1 2 9 0

UNIVERSIDADE D
COIMBRA

Vasco Araújo Goulart Quaresma

HANDWRITING REHABILITATION USING A HAPTIC JOYSTICK

Dissertation within the scope of Integrated Master's in Biomedical Engineering, Specialization in Biomedical Instrumentation, oriented by Professor António Paulo Mendes Breda Dias Coimbra, Professor João Paulo Morais Ferreira and Professor Manuel Marques Crisóstomo.

October 2021

1 2 9 0

UNIVERSIDADE D
COIMBRA

Vasco Araújo Goulart Quaresma

HANDWRITING REHABILITATION USING A HAPTIC JOYSTICK

Dissertation within the scope of Integrated Master's in Biomedical Engineering, Specialization in Biomedical Instrumentation, oriented by Professor António Paulo Mendes Breda Dias Coimbra, Professor João Paulo Morais Ferreira and Professor Manuel Marques Crisóstomo

October 2021

Agradecimentos

Em primeiro lugar, gostaria de agradecer aos meus orientadores Professor Doutor António Coimbra, Professor Doutor João Ferreira e Professor Doutor Manuel Crisóstomo, pelo valioso acompanhamento neste último ano letivo.

Em segundo lugar, aos meus pais, pelo seu apoio e amor incondicionais que, a 1500 km de distância, sempre acreditaram não só em mim, mas também no meu sucesso académico. Um especial agradecimento à minha mãe, que dispensou não só dias úteis, como também fins de semana e feriados na revisão ortográfica e gramatical desta dissertação.

Ao meu irmão Rodrigo que, em todas as etapas da minha vida, me ensinou e relembrou que a resiliência é um elemento crucial no processo de concretização dos nossos sonhos e objetivos.

À Andreia, minha companheira de vida, por todas as experiências que partilhámos e que continuamos a partilhar, pelos bons e maus momentos, por tudo.

Por último, mas não menos importante, a toda a minha família e amigos, por terem tornado tudo isto possível.

Table of contents

List of Figures

Acronyms

AI — Artificial Intelligence

CNN – Convolutional Neural Network

DL — Deep Learning

DTW — Dynamic Time Warping

HOG — Histogram of Oriented Gradients

HOG-SVM — Histogram of Oriented Gradients coupled with a multiclass Support Vector Machine

ML — Machine Learning

OCR — Optical Character Recognition

SVM — Support Vector Machine

Abstract

Writing skills play a major part in communication, whether in a typing or in a handwriting scenario. Thus, a decrease or even loss of these skills can be a major setback. The system described herein aims to promote handwriting rehabilitation through the analysis of characters written using a haptic joystick.

From a clinical perspective, the goal of this project is to encourage the user to write better through repetition. Reinforcement through the obtainment of consecutive good results during a rehabilitation process can be beneficial in the improvement of motor and cognitive skills, thus having an impact on the proprioceptive system.

In this project, there is a character analysis model evolution. Character analysis is performed by digit or letter classification and quantification of written characters quality. As such, different methods are studied and evaluated to achieve the best classification accuracy.

Regarding handwritten characters classification, the pre-eminent procedure for digits consists in the use of Histogram of Oriented Gradients coupled with a multiclass Support Vector Machine (HOG-SVM) whereas for letters the strategy involves using a Convolutional Neural Network (CNN). For handwriting quality quantification, Dynamic Time Warping (DTW) was performed between the written character and ten reference images of the same character.

In sum, the use of these methods enables handwriting analysis and could thus contribute to writing skills rehabilitation through training and evaluation.

Keywords: Handwriting rehabilitation; Haptic joystick; HOG-SVM; CNN; DTW.

Resumo

A escrita desempenha um papel importante na comunicação, quer num cenário de escrita digital quer num cenário de escrita manual. Assim, uma diminuição ou mesmo perda destas competências pode constituir um grande contratempo. O sistema aqui descrito visa promover a reabilitação da escrita através da análise de caracteres escritos, usando um *joystick* háptico.

De um ponto de vista clínico, o objetivo deste projeto é encorajar um utilizador a escrever melhor através de repetição. A obtenção de resultados positivos consecutivos durante um processo de reabilitação pode constituir um reforço positivo e contribuir assim para a melhoria de capacidades motores e cognitivas, melhorando assim o sistema propriocetivo do utilizador.

Nesta dissertação encontra-se a evolução de um modelo de análise de caracteres. A análise é feita através de classificação de números ou de letras e de quantificação da qualidade de caracteres escritos. Para tal, são estudados e avaliados diferentes métodos para a obtenção da melhor precisão de classificação.

Relativamente à classificação de caracteres, o procedimento proeminente para números consiste na utilização de Histograma de Gradientes Orientados acoplado a uma Máquina de Vetores de Suporte multiclasse (HOG-SVM – Histogram of Oriented Gradients-multiclass Support Vector Machine), enquanto no caso de letras recorre-se a uma Rede Neuronal Convolucional (CNN – Convolutional Neural Network). Para a quantificação da qualidade da escrita, foi realizada a Sincronização Temporal Dinâmica (DTW – Dynamic Time Warping) entre o caractere escrito e dez imagens de referência do mesmo caractere.

Em suma, a utilização destes métodos permite realizar uma análise de caracteres escritos e poderá assim contribuir para a reabilitação das capacidades de escrita através do treino e da avaliação das mesmas.

Palavras-Chave: Reabilitação da escrita; *Joystick* háptico; HOG-SVM; CNN; DTW.

1. Introduction

Nowadays, writing is still one of the most important skills to possess, alongside speaking, listening, and reading. Whether in a digital or a traditional way, the ability to write allows the development of thinking skills, the expression of opinions, or even to serve as a proof of identity [1]. Consequently, losing the ability to write is a major drawback that requires tackling as soon as possible.

1.1 Handwriting

The process of creating text with a writing instrument requires a panoply of underlying skills, including complex motor and cognitive skills, like for example fine motor control, in-hand manipulation, visual perception, visual-motor integration, and others [2].

Handwriting is a personal way of graphic communication and constitutes from 31 to 60% of children school days. As such, mastering this task is time consuming and is key for academic success. This task is of vital importance not only for children, as it is also required in the adulthood, as it enables a form of identification through signatures, an essential step in authorizing documents or writing cheques.

1.2 Dysgraphia

Dysgraphia is a disorder that affects writing skills of 5 to 20 percent of all children [3]. The term dysgraphia is usually used to specify this condition during childhood and if it is an acquired lack of writing capabilities in adulthood, the more commonly used term is agraphia [4]. Both forms can be caused by a variety of mechanisms, as the first can be isolated or co-occurring with other disorders such as dyslexia [5]. The latter, agraphia or acquired dysgraphia, can be a consequence of brain damage or loss of motor function. Given the diversity of causes, dysgraphia can be classified in different types, such as lexical (irregular sound-to-letter patterns generate misspellings), phonological (writing and spelling disturbances), mixed, and others [6].

1.3 Stroke

In adults, stroke is the second largest cause of death and the number one source of long-term severe adult disability worldwide [7]. A large proportion of strokes interferes with middle cerebral artery blood supply, consequently interfering with the motor cortex, which is responsible, among other activities, for handwriting. Given the human and economic burden of stroke, post-stroke care is of extreme importance.

1.4 Objectives

The developed system aims to promote handwriting rehabilitation through an analysis of writing skills of people who had their movements or fine motor skills limited (e.g., a stroke) and could also be applied in other cases, such as dysgraphia. This way, the person in need of therapy can write any character and the system can classify what was written and evaluate how well it was written. The pursuit of better results during a rehabilitation process could make the user write more, thus improving proprioceptive skills.

A haptic joystick was chosen for this purpose since it can provide live feedback by applying force to the user’s hand, thus aiding in the process of handwriting training.

Previously developed software [8] allowed the use of the haptic joystick as a writing guiding tool. It was used to serve as an interface between the haptic joystick and the implemented functionalities. Consequently, previous features make the joystick useful for training, while the system described herein can be used to assess the writing skills. The combination of both gives the user the option to train or evaluate the writing skills.

