush movement, pressure and velocity, technical systems have been created for assisting calligraphy training. He et al. (2020) proposed a system using a camera and a projector with the camera registering the strokes made by the user, the system determining from a database of calligraphy

Attributes	Instructional design methods	Implementation
	Learning Task	
Alphabets Structure	Augmented Paths	Displayed on tablet for tracing or imitating, color of the stroke changes when the color stroke is out of bounds
	Procedural Information	
Force used to grip the pen	Haptic feedback	Vibrate vibrating motor when the grip is too tight or the angle is beyond the threshold
Pressure used to create the strokes	Object enrichment	Stroke thickness is directly proportional to the pressure, The stroke darkness/lightness is also directly proportional to the pressure
	Supportive Information	
Speed of writing, alphabet structure	Animation	animation depicting the speed and the path in which the alphabet was written
	Part task practice	
Over all performance	Summative feedback	Summative results produced by comparing with the expert recording

Table 3.1: Attributes tracked and feedback provided by the feedback system (Limbu et al. 2019)

characters the closest version of that character and projecting either the next stroke or the whole character over the user's solution so that the user has a tracing aid to follow with the brush.

Limbu et al. (2019) studied the mental load resulting from a multimodal feedback system for tracing tasks. The described system gives haptic, visual and/or auditory feedback about procedural information and speed, see table 3.1. Though this pilot study is limited in the number of participants (10 overall, 5 in the treatment group), number of letters written by each participant (3 characters of the Devanagari script, unknown to the participants prior to the study, each written 4 times) and short time for familiarising themselves with the system, the researchers concluded that the feedback provided by the system does not impose a high mental effort on the learners. The study did not, however, analyse the learning outcome resulting from the various feedback methods used.

Further studies on visual feedback for handwriting tracing exercises were done by Mitchell and Fairhurst (1992) and Loup-Escande et al. (2017a).

3.1.2 Haptic feedback

Morikawa et al. (2018) presented a system for providing vibrating feedback during Japanese brush calligraphy exercises. A leapmotion sensor is used to track the stu-

dent's brushwork and if the handwriting is found to be defective, the student's wrist is stimulated. A defect is determined if the brush leaves the reference trace provided or the brush is expected to overrun the proper end-position during a stroke.

The study described above by Limbu et al. (2019) also used haptic feedback to provide real-time information when the grip on the pen is too tight or the pen angle is beyond a threshold (see table 3.1).

3.1.3 Verbal feedback

Kulesh et al. (2001) presented an approach to evaluate the quality of handwritten letters based on a set of features that are used by human handwriting experts. They grade each letter in a sequence of known handwritten letters on a scale from 1 to 5 by estimating 4 criteria: shape, size, slant and position and then provide feedback to the user based on information from the expert system. The knowledge base of that expert system incorporates two major sets of rules. The first set contains rules that allow a grading of each letter based on the 4 high-level criteria described above. The second set includes rules that incorporate the knowledge of a teacher about the process of teaching handwriting. The authors provide an example of the feedback:

> The size and superposition are good (confidence level 9 out of 10), the slant is satisfactory (confidence level 7 out of 10), shape is poor (confidence level 8 out of 10), and therefore you need to work on the shape of letters in class A, i.e. letters that belong to the same cluster. Please take remedial lesson i. (Kulesh et al. 2001)

Thus, while grading the quality of the letter's shape, they do not give explicit feedback about what makes the shape of that letter poor and how to improve the shape and the student will need to study the letter on their own to find out.

3.2 Different perspectives on assessment of handwriting

Various works studied the quality of handwriting samples. One rather active area of research is the automatic detection of dysgraphia with machine learning methods using online writing samples, e.g. by Rosenblum et al. (2013), Asselborn et al. (2018) and Bublin et al. (2023). These studies focus on assessing writing but not on giving feedback as to how to improve the handwriting.

3.3 Existing apps

There exist some apps for literacy instruction for adults, but they don't include handwriting tuition. Therefore, they are only mentioned here to show the lack in this regard.

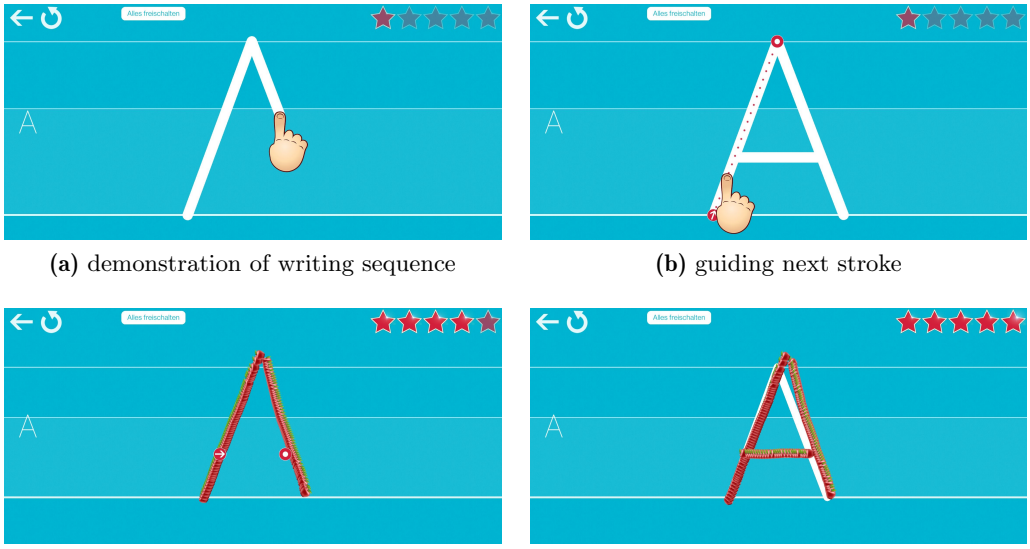
Diglin (Cucchiarini et al. 2013), the *digital instructor for literacy learning*, was a project developped by the European 'Lifelong Learning Programme' (LLP) for low-literate immigrant and refugee adults who need to learn to read and write for the first time in a language other than their mother tongue. The app is provided in four different languages (Dutch, English, German and Finnish) and uses Automatic Speech Recognition (ASR) to analyse the learner's read speech output and provide feedback.

Irmgard is an app for literacy education for German native speakers provided by the charitable KOPF, HAND + FUSS gGmbH company that runs projects for an inclusive society. While the app includes reading and writing lessons, the writing lessons focus on typing and spelling, but not handwriting.

There exist some apps that provide assistance with handwriting learning, but most seem to be restricted to tracing letters.

Writing Wizard is one such app by L'Escapadou that was used in a study by Patchan and Puranik (2016) assessing the learning outcomes of using paper and pencil, tablet computer and finger or tablet computer and stylus by kindergarten children. During exercises, the the app first demonstrates how to write the letter, then the user is tasked to trace the letter with their finger, if they stray too far from the model, the app stopps and shows an arrow at the start of the current stroke from where the student should restart. Upon successful completion of the letter, traced strokes are shown superimposed over the model letter (see figure 3.1). The app also provides an option to hide the model so that letters can be written freehand and an option to write complete words. This way, the app is providing intrinsic feedback by showing the comparison between the written trace and the standard letter and extrinsic feedback by refusing to trace too far away from the expected model letter strokes. The exercises seem to follow a sensible order of decreasing scaffolding assisting in the first steps of learning new letter shapes, however the rather childlike game elements added in between exercises might not appeal to adult learners.

Duolingo is a popular general language learning app for a variety of languages provided by the Duolingo company. It includes basic writing system tuition for reading and writing letters, however it seems to be created from a somewhat Western view-point of not including these reading and writing exercises for languages written in a Latin-based script. The existing writing exercises are done by finger-tracing the letter shape in the same modern screen font that is used for the rest of the app but not a handwriting font. The tracing exercise is broken down into the individual strokes of the letter. First the stroke is demonstrated and then the user is asked to trace the stroke. The stroke trace is shown as long as the user stays within a certain distance from the trace. If the user moves further away from the stroke the trace is discontinued but the user can continue again by starting from where the trace was discontinued (see figure 3.2). Overall it



(a) demonstration of writing sequence

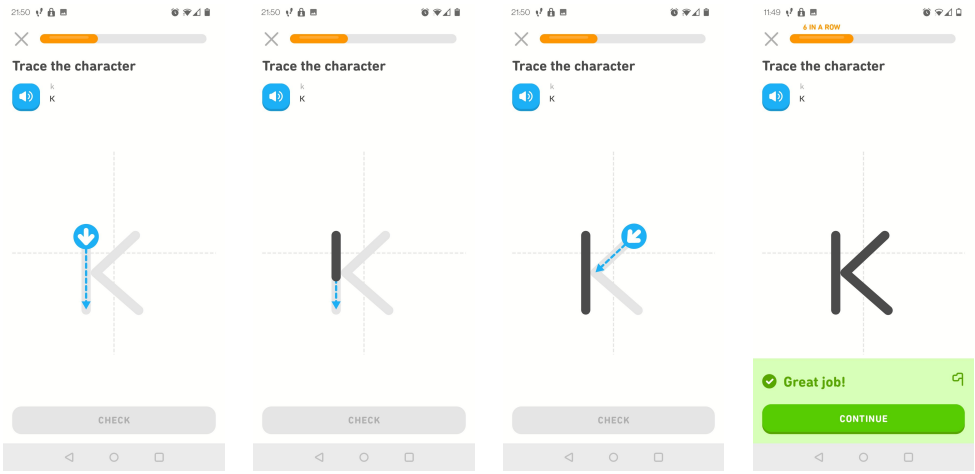
(b) guiding next stroke

(c) guiding in freehand mode

(d) showing result

Figure 3.1: Assisted writing process for uppercase letter *A* in Writing Wizard app

is doubtful whether such simple tracing exercises lead to any meaningful handwriting skills, but it might give a first impression and feeling for a new writing system.



(a) guiding stroke order

(b) tracing

(c) guiding next stroke

(d) enthusiastic verbal feedback

Figure 3.2: Tracing exercise for Russian lowercase letter *κ* [k] in Duolingo app

Chapter 4

Data collection

This thesis uses both existing datasets adapted to the task at hand as well as self-created and collected data.

4.1 Existing handwriting datasets

For handwriting tasks exist a variety of datasets that can be used for various research objects. According to Kotani et al. (2020), handwriting modelling methods either handle images, which capture writing appearance, or handle the underlying strokes collected via digital pens. Each may be online, where observation happens along with writing, or offline.

Online stroke data is richer in information than pixel data and may contain information about timing, velocity and pressure applied on the stylus while writing.

Existing present-day (i.e. not historical), Latin-script-based online handwriting datasets were presented by Liwicki and Bunke (2005) (IAM Online Database) labelled as text, line and word data, a newer dataset from Aksan et al. (2018) accumulated from IAM-OnDB and newly collected samples contains 406,956 handwritten characters and Kotani et al. (2020) presented the BRUSH dataset (BRown University Stylus Handwriting) of 27,649 handwritten digital strokes in the Latin alphabet provided at sentence level with segmentations for letters.

Offline datasets on the other hand are EMNIST, IAM, CVL and FD-LEX. The EMNIST database (Cohen et al. 2017) is an extension of the well known MNIST digit database to handwritten letters. It contains letter samples from writers with a fully-developed handwriting. The samples are everyday and possibly fast, but not careful writing (see figures 4.2 and 4.3). So, the letters are written confidently and exhibit the variety of real handwriting. However, they are not learners' letters and therefore do not exemplify problems in learner's writing. So, while not optimal for training a feedback algorithm for handwriting learning, it is still a big existing database that can be used for a proof of concept of the intended algorithm.

Similar to the online IAM database, there is an offline IAM database segmented at word level (Marti and Bunke 2002). The CVL database constitutes another offline database, segmented at word level (Kleber et al. 2013). The German learners dataset FD-LEX (Becker-Mrotzek and Grabowski 2018) provides several corpora with texts from German school children of varying ages. The texts are provided at text level with

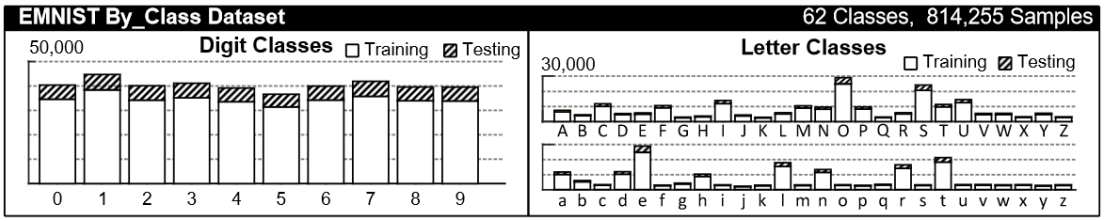
transcription at varying image quality and usually written on ruled paper.