The implemented functionalities in this project consist in automatic digit or letter classification and quantification of written characters quality, making use of the National Institute of Standards and Technology Special Database 19 [9]. The general framework of this project can be seen in a flowchart form in Chapter 3.

For the classification part, i.e., the identification of the written character, the methods evaluated were Convolutional Neural Network (CNN), Optical Character Recognition (OCR) and Histogram of Oriented Gradients coupled with a multiclass Support Vector Machine (HOG-SVM).

For the quality quantification part, the method assessed was Dynamic Time Warping (DTW).

This way, the system described herein could be a complement in post-stroke therapy, enabling handwriting training through repetition and could also intervene in school dysgraphia diagnostics as a handwriting performance tool.

In Chapter 2 there is a literature review of the topics mentioned herein. The materials and methods used and assessed in this project are present in Chapter 3. This assessment and method iteration are present in Chapter 4. In Chapter 5, the results are shown and discussed. The conclusions and future work proposals are present in Chapter 6.

2. Literature review

The study of the hand is a widely sought-after field of research, mainly because most of daily activities rely on this body part. The complexity of the hand and its vast number of medical conditions make hand rehabilitation a field of study with plenty of developed work.

2.1 Post-stroke hand rehabilitation

As hand complications can be caused by different reasons, such as brain damage or loss of motor function, most cases are handled either by physiotherapy or occupational therapy sessions. As a result of this multitude of causes, a vast number of applications have been designed in this field.

Due to the high and well-known stroke prevalence [10], a lot of the applications that are developed to promote hand rehabilitation mention and target individuals who suffered complications due to this disease. There are tablet applications [11] [12] for this kind of rehabilitation, interactive games that include letter tracing [13], mechanical products (such as Neuroball by Neurofenix [14], pneumatic devices [15] [16], exoskeletons [17]), or others making use of Virtual Reality [18].

Figure 1 - Example applications developed for hand therapy.

Passiveness is a common feature amongst these hand therapy solutions as they do not apply feedback force to the user’s hand. The mentioned applications resolve around fine motor control improvement, undesirable movements restriction or range of motion improvement.

2.2 Dysgraphia treatment

The tuning of mechanical properties (namely increasing viscosity and inertia, or stiffness and weight, respectively) of a robotically assisted pen (see Figure 2) has demonstrated to provoke modifications in children motor commands, thus improving handwriting quality [19].

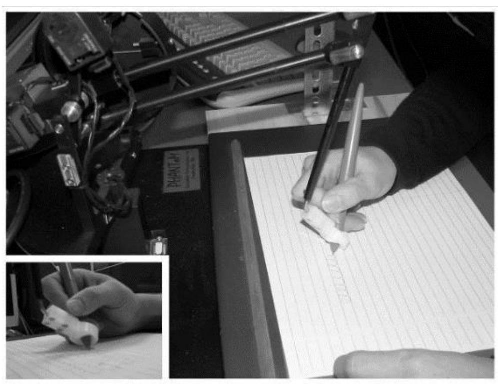


Figure 2 – Robotically assisted pen used in children handwriting evolution assessment.

Considering dysgraphia targeting, robot guided training has been performed in children with motor difficulties. In this study, the system showed to be effective in manual skills training in children of various ages [20]. Although fine motor skills could be an academic attainment predictor, the system was unable to prove writing skills improvement through the robot interaction.

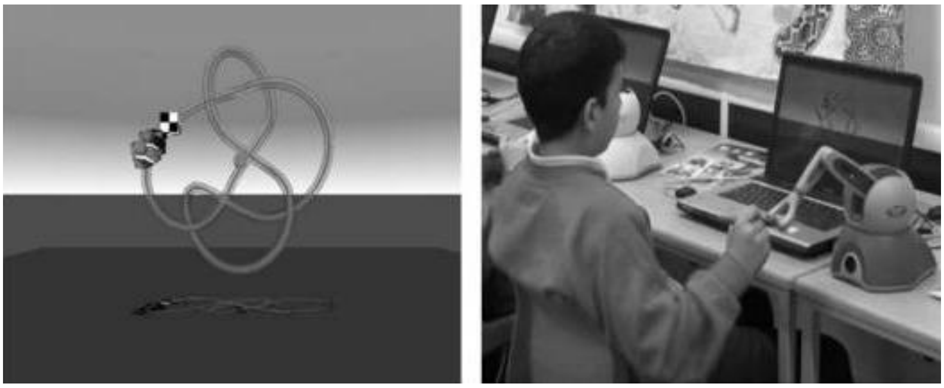


Figure 3 – Robot guided manual skills training.

However, child-robot interaction has proved to be beneficial even in severe dysgraphia cases. In a study composed of occupational therapy that lasted for 20 weeks, the child in question displayed a good engagement to a physical robot that incorporated games and numerous tasks. As a result, the child-robot interaction played a major part in handwriting quality improvement [21].

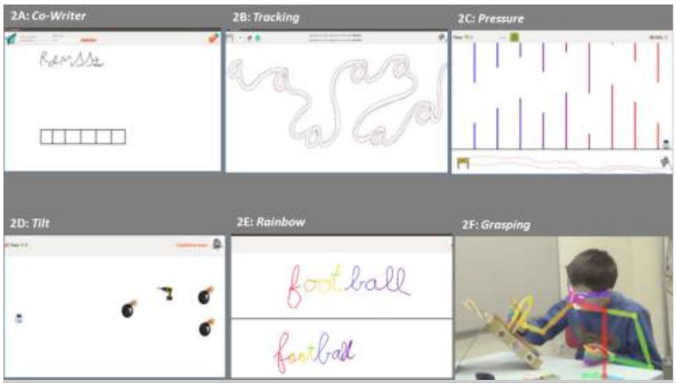


Figure 4 – Child-robot interaction in a severe dysgraphia case.

2.3 Haptic technology

A big proportion of hand complications can affect handwriting, given this task complexity [22]. Handwriting experience, as stated in the previous chapter, can influence learning experience on a broad scale. For example, it has been proven through functional neuroimaging that early handwriting practice can affect communication between the visual and motor systems, thus improving visual symbol recognition and production of visual forms [23], skills which not only assist but also expedite character recognition and elaboration.

Haptics is a type of technology that can create a sense and an experience of touch by the application of forces, vibrations, or motions to the user. Focusing on the use of this technology, haptics has been proven to be beneficial for physical rehabilitation, as setting up an environment with this type of technology can enable a more prolonged therapeutic process with less resource and time restrictions [24]. In fact, sensory feedback (for example, given by a haptic joystick) has been proven beneficial in handwriting learning and in rehabilitation by tuning two important sensory modalities, namely vision and proprioception (defined as the body's ability to sense movement, action, and location) [25].

In the literature, two applications focused on the use of haptic technology to perform handwriting rehabilitation. The first [26] focused on the use of a joystick to guide the user hand through text typed in a field. The major disadvantage of this application consists in the fact that the text is always based in a font (Comic Sans MS), which does not replicate a real handwriting scenario.

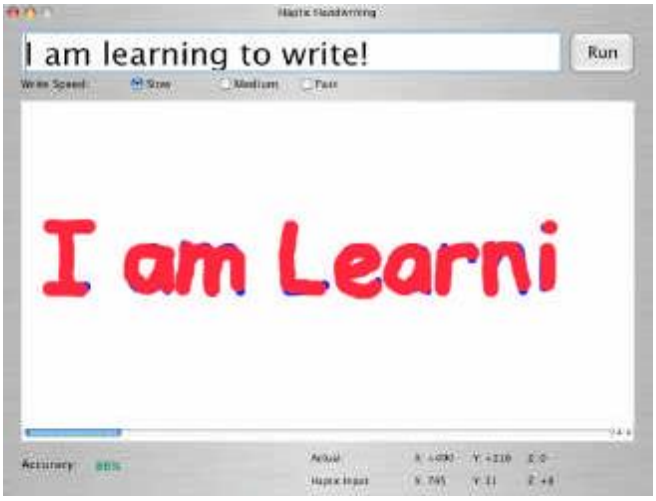


Figure 5 – Application focused on handwriting guidance.

The second application [27] enables both handwriting learning and evaluation, focusing only on alphabetical writing. While promising, this application does not perform handwriting classification and the evaluation is performed through comparison of the user input with only one reference letter, which does not take into account different writing styles.

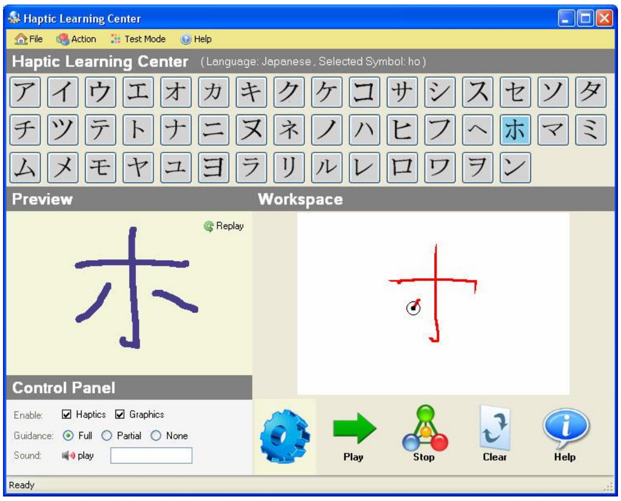


Figure 6 – Application that enables handwriting learning and evaluation.

2.4 Handwriting recognition and evaluation

Regarding handwritten characters recognition, different Machine and Deep Learning methods can be found in the literature. Deep Learning (DL) is a field of Machine Learning (ML), which in turn is included in Artificial Intelligence (AI). While AI is defined as the simulation of human intelligence in machines [28], ML is focused on the use of data and algorithms to implement AI. DL refers to artificial neural networks.

The two methods that appear more frequently in the field of character recognition are Optical Character Recognition (OCR) [29] [30] [31] and Convolutional Neural Networks (CNN) [32]. Histogram of Oriented Gradients coupled with a multiclass Support Vector Machine (HOG-SVM) has also been used in digit recognition [33] and has been proven advantageous in this field.

Despite the numerous developed works in the area of DL, CNN still figures as the state-of-art contribution in the field of character recognition [34] [35].

In relation to handwriting quality evaluation, most of the methods reported in the literature don’t resort to ML. There is no standard digital procedure of recognizing and evaluating handwritten characters, as this is more commonly done *in situ* in sessions by a graphologist [36], in school environments by teachers or by occupational therapists. However, Dynamic Time Warping (DTW), a method commonly used in speech recognition, has been deemed and proven promising to assess handwriting, both in multimedia tools [27] and on graphical tablets [37].

Taking into consideration every aspect addressed in this chapter, the use of a haptic joystick to perform handwriting analysis through classification and evaluation (making use of the pre-eminent methods reported in the literature) deems promising. Plus, as there are no handwriting rehabilitation sessions per se, the developed system could have a positive outcome in complementing hand rehabilitation sessions not only in post-stroke but also in dysgraphia therapeutic aspects.

3. Materials and methods

In order to have a system capable of performing handwriting analysis, several materials and methods were used. The framework of this project can be seen in a form of flowchart in Figure 7.

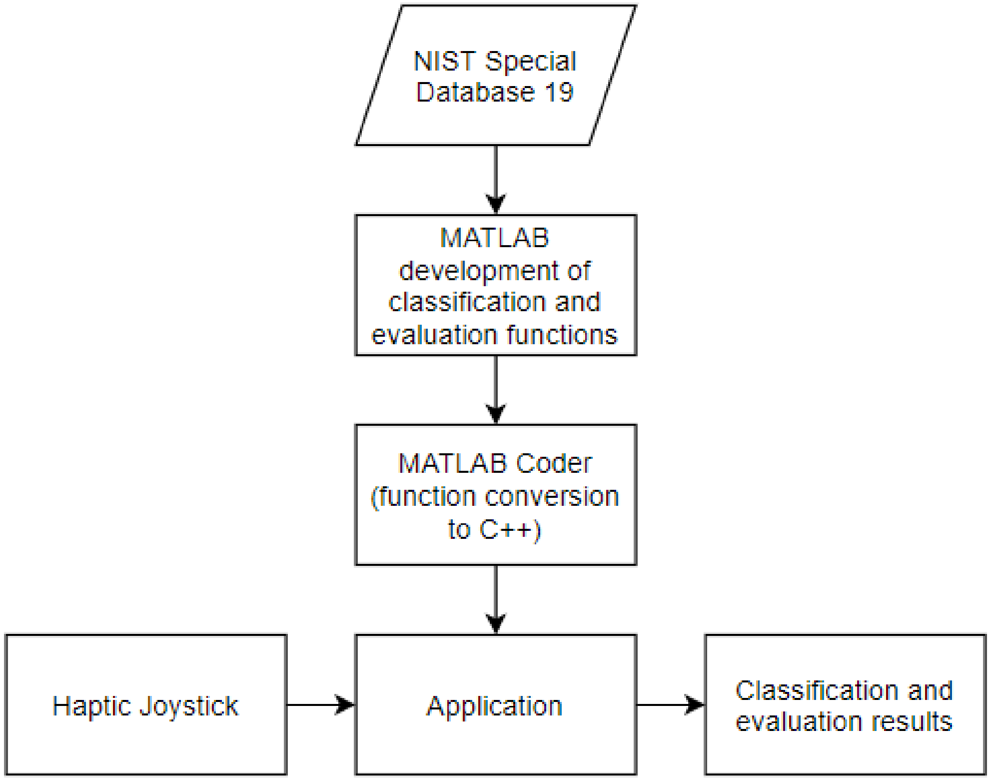


Figure 7 - Flowchart of the underlying procedure of handwriting analysis implementation.

The MATLAB functions mentioned in the figure above made use of different methods. The choice of the methods to evaluate was based on the most commonly used methods found in the literature. To build a character analysis model, a kind of experimental study was performed, where the methods were evaluated in the classification of collected real handwriting (explained in more detail in Chapter 4).

3.1 Materials

Concerning materials, a dataset was used to implement the methods described in more detail in a subsequent section of this chapter. The experiments reported in Chapter 4 were performed on an Intel(R) Core (TM) i3-4010U CPU @ 1.70GHz, with 4 GB of RAM and a NVIDIA GeForce 820M GPU. A haptic joystick was used as a writing tool.

3.1.1 Dataset

The dataset used to apply the methods to was the National Institute of Standards and Technology (NIST) Special Database 19 [9] and its variations (MNIST [38] and EMNIST [39]), which contain over 800,000 character images. The images have a size of 28 by 28 pixels.

MNIST dataset is split into training and testing sets, which contain 60,000 and 10,000 images of digits, respectively. This dataset does not contain an equal number of samples per class, i.e., contains a different number of images for each digit.

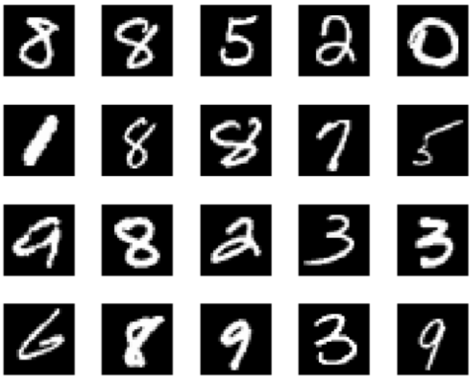


Figure 8 – MNIST dataset example images.

EMNIST dataset contains six different subsets. The ones that were used in this project were EMNIST Digits and EMNIST Letters, both of which have an equal number of samples per class.

EMNIST Digits is composed of 280,000 characters (24,000 in the training set and 4,000 in the testing set for each digit), and EMNIST Letters has 145,600 characters (4,800 make for the training set and 800 the testing set for each letter). EMNIST Letters contains 26 different classes, one for each letter, as uppercase and lowercase versions of the same letter make for a single class.