As described in 2.6, the planned handwriting app is intended for use with pen and paper to allow learners to write on the familiar paper surface instead of the smooth and small phone surface. They would then use the phone cam to take images or videos of the writing. The writing could then either be processed as offline pixel images, online pixel data or online vector data to form the basis for the feedback algorithm. As the creation of pixel images is the easiest of these three options, this option is going to be explored in this thesis. Therefore, the EMNIST dataset will be used as data basis as it provides a large number of handwriting samples segmented and labelled at letter level.

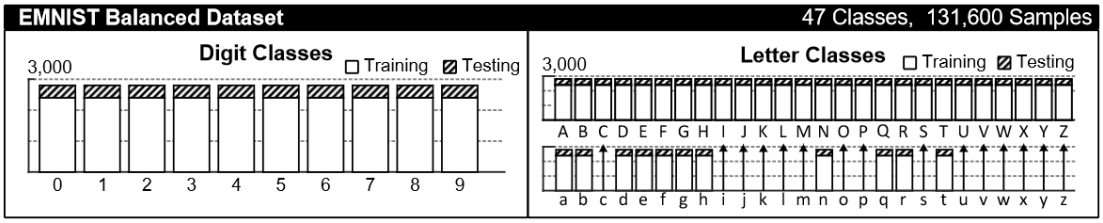
4.2 Preparation of EMNIST dataset

As the EMNIST dataset was presented for a different task, i.e. handwriting recognition, than is undertaken here, some preparation steps are necessary to adapt this dataset for this work.

The original dataset is rather imbalanced see the *EMNIST By_Class Dataset* diagram in figure 4.1a. Imbalanced datasets are a well known problem for machine learning tasks (Fernández et al. 2018). Known solutions to this problem are undersampling of majority classes and oversampling of minority classes. Here, the strategy of undersampling is used.



(a) By_Class dataset

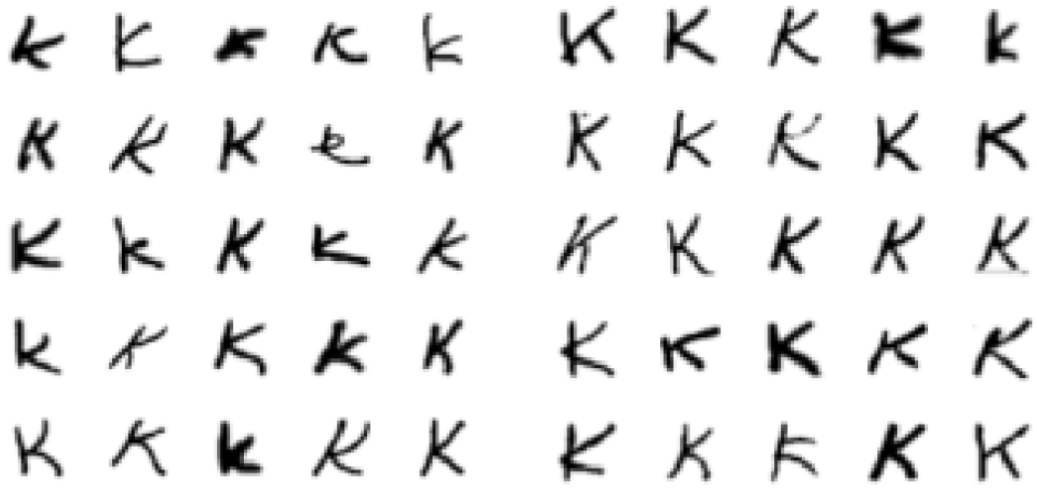


(b) Balanced dataset

Figure 4.1: Visual breakdown of EMNIST By_Class and Balanced datasets with classes and number of samples per class from Cohen et al. (2017)

The authors of the EMNIST dataset realised that there are some letters where the lowercase version is hardly distinguishable from the upper case letter if there is no information about the size of the letter, see figure 4.2. This is the case in this dataset

as all letters are scaled to the same size irrespective of whether it is a lowercase or uppercase letter. Therefore they decided to offer versions of the dataset where those classes are merged into one, see the *EMNIST Balanced Dataset* diagram in figure 4.1b.



(a) (b)

Figure 4.2: Example letters from classes with very similar letters; (a) lowercase k class and (b) uppercase K class

Another issue with the EMNIST data is that it does not contain any information about letter size and positioning relative to the base line, which is relevant for the task of assessing the correctness of a letter.

Due to the issues mentioned above, two datasets were created out of the existing datasets with the purpose of making up for the existing limitations:

the merged dataset is created out of the EMNIST balanced dataset by removing the digit classes. It contains 16 *merged* classes where uppercase and lowercase letters are combined into one class. Because of this, the dataset only comprises a total of 37 classes. Each letter is scaled to the image size and therefore does not include any information about size and position. The dataset contains 2800 images per class and training/test split of 80/20 leading to a total of 88800 images in the training set and 14800 images in the test set.

the padded dataset is created out of the EMNIST by_class dataset. The following steps were taken with this dataset:

- delete number classes (not relevant for the present task)
- partially balance dataset, by only keeping a maximum of 5000 instances per class; some classes have only about 2000 instances though.

- translate and rescale images to account for letter sizes and position (see figure 4.3)

This dataset contains all letters of the English alphabet, so it has 52 classes of uppercase and lowercase letters. The dataset contains at total of 261928 images and a training/test split of 80/20 leading to a total of 203523 images in the training set and 58405 images in the test set.

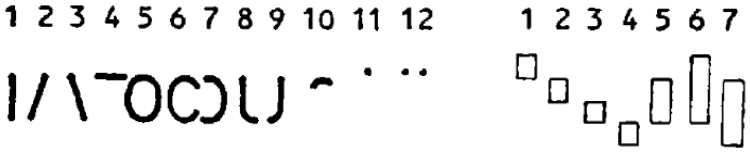
Figure 4.3: EMNIST letters padded with empty pixels depending on their type, i.e. lowercase or uppercase and with or without descenders and ascenders (gray padding colour only for visualisation, padding is done in white in the data)

4.3 Synthetic segmentation dataset of letter features

One objective of this thesis is to be able to give feedback about individual features of letters like ascenders or descenders, straight or curved lines, dots or corners to be able to reason about the letter form.

The idea of discussing letter forms by way of letter features has been discussed for printed scripts by Althaus (2011) who divides letter features into graphically distinctive features that differentiate between graphs (i.e. letters, digits, punctuation marks) like the _ differentiates between E and F and periphery graphical features that are used for additional and artistic design of the graphs.

The graphically distinctive features (see 4.4a) are further characterised by their location in the writing space (see 4.4b).



(a) graphically distinctive features **(b)** feature locations

Figure 4.4: Letter feature system based on Althaus (2011)

Fitjar et al. (2022) describes form, size and positioning of handwriting features with the goal of being able to analyse handwriting and to identify whether a feature has been produced with an acceptable degree of accuracy. For an example of the feature system see figure 4.5.

Letter	Feature code	Shape and orientation	Size	Position
k	k1	Straight Vertical	Length: longer than k2 and k3	To the left of k2 and k3
	k2	Straight Diagonal	Shorter than k1, Similar to or shorter than k3	Slant top to right Bottom meets in the lower half of k1
	k3	Straight Diagonal	Shorter than k1, similar to or longer than k2	Slant down to right Top meets in the lower half of k1, but not above k2 bottom

Figure 4.5: Letter features of the letter k as described by Fitjar et al. (2022)

Reinken (2023) presented a very detailed analysis of handwriting features based on studying a corpus of handwritten texts and categorising the forms of each letter found in these texts. For an example of the resulting feature system see figure 4.6.

While the approaches by Fitjar et al. (2022) and Reinken (2023) are very detailed and comprehensive, a similar level of detail for describing letter features is out of the

Prototypen / Grundform		Existenz		Form		Konstitution		Anzahl	Anteil	Anzahl Texte
		Kopf	Koda(s)	Kopf	Koda(s)	geschl.	Verb.			
k										
	k1	+	++			+	2,1	54	3,6 %	19
	k2	+	++			–	1,0	91	6,0 %	23
	k3	+	++				1,0	65	4,3 %	14
	k4	+	++			+	2,1	166	10,9 %	31
	k5	+	++				1,1	230	15,2 %	38
	k6	+	++				1,0	518	34,2 %	61
	k7	+	++				0,0	63	4,2 %	13
	k8	+	+–				1	246	16,2 %	51
	k99			Rest				83	5,5 %	37

Figure 4.6: Letter features of the letter *k* as described by Reinken (2023)

scope of this thesis. As this thesis attempts a proof of concept for using letter features for feedback, a simple letter dataset with a limited number of features was created as follows:

Step 1: As the process of creating features masks is very time consuming, only two digital handwriting fonts (*Grundschrift* and a font named *Amelies*, see figure 4.7) were chosen and images of each letter created with them.

Step 2: In each letter image the actual letter was isolated in a graphics editor and then separated into the individual letter features. Each feature was then saved into a separate image see figure 4.8.

(a) Grundschrift

(b) Amelies

Figure 4.7: Handwriting fonts used for letter segmentation

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

Figure 4.8: Example letter image *k* (a) and feature mask images *left ascender* (b) and *left corner* (c)

Step 3: Then, the synthetic images were created by taking a random letter (both the letter image and the corresponding mask images) and transforming them. As transform methods were used: slight down scaling and moving the position as well as rotation by a random angle from -5 to 10 degrees. Any angle beyond that would distort the letter in an uncharacteristic way. The same number of synthetic letters was created this way for each letter of the lowercase and uppercase alphabets.

Step 4: As the task studied here presupposes that the letter to be written is known in advance, this information can also be used for recognising the letter features. For this reason, separate synthetic segmentation training datasets were created for each letter. Each letter dataset was compiled from letter images and their respective masks of the target letter. The dataset was then created using the letter images as input data and the mask images as target data. The input shape is the size of the images and the output shape is the size of the images multiplied by the number of existing letter features. Pixels representing feature masks are coded as *1* and non-feature mask pixels as *0*. Each letter feature not existing in a given letter is represented by an array of zeros.

4.4 Preprocessing of handwriting samples

A small dataset of handwriting samples of L2 German learners with a non-Latin first script was collected. The questionnaire requested the learners to provide samples of all lowercase and uppercase letters of the German alphabet in individual boxes (questionnaire sample see fig. 4.9). The returned questionnaires were filled in by German learners with Ukrainian, Russian, Arabic and one with Azerbaijani first languages.

The returned questionnaires were processed by scanning, removing personal information if necessary, slight blurring to remove noise, whitening all pixels below a threshold and cutting out the letter boxes. The non-trivial task was how to cut the letters out of the boxes. Just cutting the whole box would result in very small letters as most participants wrote letters that did not fill the whole box. The assumption when designing the questionnaire had been that the participants would have a feeling for the baseline and write the letters on that imagined baseline depending on the required positioning of the letter in respect to that baseline. It would have been possible to put a baseline in the boxes but that would have required to remove them afterwards so the decision was taken to create the boxes without baselines. Because of this, the letters were cut out as follows: for each questionnaire all letter boxes were stacked and the first and last non-white pixel for each row and column determined. All letters were then cut out with these margins. The resulting letter data was scaled to 28×28 pixels and saved as tensors together with the letter labels see figure 4.10. As the current work is done on a training dataset without German specific characters *ä, ö, ü* and *ß*, those letters, while included in the questionnaire, where not included into the dataset.

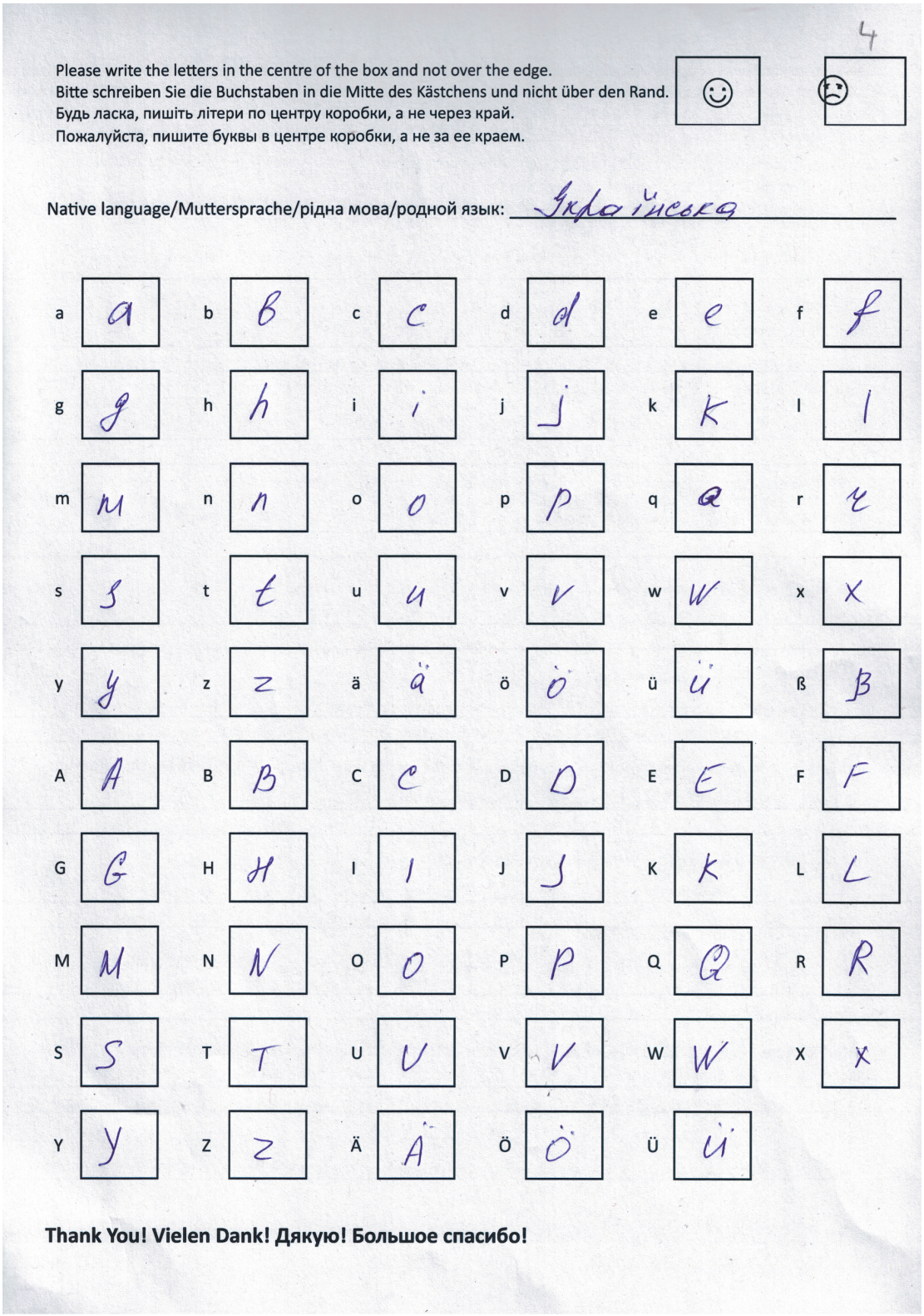


Figure 4.9: Completed questionnaire for learners' letter handwriting samples

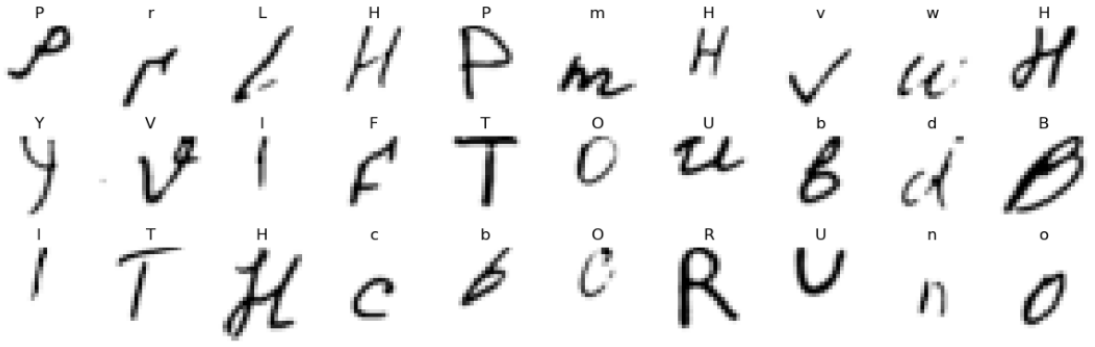


Figure 4.10: Letters extracted from questionnaires

Chapter 5

Implementation

Since it is not possible within the scope of this Master's thesis to develop a complete writing learning application, the investigations will be limited to feedback generation.

5.1 Preliminary assumptions

As described in section 4.1, the planned app is intended for use with pen and paper and the feedback will be generated based on pixel images taken with the phone cam. For this purpose, the following preliminary assumption is made: the writing exercise has already been photographed and the photo has been rectified, qualitatively optimised and converted into grayscale.

Since learners' writing is very variable, it can be assumed that image processing approaches will not be enough to make sufficient statements about the exercises. Therefore, a machine learning approach is used.

This thesis will focus on learning an alphabetic script based on the Latin alphabet like English or German.

As the feedback generation for complete texts and words is a complex task, this work will focus on individual letters only. When focusing on single letters only, the assessment of writing criteria as mentioned in section 2.4 that are measured at text level, like slant, spacing, size and position with regards to the baseline cannot be interpreted. Therefore, the focus here is on the letter shape.

The writing exercise studied here is to write a given letter. Therefore, it is always known which letter is the expected letter and the written letter can be compared to that expectation.

Second script learners who are already familiar with the process of writing itself, just not in the new script, bring their own history and experiences with writing into the new learning process. And while they need to learn the new script and its letter shapes, they already might have their own style or idiosyncrasies that is reflected when they write the new script and which is ok. Therefore, one objective of the algorithm developed here is to not base the grading and correction of letters on a narrow definition of some kind of standard letter, but to follow a more open approach of accepting letters based on their legibility and correcting only insofar required to improve legibility.

5.2 Baseline

The baseline algorithm for this thesis returns a standard version of the required letter if the letter written is categorised as erroneous. As there exist many handwriting standards, a decision has to be taken which standard to choose. This thesis focusses on the situation in Germany, so a handwriting standard used in Germany is used.

Cursive and manuscript scripts for teaching purposes In primary school handwriting lessons in Germany, four standard scripts are taught: the three cursive scripts *Lateinische Ausgangsschrift* (LA), *Schulausgangsschrift* (SAS) and *Vereinfachte Ausgangsschrift* (VA) as well as the manuscript script *Grundschrift* (see figure 5.1).

Figure 5.1: Overview of the three cursive scripts and the manuscript script taught at German primary schools as well as a possible development of that manuscript script with combinations (Pieper 2014)

For purposes of simplicity, the Grundschrift script is used as feedback for the baseline algorithm and overlayed over the letter to be corrected. Some examples of the baseline feedback are shown in figure 5.2. Letters graded as correct are shown in green boxes, letters graded as incorrect in red boxes. The title for each letter shows the expected letter before the colon and the letter recognised after the colon.

For later development stages of the app, it might be useful to also take into account the type of L1 script already learned by the learners. Berkemeier (1997) who studied bilingual children learning to write both of their languages one shortly after the other, argues that if their first learned script is a manuscript alphabetic script it might be easier for them to learn the other language with a manuscript script as well and vice versa. There is reason to believe that it would apply also to adults learning a new

Figure 5.2: Some examples of feedback with the baseline algorithm

language, that they find it easier to learn a new script of the same type. So, this would necessitate an option to choose between manuscript and cursive tuition for the app.

5.3 Detailed feedback algorithm

The detailed feedback algorithm is a three step algorithm as follows and as visualised in figure 5.3.

1. Use a handwriting text recognition model to detect whether the correct letter has been written. Evaluate the quality of the letter depending on the probabilities predicted for each letter.

2. If the quality of the letter is insufficient, use a Variational Autoencoder (VAE) to change the letter in the latent space into a more correct form.

3. Use a model trained with specific letter features to mark those feature segments in colour. Green for correctly written segments and blue for segments needing improvement.

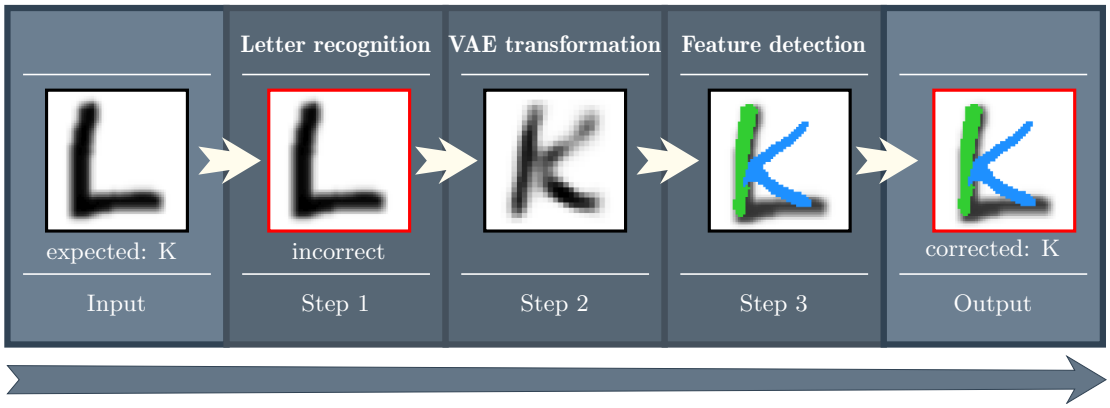


Figure 5.3: Pipeline

5.3.1 Letter quality assessment

The first step of the detailed feedback algorithm is to use the prediction values of a classification model to decide whether the letter was written correctly or not. Classification has been the subject of intensive research, specifically also on the MNIST and EMNIST datasets (Baldominos et al. 2019) which have also been used as benchmark datasets for computer vision tasks.

Here, however, the task is not a classification task, as the class of the expected letter is already known. But the idea is to use the prediction values as proxy for the quality of the letter. This is based on the assumption, that the prediction values are higher the clearer and more legible the letter. The purpose here is to find out whether this assumption holds true or whether another approach needs to be taken to assess the quality of the letter.

Architecture of the classification model used For the purpose of this thesis, a very simple neural net is used with the following architecture see listing 5.1:

```
==========================================================================
Layer (type (var_name))  Input Shape          Output Shape         Param #
==========================================================================
BasicCNN (BasicCNN)      [128, 1, 28, 28]     [128, 62]            --
 -Conv2d (conv1)         [128, 1, 28, 28]     [128, 24, 24, 24]    624
 -Conv2d (conv2)         [128, 24, 12, 12]    [128, 32, 10, 10]    6,944
 -Linear (fc1)           [128, 800]           [128, 256]           205,056
 -Linear (fc2)           [128, 256]           [128, 62]            15,934
==========================================================================
Total params: 228,558
Trainable params: 228,558
Non-trainable params: 0
Total mult-adds (M): 163.18
==========================================================================
```

```
Input size (MB): 0.40
Forward/backward pass size (MB): 17.76
Params size (MB): 0.91
Estimated Total Size (MB): 19.07
==========================================================================
```

Listing 5.1: CNN architecture for letter assessment for padded dataset

The model shown is the model for the padded dataset with 52 letter classes. Although the digit classes from the original EMNIST dataset were deleted in the padded dataset, the label numbers were kept the same as in EMNIST. Despite this leading to the model having 62 classes, this decision was taken to keep the label numbers consistent over all experiments. If there are no instances of the digit classes in the dataset, those additional classes will never show up in the outputs. The model will get bigger with additional classes, but as this is a proof-of-concept study, this was considered not as relevant. For the same reason, the model for the merged dataset has 47 classes, even though the dataset only contains images for 37 classes.