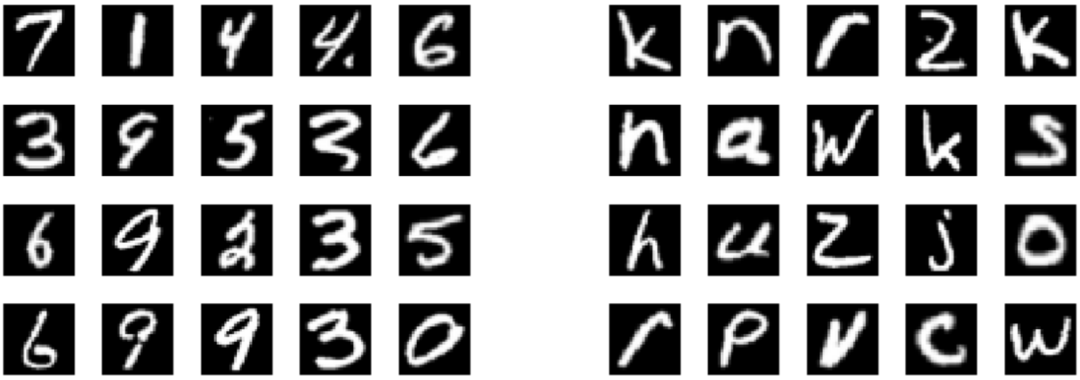


Figure 9 – EMNIST dataset examples images: EMNIST Digits (left) and EMNIST Letters (right).

3.1.2 Haptic joystick

The haptic device used as a tool for handwriting acquisition was the joystick Touch by 3D Systems – See Figure 10.

Figure 10 - Haptic joystick Touch by 3D Systems.

A writing platform [40] was used to give the joystick a more user-friendly experience, as it can better mimic traditional writing. The use of this platform positions the base of the joystick 52 mm below the platform, allowing the tip of the joystick pen to touch the writing plane.

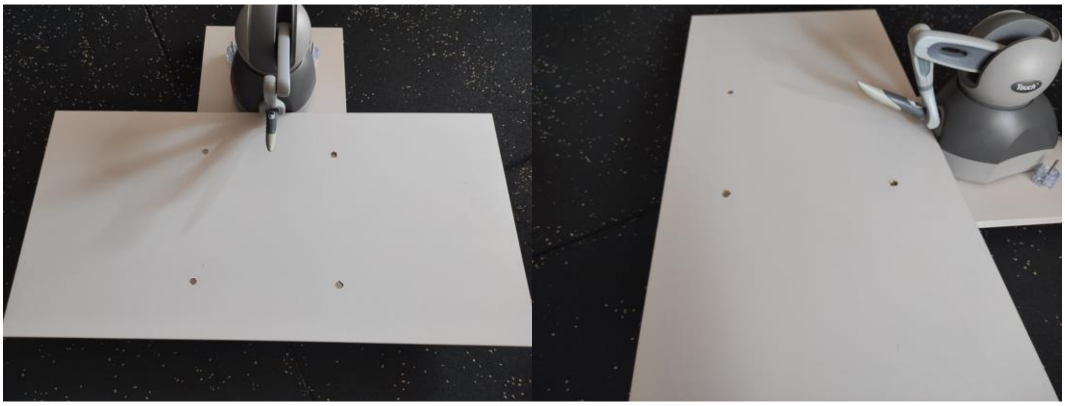


Figure 11 – Writing platform used in conjunction with the haptic joystick.

3.2 Methods

The methods used in this project to perform character classification and quantification of written characters quality made use of Machine and Deep Learning and Computer Vision:

- Convolutional Neural Network (CNN)
- Optical Character Recognition (OCR)
- Histogram of Oriented Gradients (HOG)
- Support Vector Machine (SVM)
- Dynamic Time Warping (DTW)

As this was the first interaction with these fields of study and respective algorithms, difficulties arose, and further inspection and study was performed.

A Convolutional Neural Network (CNN) is a subclass of neural networks [41] that contains one or more layers of convolution units. Optical Character Recognition (OCR) is a method where recognized characters from an image are transformed into editable text. Histogram of Oriented Gradients (HOG) is a feature descriptor used to extract features from image data. Support Vector Machine (SVM) is a Machine Learning algorithm, used to analyse data for classification and regression analysis. This method is used herein as a non-binary linear classifier to classify HOG features [42]. DTW is a method to optimally align, i.e., to minimize the sum of distances between respective points, of two given signals [43]. In this project, DTW is performed between images instead of signals, so that the sum of distances between respective pixels is minimal. As such, the comparison between two images using this algorithm returns a sum of distances, evaluated herein to perform handwriting analysis.

Automatic written character classification was developed using MATLAB R2019b. As mentioned in Chapter 1, a haptic system interface [8] was used to perform linkage between the haptic device and the implemented functionalities. The use of this interface required examination and investigation, given the fact that this was the first contact both with the coding language and the interface itself.

As the interface was written in C++, MATLAB Coder application was used to convert from one programming language to the other. This way, it became possible to include the developed methods in the interface, making use of Visual Studio 2017, Qt 5.14.2, and Qt Visual Studio Tools 2.7.1.

To be able to generate and run the converted CNN code in C++, Intel Math Kernel Library for Deep Neural Networks (Intel MKL-DNN) was built and used, according to MathWorks build instructions [44].

For character quality quantification, MATLAB R2021b was used, since R2019b did not support code generation to C++ for the used methods.

CNNs described in this project were trained using MATLAB, making use of the Deep Learning Toolbox. The general architecture and the procedures were the same as those found in a MathWorks example [45].

The layers of the CNNs consist of three consecutive sets composed of a 2-D convolutional layer, a batch normalization layer, and a rectified linear unit layer, followed by a max pooling layer – see Figure 12.

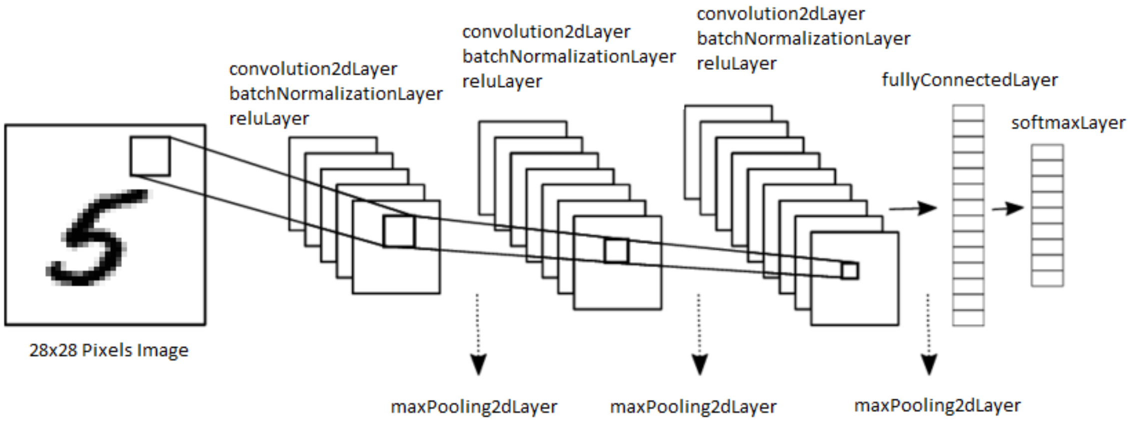


Figure 12 - Convolutional Neural Network architecture used. Image adapted from [46].

The convolutional layers have, from left to right, 8, 16 and 32 filters of size [3x3]. All three of the convolutional layers have “SAME” padding (layer output the same size as the layer input size). The max pooling layers have pooling regions of [2x2] and a stride of [2x2].

The images used in the example are subdivided in ten folders, each containing 1000 examples of the same digit, thus making a total of 10000 images. They can be found inside MATLAB root folder and are made available upon installation of the Deep Learning Toolbox. The images are 28 by 28 pixels.