Assessment of letter quality with the Convolutional Neural Network (CNN) is done as follows: the model is used to predict the class of the image to be assessed. If the predicted class is not the class expected, the letter is graded as incorrect. If the predicted class is correct, but the probability of the prediction is lower than 0.80, the letter is still graded as incorrect. This probability value was found by comparing the letters with the predictions and deciding on a sensible, if somewhat arbitrary cut-off value.

5.3.2 Transformation in latent space

The second subtask is to transform an incorrect letter into a more correct form. The idea here is that the standard script letter is not the only acceptable form of a letter. The learner might have their own writing style that is somewhat different from the standard letter but can still result in correct letter forms. The best solution would be one that keeps the personal style of the letter and changes only the minimum necessary to make the letter correct. This way the learner gets feedback about what they really need to correct.

A similar task, of keeping some characteristics in data while changing others is tackled by the task of style transfer that has also been studied for handwriting.

Hu et al. (2019) presented a method and tool for smooth interpolation between images of different styles even when only one sample of a given style is provided. They trained VAEs on EMNIST to learn character representations and then fine-tunes the models on samples of their own handwriting to create person-specific networks for style. After creating the individualized style networks, they investigated latent space clusterings and linear transformations as potential methods for extracting semantic meaning from the learned representations.

Kotani et al. (2020) propose a model that allows to distinguish between character-specific and writer-specific style components in letters which allows them to generate

handwriting in new writer styles based on only a few examples.

(Aksan et al. 2018) proposes a generative neural network architecture capable of disentangling style from content and thus making online handwriting editable. Their model can synthesize arbitrary text, while giving users control over the visual appearance (style). For example, allowing for style transfer without changing the content, editing of digital ink at the word level and other application scenarios such as spell-checking and correction of handwritten text.

While all of these techniques seem promising to be adapted for the task here, the scope of this thesis allowed only to choose one for experimenting. Therefore an approach using VAEs was chosen. VAEs are a type of generative model that has gained popularity recently (Kingma and Welling 2019; Doersch 2021). The advantage of VAEs is that they encode multidimensional data into a lower-dimensional latent space. The encoding in this latent space can then be used to decode the data back to the full multidimensional version, however with a certain loss of information. The properties of the VAEs allow for manipulating the data in the latent space and then sampling the manipulated data. This way, the general information of letter images, like distribution of black and white and type of lines can be stored in the trained weights while the relevant information differentiating one letter from another is accessible in the latent space.

Therefore, the VAE approach was chosen here to see whether it is possible to use this encoding to transform the given letter in the latent space into a more correct form. The assumption is that each alphabet letter is represented by a latent space clustering the dimensions of which can be calculated. While the sample latent space encoding shown in figure 5.4 give rise to the assumption that the letter distributions in the latent space do not fully follow a normal distribution, it is nevertheless tried to approximate the distributions with multivariate Gaussian distributions as other clustering types are algorithmically more complicated. If this approximation turns out not to be enough for the present case, other methods will need to be explored.

With these conditions in place, the following steps are taken: A VAE with the architecture shown in listing 5.2 ist trained on the dataset. Then, the means and variances of the letter encoding distributions in the latent space are calculated and saved for reference. Then, wenn an incorrect letter is to be transformed into a more correct form, it is encoded into the latent space. Assuming that the letter distributions are built in such a way that the more frequent letter forms are distributed in the centre and the more atypical representations are found more at the edge of the distribution and that a straight line is the most direct path to transform the given letter into a more correct form, a line is calculated between the latent coordinates of the given letter and the mean of the letter's distribution. As the goal is not just to return the optimal solution, which would be the mean letter in this approach, a suitable distance needs to be determined for sampling the transformed letter. As the distribution is treated as normal here, a Mahanalobis distance of 1 is assumed to be a good starting point for experimentation.

```
==========================================================================
Layer (type (var_name))  Input Shape          Output Shape         Param #
```

```
==========================================================================
VAE (VAE)                [64, 784]           [64, 784]           --
 -Sequential (encoder)   [64, 784]           [64, 200]           --
    - Linear (0)         [64, 784]           [64, 400]           314,000
    - LeakyReLU (1)      [64, 400]           [64, 400]           --
    - Linear (2)         [64, 400]           [64, 200]           80,200
    - LeakyReLU (3)      [64, 200]           [64, 200]           --
 -Linear (mean_layer)    [64, 200]           [64, 16]            3,216
 -Linear (logvar_layer)  [64, 200]           [64, 16]            3,216
 -Sequential (decoder)   [64, 16]            [64, 784]           --
    - Linear (0)         [64, 16]            [64, 200]           3,400
    - LeakyReLU (1)      [64, 200]           [64, 200]           --
    - Linear (2)         [64, 200]           [64, 400]           80,400
    - LeakyReLU (3)      [64, 400]           [64, 400]           --
    - Linear (4)         [64, 400]           [64, 784]           314,384
    - Sigmoid (5)        [64, 784]           [64, 784]           --
==========================================================================
Total params: 798,816
Trainable params: 798,816
Non-trainable params: 0
Total mult-adds (M): 51.12
==========================================================================
Input size (MB): 0.20
Forward/backward pass size (MB): 1.03
Params size (MB): 3.20
Estimated Total Size (MB): 4.43
==========================================================================
```

Listing 5.2: VAE architecture for letter transformation with a latent space dimensionality of 16

Some examples of the latent space transformations are shown in figures 5.4 and 5.5. These examples were created with a small VAE over just 4 letters and a 2-dimensional latent space to allow for visualisation. The letter distributions of the letters encoded into the latent space are represented by the coloured dots in figure 5.4 and each distribution's mean and confidence ellipse at standard deviation of 1.0 is shown in the distributions. Then four random coordinates were taken in the latent space and moved along the line between random point and mean to the place where that line crosses the confidence ellipse. That coordinate is then decoded from the latent space into the full 784 dimensions of the letter.

Figure 5.5 shows the letter images decoded from the latent space coordinates. It can be seen that coordinates deep within the distributions do indeed decode to correct and good letters. Whereas letters outside or at the borders of distributions do not. The latent coordinates of random letter 'a' are so far out of all of the distributions its decoding is not even recognisable as a letter. The latent coordinates of random letter 'b' are within the distribution of letter d and, accordingly, its decoding looks like d. The latent coordinates of random letter 'c' are at the edge of the distribution of letter c and its decoding is still recognisable as a c. All of the transformed letters at std 1.0 are deep

within the respective distributions and thus correct. However, the transformed d at std 1.5 is square on the border between the d and a distributions and thus not recognisable as either.

With this analysis in place, it seems that the approach is working as expected and therefore it is used for the experiments here.

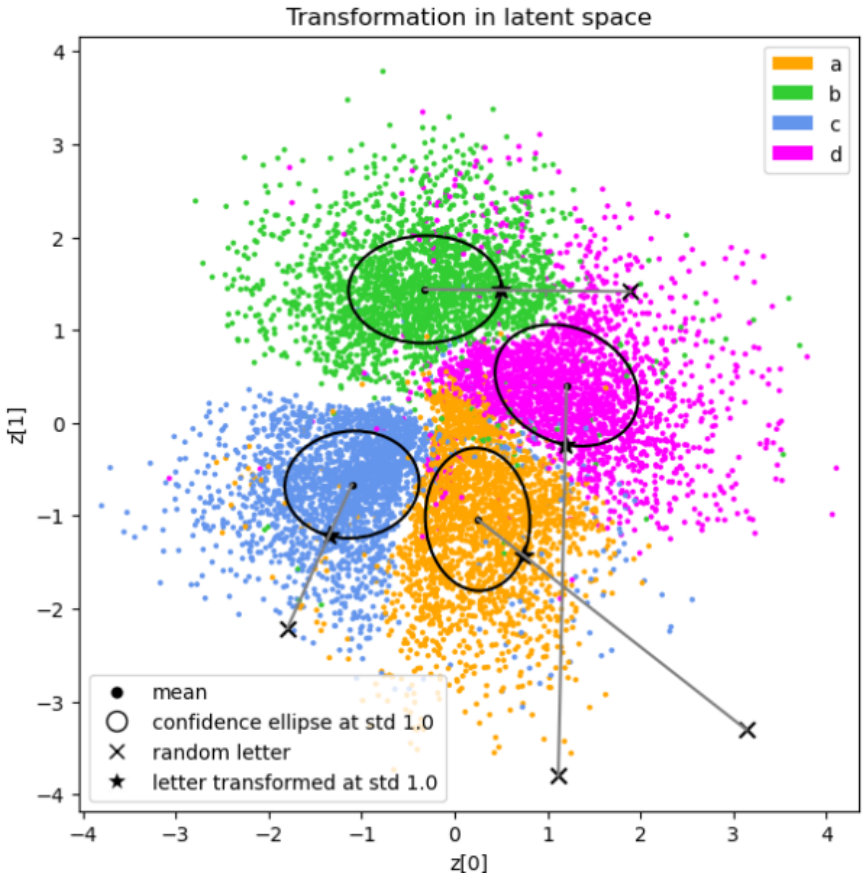


Figure 5.4: Latent space clusterings of letters of a small VAE with 4 letters and a latent space dimension of 2, Mahanalobis distance ellipses and transformed letters

5.3.3 Letter features

The third part of the detailed feedback algorithm comprises the marking of characteristic parts and shapes of letters like ascenders, descenders, closed and open forms, curves, straight lines in the letter. A table was created listing the characteristic forms for each letter (see table 5.1 for an excerpt from the feature table). These features are the same features that were used for creating the segmentation dataset as described in section 4.3. For this proof of concept, the letter segmentation was simplified insofar as each letter is represented by only one combination of features in this dataset where in reality

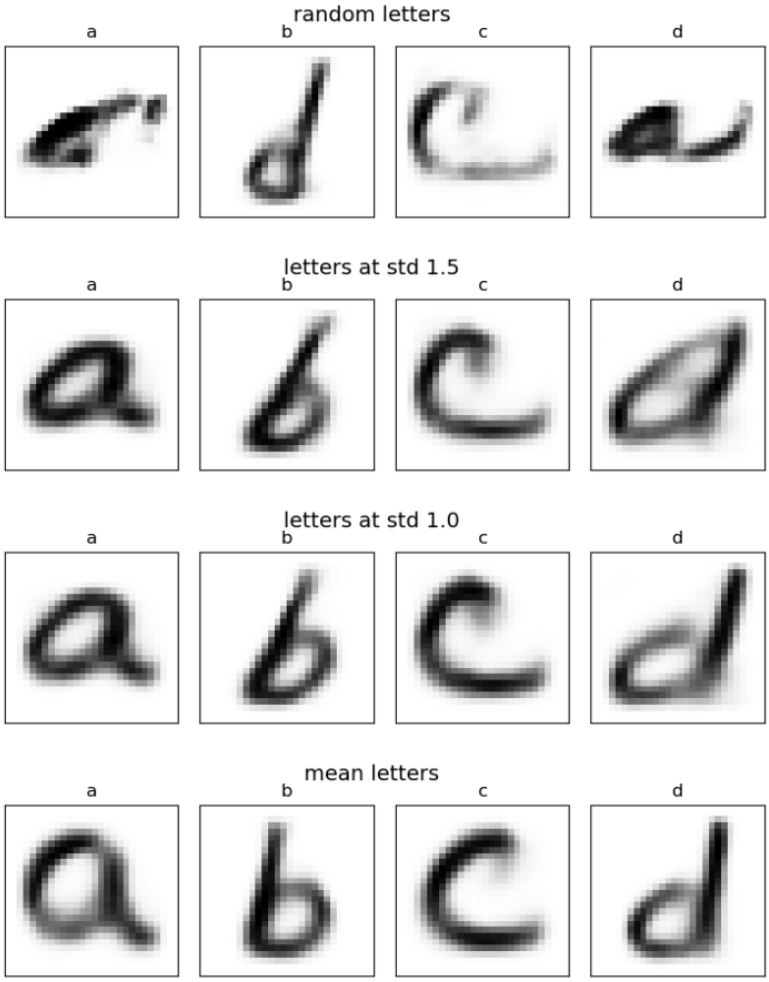


Figure 5.5: Letter transformations

some letters exist in different variants.

	straight lines				
	bottom horizontal	**left ascender**	**left diagonal**	**right ascender**	**right diagonal**
K	0	1	1	0	1
L	1	1	0	0	0
M	0	1	1	1	1
N	0	1	1	1	0

Table 5.1: Letter features

In general, features can be found in images with object detection methods. For this, image classification and instance segmentation are of relevance. Typical implementations of segmentation models like Redmon et al. (2016), Ren et al. (2016) and He

et al. (2018) make use of transfer learning with backbone architectures (ResNet, VGG, Inception, etc.) and retraining some layers on new data. As the present task uses considerably smaller images than the typical tasks, it is, however, not necessary or useful to use those rather big pre-trained models, as even the final training of these requires the use of GPU. Therefore, a simple CNN model is built and trained from scratch for this task (model architecture see listing 5.3). A separate model is trained for each letter to make use of the fact that the expected letter is known in this task. In contrast to the EMNIST images, an image size of 56 x 56 was chosen for the segmentation model to allow for a more detailed representation of the segmentation masks as it was suspected that the results of segmenting at a size of 28 x 28 would be too vague. The images and masks of the segmentation dataset were created at that bigger size, so the segmentation models do contain that amount of information. The images coming from previous steps of the pipeline have a smaller size and therefore need to be upscaled for use with the segmentation models.

```
==========================================================================
Layer (type (var_name))    Input Shape        Output Shape       Param #
==========================================================================
SegmentationModel          [32, 1, 56, 56]    [32, 3, 56, 56]    --
 - Conv2d (conv1)          [32, 1, 56, 56]    [32, 16, 56, 56]   160
 - Conv2d (conv2)          [32, 16, 56, 56]   [32, 32, 56, 56]   4,640
 - MaxPool2d (pool1)       [32, 32, 56, 56]   [32, 32, 28, 28]   --
 - Conv2d (conv3)          [32, 32, 28, 28]   [32, 32, 28, 28]   9,248
 - Conv2d (conv4)          [32, 32, 28, 28]   [32, 32, 28, 28]   9,248
 - MaxPool2d (pool2)       [32, 32, 28, 28]   [32, 32, 14, 14]   --
 - Conv2d (conv5)          [32, 32, 14, 14]   [32, 32, 14, 14]   9,248
 - Conv2d (conv6)          [32, 32, 14, 14]   [32, 32, 14, 14]   9,248
 - Upsample (upsample1)    [32, 32, 14, 14]   [32, 32, 28, 28]   --
 - Conv2d (conv7)          [32, 32, 28, 28]   [32, 32, 28, 28]   9,248
 - Conv2d (conv8)          [32, 32, 28, 28]   [32, 32, 28, 28]   9,248
 - Upsample (upsample2)    [32, 32, 28, 28]   [32, 32, 56, 56]   --
 - Conv2d (conv9)          [32, 32, 56, 56]   [32, 32, 56, 56]   9,248
 - Conv2d (conv10)         [32, 32, 56, 56]   [32, 16, 56, 56]   4,624
 - Conv2d (conv11)         [32, 16, 56, 56]   [32, 3, 56, 56]    435
==========================================================================
Total params: 74,595
Trainable params: 74,595
Non-trainable params: 0
Total mult-adds (G): 2.96
==========================================================================
Input size (MB): 0.40
Forward/backward pass size (MB): 108.38
Params size (MB): 0.30
Estimated Total Size (MB): 109.08
==========================================================================
```

Listing 5.3: CNN for letter segmentation

When feedback is created for a letter that was recognised as a different letter, the following approach is followed: the set of the segments that are listed in the feature table for the expected letter is created as well as the set of segments for the recognised letter. The intersection of both sets is taken and the features in that intersection, aka features existing in both letters are marked in green in the original letter image using the segmentation model for the recognised letter. The expected features for the expected letter not found in the feature set of the recognised letter are marked in the transformed letter image in blue. Then, both the green segments and the blue segments are overlayed over the original image indicating what is already correct and what needs improvement. For an example of this step see 5.6

This approach assumes that the transformation actually leads to letters that are changed as much as necessary but as little as possible, so that the combination of features from two different images results in a legible letter.

(a) (b) (c)

Figure 5.6: Example segmentation and marking of left ascender (green) taken as correct from the original letter recognised as *L* and the right and left diagonal (blue) from the transformed letter *K*

This combination of the correct letter features from the original letter and the changed features from the transformed image, however, works only when the recognised letter is different from the expected letter. If the letter was recognised as correct but of insufficient quality, this approach does not work as both images are of the same letter and therefore the intersection would contain all letter features and mark all features as correct in the original image. This, of course, is not the intended result. Therefore it was decided to mark the whole transformed letter in blue as to be improved. A more detailed approach for these letters is beyond the scope here.

Chapter 6

Evaluation

This chapter analyses the performance of the developed approach by calculating the experiment metrics in section 6.4, analysing errors in section 6.5 and discussing the results in section 6.6.

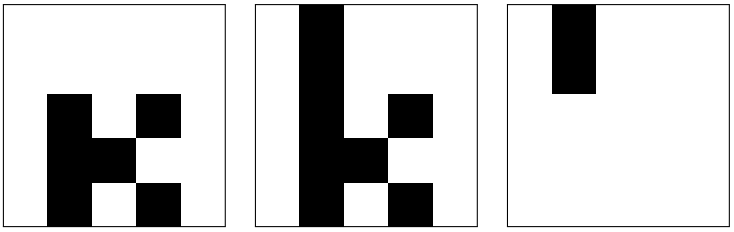
6.1 Evaluation setup

The purpose of giving feedback is to improve the quality of the written letter. Therefore the feedback should lead to better letters. As an actual evaluation of learners' learning from the feedback provided is out of the scope of this thesis, it will be assumed that the feedback given leads to the learner being able to write letters in the form of the feedback.

Therefore, it will be evaluated whether the feedback has a better quality than the original letter provided. This is done by doing step 1 of the algorithm both on the original letter image as well as on the feedback image and then comparing the prediction of the original image against the prediction of the feedback image. If the prediction of the feedback is for the correct class and is higher than the prediction for the original image, the feedback is considered to have a better quality than the original letter provided.

This evaluation is done both for the baseline algorithm and for the detailed feedback algorithm. As the baseline algorithm is always returning the standard letter it is to be expected that the prediction for the baseline feedback is very high and, thus, that the detailed feedback algorithm can never return higher quality letters than the baseline. Therefore, another metric is needed to also account for the fact that the detailed feedback algorithm is intentionally not returning the perfect letter but a letter that is correct but still as similar as possible to the original letter. Thus, the pixel difference is chosen as a secondary criterion to measure difference between original and feedback. The pixel difference is calculated by turning both images into black-and-white-images and then pixel-wise counting which pixels differ between original and feedback image, see example in figure 6.1.

Both metrics are negatively correlated and therefore a further combined metric is needed to evaluate the overall quality of the feedback. Therefore, the quality is measured as follows: for each letter image it is checked whether the feedback is for the correct class and has a prediction probability of above 0.8, the cut-off value used for step 1 of the algorithm. This prediction probability is considered sufficient to be graded as



(a) original **(b)** feedback **(c)** pixel difference $2/25 = 0.08\%$

Figure 6.1: Example for calculation of image pixel difference

correct. If so, the pixel difference between original image and detailed feedback image is calculated as well as between the original image and the baseline feedback image. The feedback with the smaller pixel difference is considered better. The count of better images per feedback type is taken and measured with the percentage of total images evaluated. See the graph in figure 6.2 for the detailed evaluation steps.

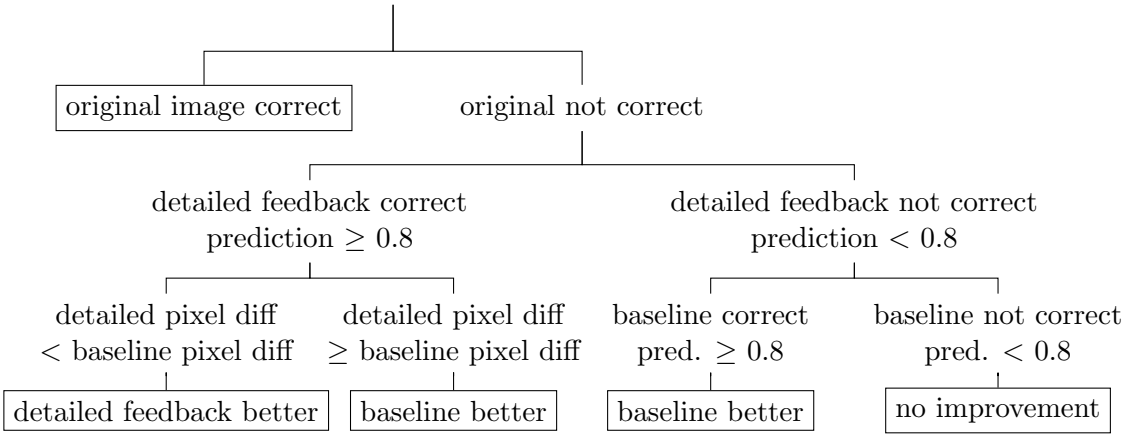


Figure 6.2: Combined image-wise evaluation

The evaluation metrics are calculated for the original image, the baseline feedback and the detailed algorithm feedback. The evaluation of the detailed feedback is calculated for several latent space locations in the second step of the algorithm to show the influence of the latent space locations on the output result.

The latent space location is calculated for several points along the line between the original letter latent space coordinates and the distribution mean of the letter. The values are calculated for the relative distances at 20% intervals starting at relative distance 0% (aka the original letter coordinates themselves) and reaching till relative distance 100% (aka the distribution mean for the letter in question). The purpose of calculating all these relative distance values is to show that the metrics gradually change from the original value towards the distribution mean.

6.2 Experiment data

The data used for the evaluation is the merged and padded EMNIST datasets as described in section 4.2 and the merged and padded questionnaire learners' samples described in section 4.4

As the EMNIST letters are letters from proficient writers, most of the letters can be considered some degree of correct, even if they are not pretty because they may be written fast. But as the objective of the app is not to grade the beauty of writing, but its correctness and legibility, the decision was taken to relabel the images as a different class so that the letters will be classified as incorrect and the algorithm can take effect. Otherwise the functionality of the algorithm could only have been evaluated with a very small number of images.

Therefore, 50% of the images were relabeled for the evaluation. For the other half the original labels were kept.

The labels of the learners' datasets were kept as they were.

6.3 Technical specifications

The experiments were done on simple consumer electronics.

Hardware Specifications:

- Processor (CPU): Intel(R) Core(TM) i5-7600 CPU @ 3.50 GHz
- Graphics Processing Unit (GPU): none
- Random Access Memory (RAM): 16.0 GB
- Storage: 1 TB SSD

Software Stack:

- Operating System: Windows 10
- Machine Learning Framework: PyTorch 1.13.1

6.4 Experiment results

The experiment results for the average prediction probabilities per original prediction, baseline prediction and detailed algorithm prediction at increasing latent space distances are shown in figure 6.3 and the results for the pixel differences between original image and baseline and detailed algorithm images at increasing latent space distances are shown in figure 6.4.

The results of the combined evaluation for the EMNIST dataset are shown in figure 6.5 and for the questionnaire learners' dataset in figure 6.6.