Figure 13 - Sample images used in MathWorks example.

As there are no originally defined validation/test images, the authors of the example split the 1000 images in two sets: one containing 750 for training and the other with 250 for validation/testing.

Images used for validation serve to periodically evaluate the network performance during training and are not used for training nor to update the network weights. Images used for testing purposes are availed at the end of the training process to do accuracy assessments. In the MathWorks example, the images split from the training set are used for both validation and testing, hence the validation/testing designation.

One other used method was composed of HOG coupled with a SVM, which was also based on a MathWorks example [47]. In the example, HOG feature extraction was performed using cells of 4 by 4 pixels. The features, based on the directional changes of colour along the image, are then used to train the multiclass classifier (SVM).

The first networks were developed for digit classification, due to having a smaller number of possible outputs when compared to letter classification, thus making the process simpler. The concept of a single neural network to classify both digits and letters was taken into consideration but was soon after discarded since it could introduce interclass confusion, such as between the number zero and the capital letter O, number eight and capital letter B, and so on.

4.1 Digit classification

There are two differences between the example network and the first developed CNN (named *cnn_digit* for distinction between networks): firstly, the size of the validation set, since in the developed network the proportion between training set size and validation one was 70-30, as it can provide uniformization on network comparison to those found in the literature [48]. This way, there were 700 training images and 300 test images for each digit. Secondly, the images used for *cnn_digit* training were the complement (see Figure 14) of those used in the given example, i.e., originally white trace on black background became the opposite.

Figure 14 – Sample image from MathWorks example (left) and its complement (right), used for training *cnn_digit*.

This step was performed simply by the fact that, in most case scenarios, writing is performed on a white surface, as paper, and it seemed more logical to have a network trained with this kind of images.

It is important to note that, despite the differences between the network from the example and *cnn_digit*, both had a similar validation accuracy value (99.88% and 98.93%, respectively).

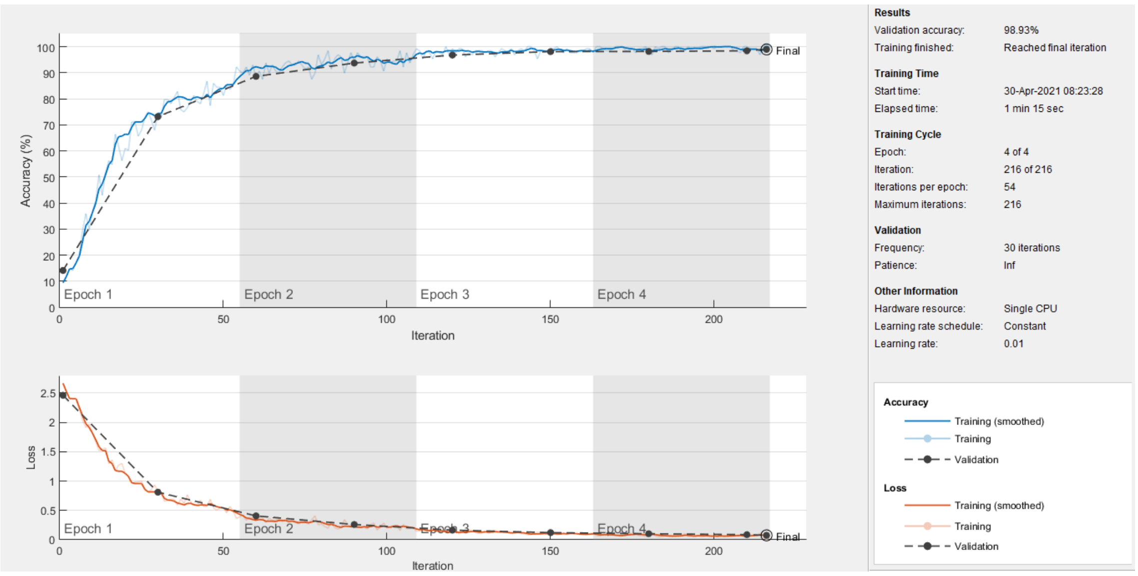


Figure 15 – Training process sample image, in this case *cnn_digit*.

The second network, *cnn_digit_bin*, differs, as the name suggests, in the binarization of the previously mentioned images. This process transforms a grayscale image (with values ranging from 0 to 255) in its binarized form. This was performed because the acquisition mode in the interface software consists in ones and zeros, which is precisely what a binarized image is made up of.

Figure 16 - Sample image used to train *cnn_digit* (left) and its binarized form (right), used for training *cnn_digit_bin*.

The validation accuracy was slightly lower than the previous, lowering from 98.93% to 97.33%.

The third trained network, *cnn_digit_rot*, evaluated the influence of data augmentation on validation accuracy. For that purpose, the original white background images used for *cnn_digit* training were rotated by +10° and -10°, thus tripling the number of used images. The accuracy was slightly lower than in *cnn_digit*, leading to a value of 99.80%.

Figure 17 - Sample image used to train *cnn_digit* (left) and its rotated forms (centre and right), used for training *cnn_digit_rot*.

Table 1 - Validation/test accuracies for the three CNNs – *cnn_digit*, *cnn_digit_bin* and *cnn_digit_rot*.

Network name	Validation/test accuracy
cnn_digit	99.88%
cnn_digit_bin	98.93%
cnn_digit_rot	99.80%

To evaluate whether the validation/test images were only being used to the purpose of validation and/or testing and not for training, 50 images were manually set apart from the rest of the dataset. Afterwards, a CNN was trained with the remaining 950 images. The classification of the 50 images achieved similar accuracies of those obtained in Table 1, thus attesting the accuracy values and the image splitting.

Although the values in Table 1 are on par with literature values [49], it is important to note that the dataset used to train *cnn_digit*, *cnn_digit_bin* and *cnn_digit_rot* is a synthetic dataset, i.e., composed of computer-generated images. Therefore, validation/test accuracy values only take into consideration the specific type of images present in the dataset.

To analyse the performance of the CNNs in classifying handwriting, eight subjects (four males, four females, ages 14, 28, 29, 44, 52, 56, 74 and 80, seven of which right-handed and one left-handed) were asked to write the numbers from zero to nine in a white blank sheet of paper (see Figure 18). The sheet of paper was then scanned in grayscale and the digits were cropped, giving a total of 80 images, and resized into 28 by 28 pixels so that they could be classified by the CNNs.

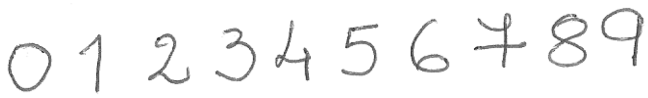


Figure 18 – Handwritten digits sample.

The classification results can be seen in Figure 19 in the form of confusion charts and in percentage form in Table 2.