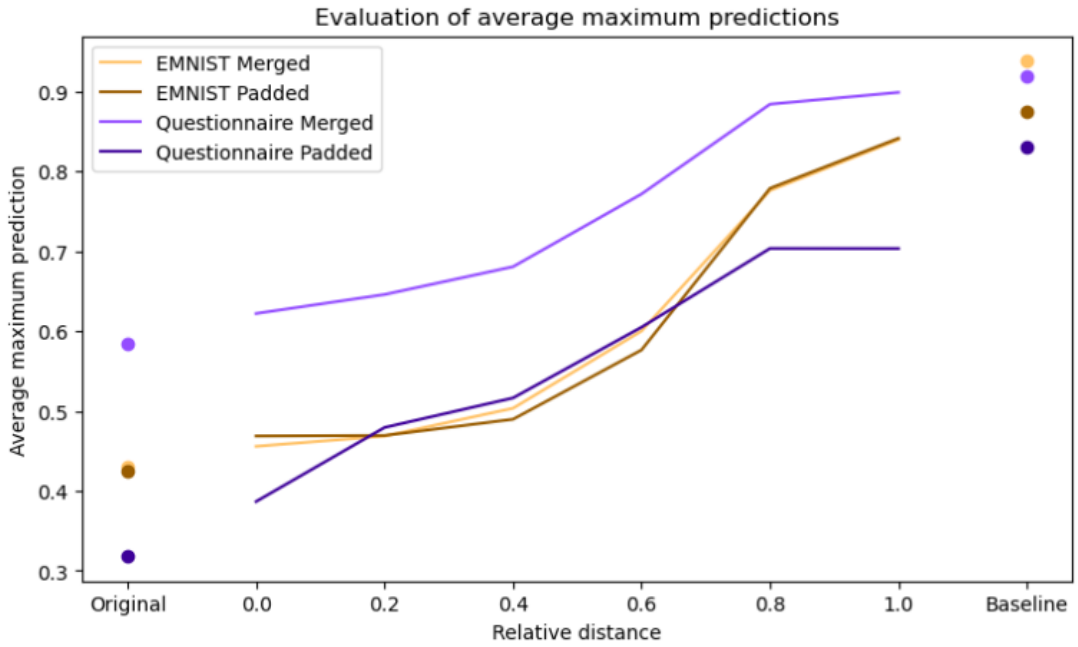


Figure 6.3: Average maximum prediction values for feedback by dataset and transformation distance

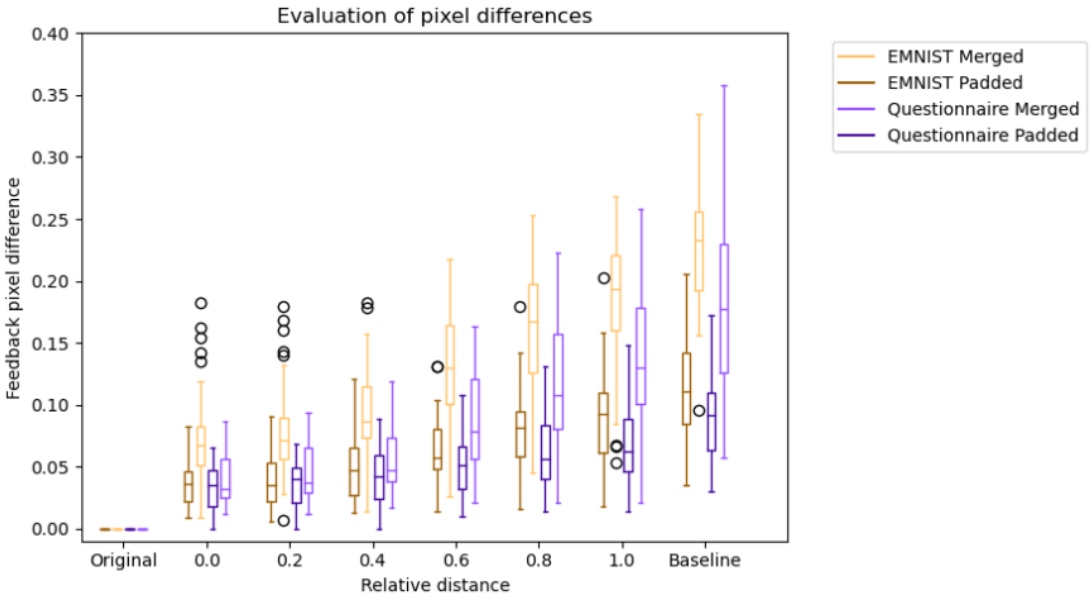
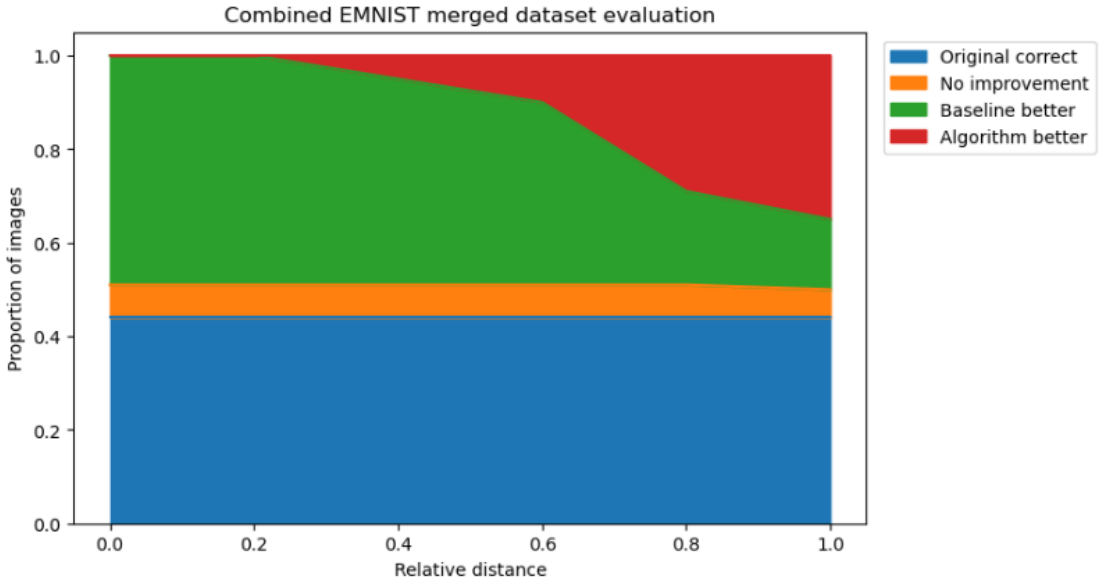
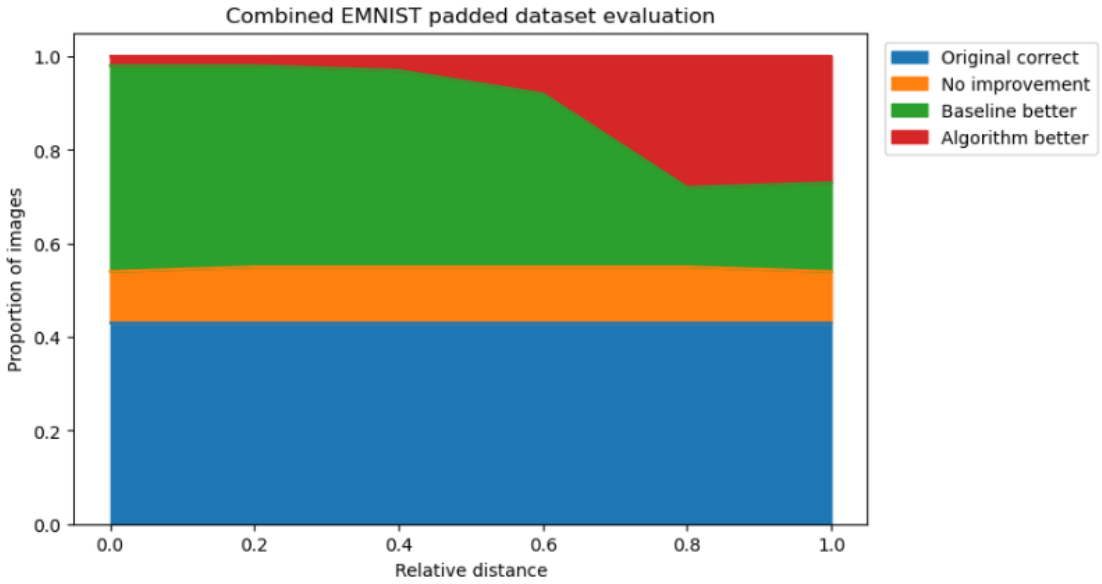


Figure 6.4: Pixel differences between original image and feedback by dataset and transformation distance



(a) EMNIST merged dataset

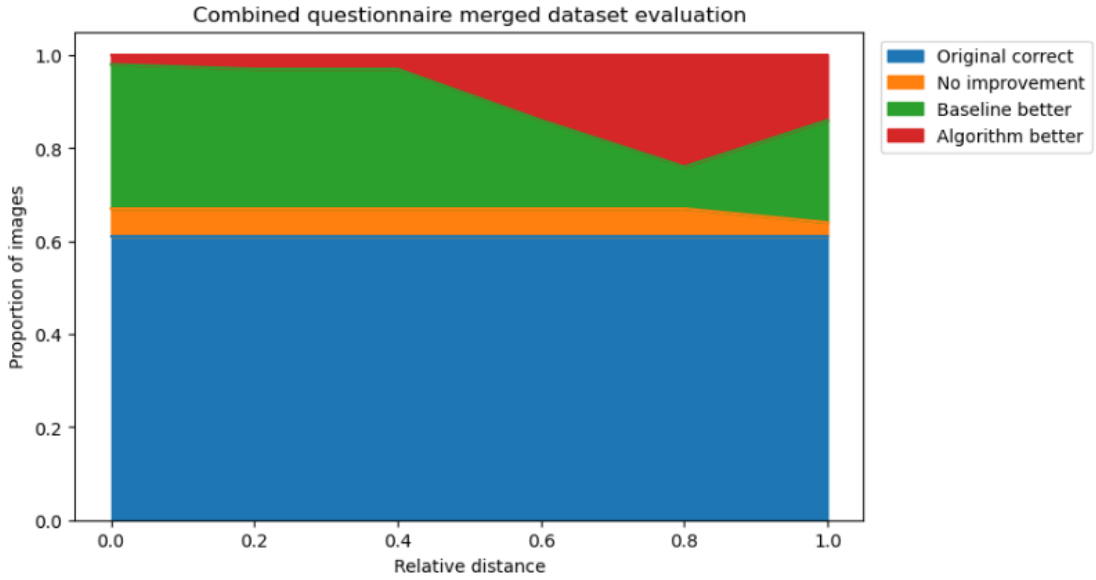


(b) EMNIST padded dataset

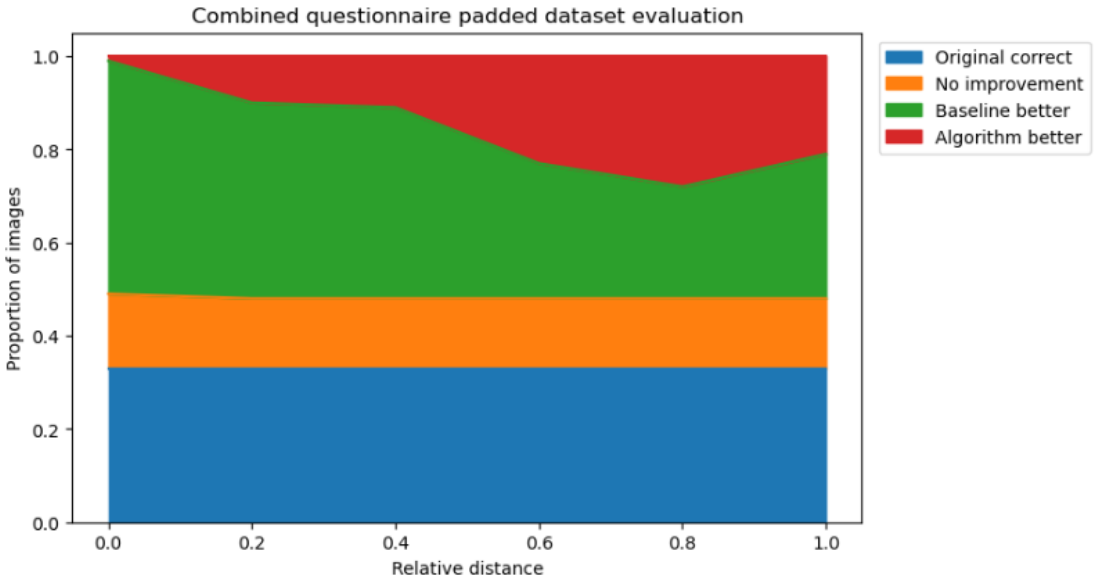
Figure 6.5: Evaluation of EMNIST dataset results

6.5 Error analysis

The results in figure 6.3 show that the letter quality does indeed improve when moving the transformed letter towards the letter's latent space mean. And also the results in figure 6.4 show that the pixel difference increases along that line. Also the pixel



(a) learners' merged dataset



(b) learners' padded dataset

Figure 6.6: Evaluation of learners' dataset results

differences for the padded data are smaller than for the merged data which is according to expectations as the padded images contain more white space that is not changed. These results are as expected and indicate that the approach of transforming a letter from the given form into a more correct form seems to work.