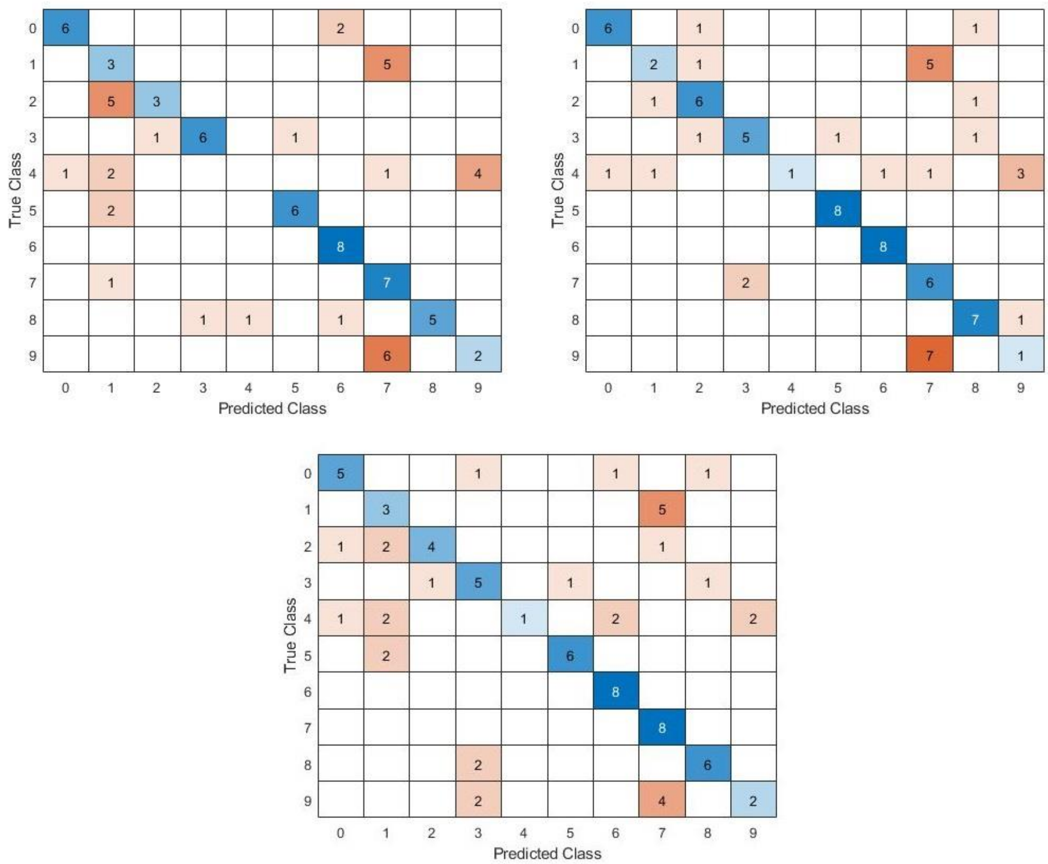


Figure 19 – Confusion charts for handwritten digits classification by *cnn_digit* (top left), *cnn_digit_bin* (top right) and *cnn_digit_rot* (bottom).

Table 2 – Handwritten digits classification accuracies for each CNN.

Network name	Test accuracy
cnn_digit	57.50%
cnn_digit_bin	62.50%
cnn_digit_rot	60.00%

It is noticeable that the validation/test accuracies present in Table 1 are very different from the test values in Table 2. While the CNNs seem to successfully classify the dataset images, the same doesn't apply in a real handwriting scenario.

Another CNN was trained using MNIST dataset, with the same network architecture and training options as *cnn_digit*, to evaluate if the Table 2 low accuracies were due to the fact that the dataset used was computer generated. MNIST was created using real handwriting and therefore is not a synthetic set.

Analogously, the resulting CNN was used to classify the same images. The accuracy of this classification was 72.50%, and although it was higher than the ones present in Table 2, it was still very low when compared to Table 1 values.

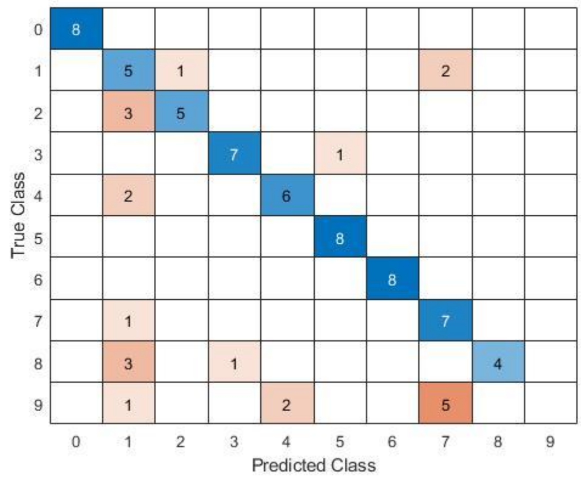


Figure 20 – Confusion chart for handwritten digits classification using a CNN trained with MNIST dataset.

As the CNN achieved lower classification accuracy than intended, two other methods were evaluated. Firstly, Optical Character Recognition (OCR) was applied to Figure 18 (using MATLAB command *ocr* from the Computer Vision Toolbox) to have general insight on how the method would perform. The results (see Figure 21) were composed of twenty characters instead of 10. OCR gave the impression of being a more suitable method for computer generated text rather than handwritten characters. Furthermore, OCR did not detect any text when trying to recognize single characters and thus it was ineffective for handwritten digit recognition.

```
ocrText with properties:

                      Text: 'O' ! Z3>115Q,if48C?↵↵'
    CharacterBoundingBoxes: [20×4 double]
```

Figure 21 – Digit recognition MATLAB output using Optical Character Recognition (OCR).

The second method was Histogram of Oriented Gradients coupled with a Support Vector Machine (HOG-SVM), as described in a MathWorks example [47]. To employ this method, MNIST dataset was used, since it achieved better results in CNN tests. The results were far better than the previous two, and 91.25% accuracy was achieved.

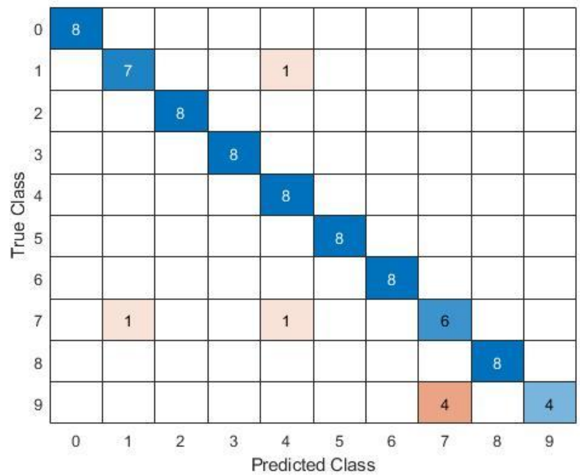


Figure 22 – Confusion chart for handwritten digit classification with Histogram of Oriented Gradients coupled with Support Vector Machine (HOG-SVM), using MNIST dataset.

Another evaluated method consisted in the conjugation of both CNN and HOG-SVM. In this scenario, the images were classified by both methods: if the classification of the two had an equal label output, then this label was the final one. Otherwise, the method that achieved a higher classification score dictated the label of the image.

The complication of conjugating the two methods is that they give different score values: CNN gives a probability of a certain class being the predicted one, while SVM returns negated average binary losses. Even normalizing these binary losses to between zero and one to match CNN scores, SVM achieved a higher score in all the 80 images, even on misclassified ones, thus limiting the use of the method combination.

4.2 Letter classification

Regarding letter classification, there was a similar strategic approach. The same eight subjects that were asked to write the numbers were asked to write the modern English alphabet, both in uppercase and lowercase (see example in Figure 23). The letters were individually cropped, leading to 52 images for each subject, giving a total of 416.

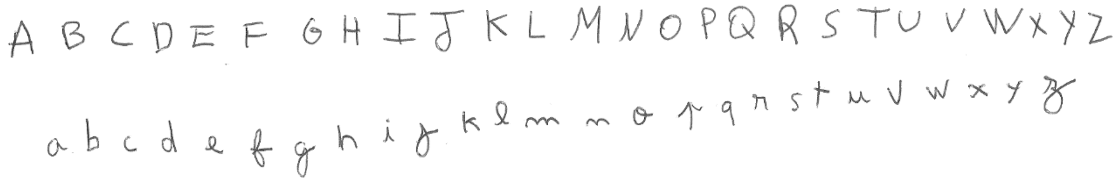


Figure 23 - Handwritten letters sample.

Firstly, a CNN was used to classify the 416 images. From this point onward, only the binarized form of the images was classified since the acquisition mode by the joystick interface is made in ones and zeros, as previously stated. The CNN was trained with EMNIST Letters dataset, and the classification had an accuracy of 75.72% (See Figure 24 or ANNEX 1). To improve readability, the next two figures are replicated in the annexes in a larger size.

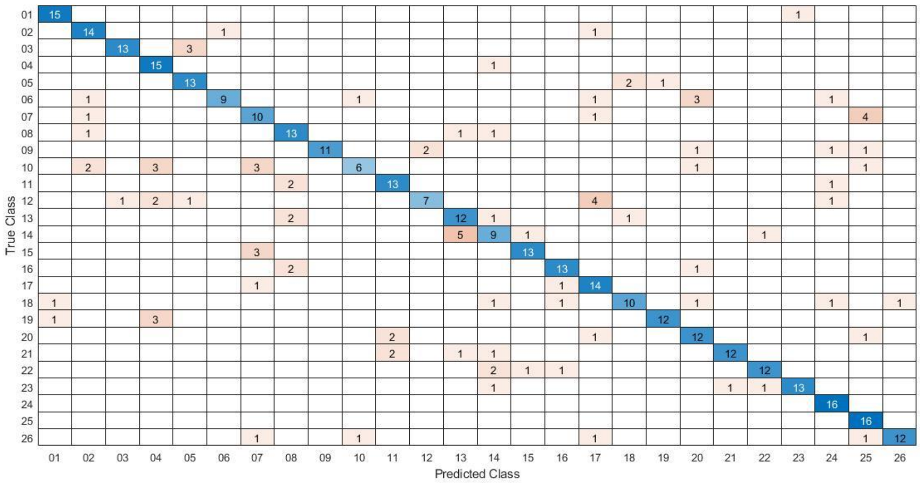


Figure 24 – Confusion chart for handwritten letter classification by a Convolutional Neural Network (CNN). The class names represent the index of the letter in the alphabet.

As OCR achieved similar results for letter classification as those present in Figure 21 for the digit case, this method was considered unfit for this type of data and the next step involved employing HOG-SVM method to classify the same letter images, using the same dataset as the CNN. HOG-SVM achieved an accuracy of 73.32% (Figure 25, replicated in ANNEX 2). The low accuracy obtained can be due to the fact that there are many more possible output classes when compared to digit classification. Furthermore, the existence of both uppercase and lowercase letters in the dataset increases intraclass variation and could thus decrease SVM classification accuracy.

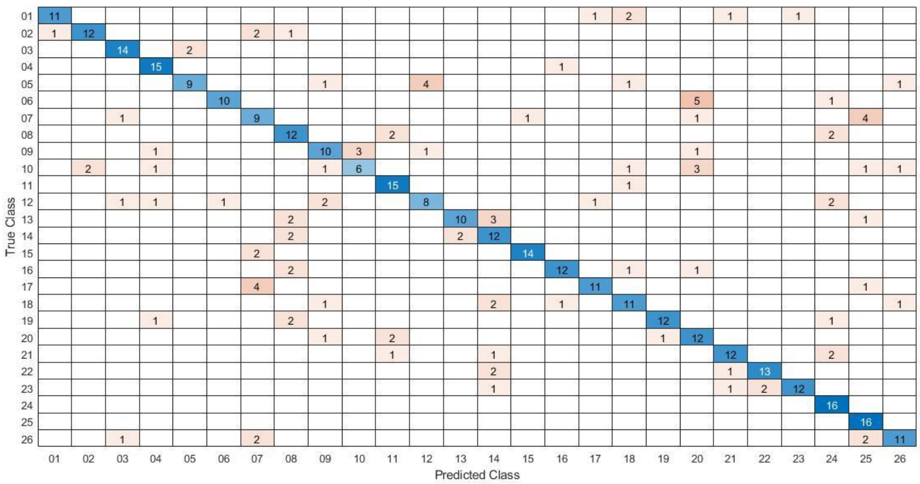


Figure 25 - Confusion chart for handwritten letter classification with Histogram of Oriented Gradients and Support Vector Machine (HOG-SVM). The class names represent the index of the letter in the alphabet.

As in the case of handwritten digits, and to try to improve classification accuracy, another method of combining a CNN and HOG-SVM was evaluated. A CNN was trained for the labels that had more HOG-SVM misclassifications. The labels that were included were the ones that had more than three misclassified labels (confusion chart predicted class column), giving a total of twelve labels. When HOG-SVM predicted one of these twelve labels, the CNN would classify the image. Despite achieving high accuracy values, the method conjugation would take too much into account the eight collected subjects' handwriting and could be a case of overfitting the method, which could cause a decrease in classification accuracy of other handwriting styles.

4.3 Digit and letter classification considerations

Considering every previous character classification accuracy, MNIST seems to be underperforming when compared to EMNIST Letters. Notwithstanding the fact that one dataset contains digit images and the other letters, EMNIST Letters achieved better accuracies with more than double the possible output classes. The two main differences between the datasets are the number of images and equality or not of the number of samples per class. To verify if a dataset with more images and equal number of samples per class would perform better, tests using a CNN or HOG-SVM that were made with MNIST were performed with EMNIST Digits.

The results were as follows: classification with only a CNN had an accuracy of 92.50%, with HOG-SVM an accuracy of 97.50%.

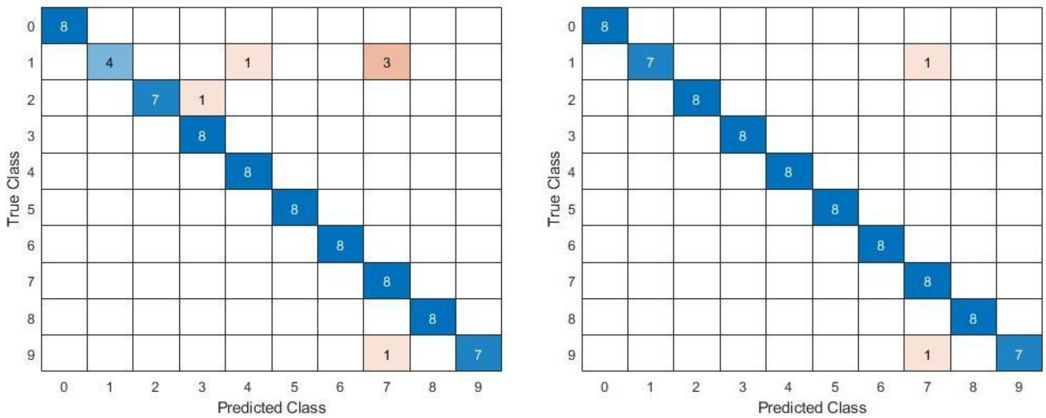


Figure 26 – Confusion charts for handwritten digits classification by different methods using EMNIST Digits dataset: CNN (left) and HOG-SVM (right).

Even though MNIST CNN and HOG-SVM were trained with grayscale images and EMNIST Digits with binarized images, EMNIST Digits proved to be superior (See Table 3), whether because of the balanced number of images per class or due to the substantially larger number of images.

Table 3 - Handwritten digits classification accuracy comparison between MNIST and EMNIST Digits with the use of different methods.

	CNN	HOG-SVM
MNIST	72.50%	91.25%
EMNIST Digits	92.50%	97.50%

Taking into consideration Table 3 results, EMNIST Digits was the chosen dataset to perform digit classification, replacing for this matter MNIST dataset.

Table 4 – Summary of handwritten letters classification accuracies.

	Digits		Letters	
	CNN	HOG-SVM	CNN	HOG-SVM
Subjects handwriting test	92.50%	97.50%	75.72%	73.32%
EMNIST testing set	99.34%	98.95%	92.51%	91.49%
Average	95.52%	98.23%	84.12%	82.41%

Both EMNIST Digits and EMNIST Letters were chosen for performing handwritten character classification. To determine what methods to implement using such datasets, an average between the previously obtained accuracies and the ones using the same methods to the respective test sets was performed. Consequently, the pre-eminent methods (as shown in Table 4) that were converted to C++ using MATLAB Coder to be implemented in the interface software were HOG-SVM for digit classification and a CNN for letter classification.