Analysis of the combined evaluation results: For the EMNIST datasets 50% of the images were relabeled. According to the analysis, 44% of images were recognised as correct which fits the number of relabelings plus about 6% of letters graded as incorrect. The class *No improvement* should, in the optimal case, be empty as the baseline feedback should be graded as of high quality. However, as the grading model was trained on different data, not including the baseline feedback, a few of the baseline feedback letters are not recognised as of sufficient quality and if the detailed algorithm for that letter also is not sufficient, both feedback versions are considered as insufficient.

The baseline feedback is always the same per letter, it does not differentiate between transformation distance as the detailed feedback algorithm does. Therefore, at lower transformation distances of the detailed feedback algorithm the feedback is not transformed enough to be sufficiently recognised as the expected letter. Therefore, the baseline algorithm *wins* over the detailed algorithm. But a transformation distance of 0.8 seems to result in a sufficient transformation for the letter quality to be better on average than the baseline algorithm. However, this trend does not continue when moving the transformation distance the full 100% towards the latent space letter mean, as then the pixel distance increases further and gets bigger for the detailed feedback than for the baseline feedback and, thus, the baseline feedback *wins* again for some images.

However, the maximum average letter quality of 0.84 for both the padded and merged EMNIST datasets currently reachable with the detailed feedback algorithm is still too close to the cutoff value of 0.8 and not yet satisfactory or sufficient for actual application of the feedback algorithm. Some example feedback for the datasets is shown in figures 6.7 and 6.8. The reasons will be discussed in the following subsections analysing the individual steps of the algorithm.

6.5.1 Letter quality assessment

The quality assessment seems to work very reliably for relabeled data, meaning the model recognises the original class of a letter before relabeling and therefore categorises it as different from the expected, aka relabeled class. However, as the dataset used is not an actual learners' dataset, no real evaluation is possible whether this approach of using the prediction probabilities as proxy for the letter quality also works with more detailed errors than writing completely different letters than expected. Therefore this step was not studied further here, as this requires different training data.

6.5.2 Transformation in latent space

As shown in the results section 6.4, the VAE approach allows for gradual transformation of a given letter into a correct letter. The difficult part is finding a suitable value for the distance. The examples shown in figures 6.7 and 6.8 were transformed to a Mahanalobis distance of 1.0 from the letter mean in the latent space. Currently, the same Mahanalobis distance is used for all letters. As the visualisation for a 2-dimensional latent space in figure 5.4 indicates, the approximation of the letter distributions with a multivariate normal distribution does not fully capture the cluster boundaries. Therefore, another

approach might be necessary for exact determination of the transformation coordinates in the latent space.

6.5.3 Letter features

Analysing the results from the segmentation step (see figure 6.9), indicates that the segmentation models are well able to differentiate between white background and black letter strokes. They can also pick out distinct segments within the continuous black letter strokes. Where the current segmentation models seem to have problems, is differentiating between parallel strokes and, thus, making it impossible to mark certain features only in certain areas of the image. The reason for this might be the model architecture or the training data used or both. It might be necessary to use a more complex model architecture to also train location information.

With regard to the training data, currently the segmentation model for one letter was only trained with images and masks of that one letter. It might be necessary to also train the model with examples of other letters that also contain some of the features of the training letter or none of them, so that the model does not over-adapt to the features sought.

The function of the segmentation step is critical for the overall algorithm results, especially in the case of letters were both the recognised and the expected letter share features. Because then, the algorithm combines the feedback from both features from the original letter (marked in green) and from the transformed letter (marked in blue). For this to work, both the transformation and the segmentation steps need to be well atuned to each other. At the current level of both steps, the combined result only works in some cases, in others it produces somewhat irregular stroke combinations.

6.6 Discussion

Based on the evaluation and error analysis, the hypotheses for this thesis can be discussed as follows:

Hypothesis 1: The results of a handwriting recognition model can be used to assess the quality of a letter. This hypothesis cannot be finally answered from the experiments done, as the training dataset used was not a learners' handwriting dataset but a dataset of handwriting samples from experienced writers. For the dataset used, it seems that it is possible to use the recognition model predictions as proxy for the letter quality, at least at the rather rough level that could be assessed by relabeling classes and transforming letters from one letter to another. Whether this holds true for more detailed corrections from an incorrectly written letter to a more correct shape, requires further experimentation.

Hypothesis 2: A letter can be transformed in the latent space of a Variational Autoencoder to create a similar but correct version of the same letter. This hypothesis

was shown to be true. Moving the latent space coordinates of the letter to be corrected did result in a more correct version of the letter based on the given dataset.

But as the dataset was not optimal for a learning context, the transformed letter mean is not guaranteed to be the *most perfect* rendition of the letter. Therefore, a different dataset would be required to study this further. Also, further research into the localisation within the latent space seems useful.

Hypothesis 3: An object detection model can be used to mark letter features in a letter. This hypothesis was shown to be true. A simple object detection model works to mark letter features. However, the simple model and training data used here have produced mixed results. Therefore, further experiments into using a segmentation model for letter feature recognition seem necessary.

Another difficult question is what parts of letters to categorize as a feature so that this allows for generating valuable feedback for learners.

(a) EMNIST merged dataset

(b) EMNIST padded dataset

Figure 6.7: Feedback for EMNIST with transformation at a mahanalobis distance of 1.0

(a) learners' merged dataset

(b) learners' padded dataset

Figure 6.8: Feedback for learners with transformation at a mahanalobis distance of 1.0

Figure 6.9: Segmentation results for example letters of the uppercase and lowercase alphabets

Chapter 7

Conclusion

7.1 Conclusion

This thesis studied a method for automatic generation of feedback for second script learners. The research question *How can feedback for single letter handwriting exercises in Second language (L2) language tuition be generated automatically?* was theoretically discussed and practically realised with a proof-of-concept implementation. The evaluation of the implementation found that...

The presented approach to handwriting feedback generation is a novel approach insofar as it does not only compare the written letter with a strict standard letter definition but allows for a more free form acceptance of legible letters and correction of the letters within the writer's style. While the existing implementation is not yet ready for actual use, the approach can have its place in app assisted handwriting learning at a learning stage between first familiarisation with new letter forms where tracing does have its place and more advanced writing exercises where letter forms can be expected to be known and questions of spelling, grammar and punctuation gain more importance.

7.2 Outlook

Options for further research

- Get actual learners' samples
- Get expert annotation of sample quality
- Try out vector data
- Adapt VAE to also take labels as input. This information is available so why not use it.
- increase the segmentation feature dataset to include variations of letters

Appendix

A Code documentation

The code is organized as follows:

Installation Install libraries according to the info in `requirements.txt`.

Usage Start `evaluation.py` to calculate the evaluation results.
Start `visualisation.py` to create the following figures in the thesis:

- figure
- figure
- figure

The following Jupyter Notebooks can be run to visualise random examples from the datasets with feedback:

- `Baseline_Pipeline.ipynb`
- `Full_Pipeline.ipynb`
- `Full_Pipeline_Questionnaire.ipynb`

The notebooks allow to switch between the padded and merged datasets.

Bibliography

E. Aksan, F. Pece, and O. Hilliges. DeepWriting: Making Digital Ink Editable via Deep Generative Modeling, Jan. 2018. URL http://arxiv.org/abs/1801.08379.

H. M. Al-Khattat. *Qawa'id al-khatt al-arabi [Arabic] (Rules of Arabic Calligraphy)*. Matibi' Youssuf Baidoun, Beirut, 1986. URL http://archive.org/details/20191016_20191016_1350.

H. P. Althaus. 11. Graphetik. In *Lexikon der Germanistischen Linguistik [German] (Graphetics, Lexicon of German Linguistics)*, pages 138–142. Max Niemeyer Verlag, Aug. 2011. ISBN 978-3-11-096084-6. DOI: 10.1515/9783110960846.138. URL https://www.degruyter.com/document/doi/10.1515/9783110960846.138/html.

T. Asselborn, T. Gargot, L. Kidzinski, W. Johal, D. Cohen, C. Jolly, and P. Dillenbourg. Automated human-level diagnosis of dysgraphia using a consumer tablet. *NPJ Digital Medicine*, 1:42, Aug. 2018. ISSN 2398-6352. DOI: 10.1038/s41746-018-0049-x. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6550155/.

A. Baldominos, Y. Saez, and P. Isasi. A Survey of Handwritten Character Recognition with MNIST and EMNIST. *Applied Sciences*, 9(15):3169, Jan. 2019. ISSN 2076-3417. DOI: 10.3390/app9153169. URL https://www.mdpi.com/2076-3417/9/15/3169. Number: 15 Publisher: Multidisciplinary Digital Publishing Institute.

BAMF. Konzept für einen bundesweiten Integrationskurs für Zweitschriftlernende [German] (Concept for a nationwide integration course for second-script learners), 2018. URL https://www.BAMF.de/SharedDocs/Anlagen/DE/Integration/Integrationskurse/Kurstraeger/KonzepteLeitfaeden/konzept-zweitschriftlernende.html?nn=282388.

B. Bassetti and V. Cook. An introduction to researching Second Language Writing Systems. In B. Bassetti and V. Cook, editors, *Second language writing systems*, pages 1 – 67. Multilingual Matters., Clevedon, UK, Jan. 2005.

M. Becker-Mrotzek and J. Grabowski, editors. *FD-LEX (2018). Forschungsdatenbank Lernertexte. [German] (Research Database of Learners' Texts)*. Mercator-Institut für Sprachförderung und Deutsch als Zweitsprache, Cologne, 2018. URL fd-lex.uni-koeln.de.

A. Berkemeier. *Kognitive Prozesse beim Zweitschrifterwerb: Zweitalphabetisierung griechisch-deutsch-bilingualer Kinder im Deutschen [German] (Cognitive processes*

in the acquisition of second script skills: Second literacy of Greek-German bilingual children in German). Peter Lang, 1997.

V. W. Berninger, R. D. Abbott, J. Jones, B. J. Wolf, L. Gould, M. Anderson-Youngstrom, S. Shimada, and K. Apel. Early Development of Language by Hand: Composing, Reading, Listening, and Speaking Connections; Three Letter-Writing Modes; and Fast Mapping in Spelling. *Developmental Neuropsychology*, 29(1):61–92, Feb. 2006. ISSN 8756-5641, 1532-6942. DOI: 10.1207/s15326942dn2901_5. URL http://www.tandfonline.com/doi/abs/10.1207/s15326942dn2901_5.

M. Bublin, F. Werner, A. Kerschbaumer, G. Korak, S. Geyer, L. Rettinger, E. Schönthaler, and M. Schmid-Kietreiber. Handwriting Evaluation Using Deep Learning with SensoGrip. *Sensors*, 23:5215, May 2023. DOI: 10.3390/s23115215.

A. Böttinger. *Schritte Plus Alpha 1 [German] (Steps Plus Alpha 1].* Hueber, Munich, 2011.

A. Böttinger and A. Wiebel. *SchlaU - Deutsch als Zweitsprache [German] (Smart - German as second language).* Werkstatt für Migrationspädagogik. Springer, München, 2018.

S. G. Calle. Lernende mit Tigrinya als Erstschriftsprache fördern und begleiten [German] (Support and accompany learners with Tigrinya as their first written language). In B. Marschke, editor, *Handbuch der kontrastiven Alphabetisierung [German] (Manual of contrastive literacy)*, pages 285–296. Erich Schmidt Verlag GmbH & Co. KG, Berlin, 2022. ISBN 978-3-503-20655-1. DOI: 10.37307/b.978-3-503-20655-1.13. URL https://doi.org/10.37307/b.978-3-503-20655-1.13.

E. Chartrel and A. Vinter. The impact of spatio-temporal constraints on cursive letter handwriting in children. *Learning and Instruction*, 18(6):537–547, Dec. 2008. ISSN 0959-4752. DOI: 10.1016/j.learninstruc.2007.11.003. URL https://www.sciencedirect.com/science/article/pii/S0959475207001363.

G. Cohen, S. Afshar, J. Tapson, and A. van Schaik. EMNIST: an extension of MNIST to handwritten letters, Mar. 2017. URL http://arxiv.org/abs/1702.05373. arXiv:1702.05373 [cs].

C. Cucchiarini, I. V. D. Craats, J. Deutekom, and H. Strik. The digital instructor for literacy learning. In *Speech and Language Technology in Education (SLaTE 2013)*, pages 96–101. ISCA, Aug. 2013. DOI: 10.21437/SLaTE.2013-16. URL https://www.isca-archive.org/slate_2013/cucchiarini13_slate.html.

J. Danna and J.-L. Velay. Basic and supplementary sensory feedback in handwriting. *Frontiers in Psychology*, 6, Feb. 2015. ISSN 1664-1078. DOI: 10.3389/fpsyg.2015.00169. URL https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2015.00169/full.

C. Doersch. Tutorial on Variational Autoencoders, Jan. 2021. URL http://arxiv.org/abs/1606.05908. arXiv:1606.05908 [cs, stat].

L. C. Ehri. Phases of Development in Learning to Read Words by Sight. *Journal of Research in Reading*, 18(2):116–25, 1995. ISSN 0141-0423. ERIC Number: EJ514638.

E. Fabiani, J.-L. Velay, C. Younes, J.-L. Anton, B. Nazarian, J. Sein, M. Habib, J. Danna, and M. Longcamp. Writing letters in two graphic systems: Behavioral and neural correlates in Latin-Arabic biscripters. *Neuropsychologia*, 185:108567, July 2023. ISSN 0028-3932. DOI: 10.1016/j.neuropsychologia.2023.108567. URL https://www.sciencedirect.com/science/article/pii/S002839322300101X.

E. G. Feldgus and I. Cardonick. *Kid Writing: A Systematic Approach to Phonics, Journals, and Writing Workshop*. Wright Group, Jan. 2002. ISBN 978-0-322-06435-5.

A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera. *Learning from Imbalanced Data Sets*. Springer International Publishing, Cham, 2018. ISBN 978-3-319-98073-7 978-3-319-98074-4. DOI: 10.1007/978-3-319-98074-4. URL http://link.springer.com/10.1007/978-3-319-98074-4.

C. L. Fitjar, V. Rønneberg, and M. Torrance. Assessing handwriting: a method for detailed analysis of letter-formation accuracy and fluency. *Reading and Writing*, May 2022. ISSN 1573-0905. DOI: 10.1007/s11145-022-10308-z. URL https://doi.org/10.1007/s11145-022-10308-z.

B. Florence and B.-B. Nathalie. Handwriting isolated cursive letters in young children: Effect of the visual trace deletion. *Learning and Instruction*, 74:101439, Aug. 2021. ISSN 0959-4752. DOI: 10.1016/j.learninstruc.2020.101439. URL https://www.sciencedirect.com/science/article/pii/S0959475220307349.

U. Frith. Beneath the surface of developmental dyslexia. *Developmental dyslexia*, 13, Jan. 1985.

S. Gerth, T. Dolk, A. Klassert, M. Fliesser, M. H. Fischer, G. Nottbusch, and J. Festman. Adapting to the surface: A comparison of handwriting measures when writing on a tablet computer and on paper. *Human Movement Science*, 48:62–73, Aug. 2016. ISSN 1872-7646. DOI: 10.1016/j.humov.2016.04.006.

S. Graham. A review of handwriting scales and factors that contribute to variability in handwriting scores. *Journal of School Psychology*, 24(1):63–71, Mar. 1986. ISSN 0022-4405. DOI: 10.1016/0022-4405(86)90043-9. URL https://www.sciencedirect.com/science/article/pii/0022440586900439.

K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN, Jan. 2018. URL http://arxiv.org/abs/1703.06870. arXiv:1703.06870 [cs].

Z. He, H. Xie, and K. Miyata. Interactive Projection System for Calligraphy Practice. pages 55–61, June 2020. DOI: 10.1109/NicoInt50878.2020.00018.

W. Hu, S. Thompson, and N. Tsoi. LLAMA: Learning Latent trAnsforMations for generative style trAnsfer. 2019.

D. P. Kingma and M. Welling. An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019. ISSN 1935-8237, 1935-8245. DOI: 10.1561/2200000056. URL http://arxiv.org/abs/1906.02691. arXiv:1906.02691 [cs, stat].

F. Kleber, S. Fiel, M. Diem, and R. Sablatnig. CVL-DataBase: An Off-Line Database for Writer Retrieval, Writer Identification and Word Spotting. In *2013 12th International Conference on Document Analysis and Recognition*, pages 560–564, Washington, DC, USA, Aug. 2013. IEEE. ISBN 978-0-7695-4999-6. DOI: 10.1109/ICDAR.2013.117. URL http://ieeexplore.ieee.org/document/6628682/.

A. Kotani, S. Tellex, and J. Tompkin. Generating Handwriting via Decoupled Style Descriptors. In A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, editors, *Computer Vision – ECCV 2020*, volume 12357, pages 764–780. Springer International Publishing, Cham, 2020. ISBN 978-3-030-58609-6 978-3-030-58610-2. DOI: 10.1007/978-3-030-58610-2_45. URL https://link.springer.com/10.1007/978-3-030-58610-2_45.

V. Kulesh, K. Schaffer, I. Sethi, and M. Schwartz. Handwriting Quality Evaluation. volume 2013, pages 157–165, Mar. 2001. ISBN 978-3-540-41767-5. DOI: 10.1007/3-540-44732-6_16.

B. Leupolz-Oebel. *Alphabetisierung in der Zweitschrift Deutsch - Ergebnisse einer Handschriftenuntersuchung arabisch erstalphabetisierter SeiteneinsteigerInnen der Sekundarstufe I [German] (Literacy in the second script of German - results of a handwriting study of Arabic first-literate lateral entrants at lower secondary level).* PhD thesis, Pädagogische Hochschule Freiburg, Freiburg, Apr. 2020.

S. H. Lichtsteiner. Differenzierende Beurteilung der Handschrift – ein Bestandteil der Schreibförderung.

Y. Lili and Y. Zhengwei. Real-time Feedback and Evaluation Algorithm for Children's Digital Writing Practice. In *2021 3rd International Workshop on Artificial Intelligence and Education (WAIE)*, pages 10–16, Nov. 2021. DOI: 10.1109/WAIE54146.2021.00011. URL https://ieeexplore.ieee.org/abstract/document/9743162.

B. H. Limbu, H. Jarodzka, R. Klemke, and M. Specht. Can You Ink While You Blink? Assessing Mental Effort in a Sensor-Based Calligraphy Trainer. *Sensors*, 19(14):3244, Jan. 2019. ISSN 1424-8220. DOI: 10.3390/s19143244. URL https://

www.mdpi.com/1424-8220/19/14/3244. Number: 14 Publisher: Multidisciplinary Digital Publishing Institute.

G. Lindsay. Lined paper: its effects on the legibility and creativity of young children's writing. *British Journal of Educational Psychology*, 53:364–368, May 2011. DOI: 10.1111/j.2044-8279.1983.tb02569.x.

M. Liwicki and H. Bunke. IAM-OnDB - an On-Line English Sentence Database Acquired from Handwritten Text on a Whiteboard. In *Proceedings of the Eighth International Conference on Document Analysis and Recognition*, ICDAR '05, pages 956–961, USA, Aug. 2005. IEEE Computer Society. ISBN 978-0-7695-2420-7. DOI: 10.1109/ICDAR.2005.132. URL https://doi.org/10.1109/ICDAR.2005.132.

E. Loup-Escande, R. Frenoy, G. Poplimont, I. Thouvenin, O. Gapenne, and O. Megalakaki. Contributions of mixed reality in a calligraphy learning task: Effects of supplementary visual feedback and expertise on cognitive load, user experience and gestural performance. *Computers in Human Behavior*, 75, May 2017a. DOI: 10.1016/j.chb.2017.05.006.

E. Loup-Escande, R. Frenoy, G. Poplimont, I. Thouvenin, O. Gapenne, and O. Megalakaki. Contributions of mixed reality in a calligraphy learning task: Effects of supplementary visual feedback and expertise on cognitive load, user experience and gestural performance. *Computers in Human Behavior*, 75:42–49, Oct. 2017b. ISSN 0747-5632. DOI: 10.1016/j.chb.2017.05.006. URL https://www.sciencedirect.com/science/article/pii/S0747563217303187.

B. Marschke. *Handbuch der kontrastiven Alphabetisierung [German] (Manual of contrastive literacy)*. Studien Deutsch als Fremd- und Zweitsprache. Erich Schmidt Verlag GmbH & Co. KG, Berlin, 2022. ISBN 978-3-503-20655-1. DOI: 10.37307/b.978-3-503-20655-1. URL https://link.springer.com/10.37307/b.978-3-503-20655-1.

U.-V. Marti and H. Bunke. The IAM-database: an English sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5(1):39–46, Nov. 2002. ISSN 1433-2833. DOI: 10.1007/s100320200071. URL https://doi.org/10.1007/s100320200071.

F. Minuz, J. Kurvers, K. Schramm, L. Rocca, and R. Naeb. *Literacy and second language learning for the linguistic integration of adult migrants: reference guide*. Education and modern languages. Council of Europe Publishing, Strasbourg, 2022. ISBN 978-92-871-9189-2.

J. Mitchell and M. Fairhurst. Handwriting analysis for application in visuo-motor coordination therapy. Oct. 1992. URL https://www.semanticscholar.org/paper/Handwriting-analysis-for-application-in-visuo-motor-Mitchell-Fairhurst/83111e052f057e1e3688363d21fd28e1b4eba0e3.

A. Morikawa, N. Tsuda, Y. Nomura, and N. Kato. Double Pressure Presentation for Calligraphy Self-training. pages 199–200, Mar. 2018. DOI: 10.1145/3173386.3177010.

M. M. Patchan and C. S. Puranik. Using tablet computers to teach preschool children to write letters: Exploring the impact of extrinsic and intrinsic feedback. *Computers & Education*, 102:128–137, Nov. 2016. ISSN 03601315. DOI: 10.1016/j.compedu.2016.07.007. URL https://linkinghub.elsevier.com/retrieve/pii/S0360131516301439.

S. Piccinin and S. Dal Maso. Promoting Literacy in Adult Second Language Learners: A Systematic Review of Effective Practices. *Languages*, 6(3):127, Sept. 2021. ISSN 2226-471X. DOI: 10.3390/languages6030127. URL https://www.mdpi.com/2226-471X/6/3/127. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

K. Pieper. *Kalligrafie & Handschrift [German] (Calligraphy & Handwriting)*. Christophorus Verlag, Freiburg, Sept. 2014. ISBN 978-3-8388-3532-7. URL https://shop.buchundtoene.com/shop/item/9783838835327/kalligrafie-handschrift-von-katharina-pieper-gebundenes-buch.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection, May 2016. URL http://arxiv.org/abs/1506.02640. arXiv:1506.02640 [cs].

N. Reinken. *Die Grammatik der Handschriften [German] (The grammar of handwriting)*. Universitätsverlag WINTER, Heidelberg, Germany, 2023. ISBN 978-3-8253-8630-6. DOI: 10.33675/2023-82538630. URL https://www.winter-verlag.de/de/detail/978-3-8253-8630-6.

S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, Jan. 2016. URL http://arxiv.org/abs/1506.01497. arXiv:1506.01497 [cs].

S. Rosenblum, P. Weiss, and S. Parush. Product and Process Evaluation of Handwriting Difficulties. *Educational Psychology Review*, 15:41–81, Jan. 2003. DOI: 10.1023/A:1021371425220.

S. Rosenblum, J. A. Margieh, and B. Engel-Yeger. Handwriting features of children with developmental coordination disorder – Results of triangular evaluation. *Research in Developmental Disabilities*, 34(11):4134–4141, Nov. 2013. ISSN 0891-4222. DOI: 10.1016/j.ridd.2013.08.009. URL https://www.sciencedirect.com/science/article/pii/S089142221300351X.