4.4 Classification workflow

In the interface, as the user writes or draws something, the coordinates of the points of the trajectory are stored in vectors. Because of this, the MATLAB functions to convert to C++ take these vectors as inputs. Variations in both x and y coordinates are calculated so that upon transformation to image the relative dimensions are maintained. Afterwards, the coordinates are transformed into a centred 28 by 28 pixels image, with a border of two white pixels around the edges, and what was written or drawn is thickened (example in Figure 27), so that the image resembles more those present in the datasets, as otherwise the resulting image would have a thickness of only one pixel. Thickening is performed by iterating along rows and columns of the image and changing two neighbour pixels to black (intensity equal to one) upon finding a black pixel (see summary flowchart of the process in Figure 28Figure 28). The final image is then classified by HOG-SVM or a CNN, depending on whether it is a digit or a letter, and the label of the classification is the output of the function.

Figure 27 – Image example before (left) and after centring and thickening (right).

It is worth noting that MATLAB uses *categorical* data type to store data with a finite set of discrete categories to provide a more efficient storage and manipulation. This data type emerges, for example, when label extraction is performed by using the folder names in which the names are located (e.g., for training a CNN). As *categorical* is not a valid type for code generation, every *categorical* value was instead converted to *double*.

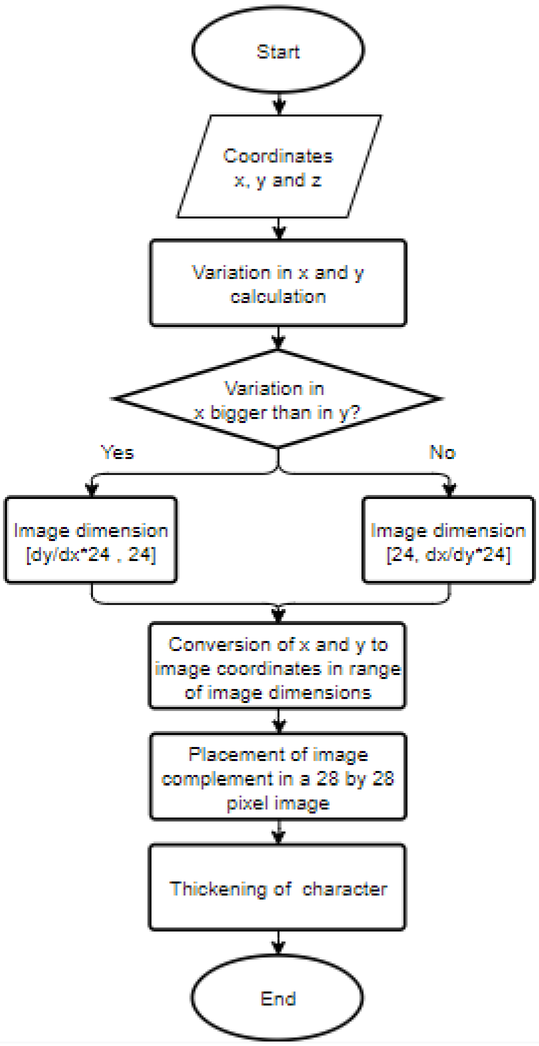


Figure 28 – Flowchart of the process of obtaining an image from coordinates, including centring and thickening.

Important to notice that only variations in coordinates x and y are considered. Variations in z coordinate were initially taken into consideration given the use of the writing platform. Since the z coordinate of the platform is -60, points with a greater value than this one would be deleted and would not be present on the image to classify.

Deleting points with a z value lesser than -60 would remove relevant information (or even leaving an empty vector of coordinates) from the final image and deleting points with a greater value would include unwanted points to a certain extent. However, and adding to this the fact that writing is a very personal task, and everyone has specific characteristics such as tilt angle that not only vary interpersonally but also on the written character, changes in z coordinate were discarded and all the points are considered to perform transformation from coordinates to the final image.

4.5 Character quality quantification

As far as handwriting quality quantification is concerned, the first proposal was to use the score of the HOG-SVM/CNN classification. As the scores are of different nature, as stated before, and this would give an insight more about the method performance rather than the character quality, Dynamic Time Warping (making use of MATLAB Signal Processing Toolbox command *dtw*) was evaluated.

In this project, DTW is performed between the image to evaluate and ten reference images, which were hand-picked from EMNIST datasets. To allow evaluation of both uppercase and lowercase letters separately, a total of 520 letter images were selected as reference images, along 100 digit images. In order to maintain homogeneity between the images, the same procedure of centring and thickening was performed on all of them. Subsequently, the images were converted into text files so that they could be read in the C++ application.

A reference DTW distance table was created to allow obtained DTW distance values comparison and assessment. The EMNIST datasets were subjected to the same procedure of centring and thickening and 100 images from each letter and digit were chosen randomly. For each one of these 100 randomly selected characters from the dataset, DTW was performed with every respective ten reference characters. The minimum of these DTW distances was then calculated. In Table 5, the average, standard deviation and minimum of the minimums of the distances are shown for the case of digits.

Table 5 – Digit reference Dynamic Time Warping distances table built using MATLAB.

Digit	Mean	Standard deviation	Minimum
0	43.34	8.10	26.45
1	20.88	8.60	7.86
2	50.98	7.62	26.41
3	43.04	7.12	28.89
4	41.78	6.18	26.38
5	48.00	9.77	27.75
6	36.57	7.91	21.70
7	34.87	7.40	19.85
8	41.38	7.68	24.29
9	32.97	6.81	16.77

Different numbers of randomly selected images from the dataset were selected. As selecting 1000 images gave similar results when compared with 100, 100 was the selected number of images to minimize computation time. The same procedure was done for EMNIST Letters and the reference letter images. Similar tables for uppercase and lowercase letters can be found in ANNEX 3 and ANNEX 4, respectively.

As different styles of digits and letters were selected to allow characters written in a different manner to not be evaluated as poorly written, comparison between DTW distances was performed using the minimums (right-most column in the above table). Furthermore, the fact that the letters randomly selected from the datasets could be uppercase or lowercase, using average as a metric of comparison could be misleading.

4.6 Methods implementation

With the intention of enabling handwritten classification and evaluation, MATLAB functions were converted to C++ using MATLAB Coder. Upon each conversion, an example main function which declares and initializes data is generated. The functions present in the generated example were modified to fit the system needs, such as the inputs changed to vectors and the inclusion of the number of points of the trajectory as input so that the vectors could have the desired dimensions. As such, three functions, one for each of the methods (CNN, HOG-SVM and DTW), were converted, adapted, and implemented on the interface software, in the MainWindow CPP file.

A total of nine buttons were added to the existing user interface: seven QPushButtons and two QComboBoxes (see user interface layout before and after the changes in ANNEX 5):

- *Train digit* and *Train letter* QPushButtons enable digit or letter classification using HOG-SVM or CNN, respectively.
- QComboBoxes allow selection of the character to be evaluated, one for selection of digits and another for letter selection.
- Two QPushButtons (*Evaluate digit* and *Evaluate letter*) enable character evaluation using DTW, one for each type of character. These buttons take into consideration the previous selection of digit or letter in the QComboBoxes to perform evaluation.
- Two other QPushButtons allow the display of training and evaluation result (character label or DTW distance, respectively, and time elapsed).
- *Clear text* QPushButton allows cleaning the QTextBrowser.

Other than including MKL-DNN library in the project, necessary changes were performed to HapticManager CPP file. For each classification method and for the evaluation method, a member function was created to allow the haptic device to be in the desired state (in this case *WAITING*). Changes were also carried out to the haptic joystick pen buttons *callback* member functions to take into consideration the member functions present in MainWindow CPP file.

To note that the classification methods, when converted using MATLAB Coder, use and define by default custom C++ data types for the respective variables (e.g., a dynamically allocated array of doubles is mapped to *emxArray_real_T*). As such, and to prevent conflict between type definitions of each classification method type in C++, MATLAB code was generated for multiple entry-point functions, one for each method.

The workflow is as follows: the system (in MainWindow CPP file) checks every second if any of the four Boolean variables, responsible for checking what method to perform (digit or letter training or evaluation) are *true*, as they are initialized as *false*. When a user clicks on any of the *Train* or *Evaluate* buttons, HapticManager sets the system state as waiting for user input. While the user is writing, the system is on its busy state. When the user finishes writing (by clicking on the lighter button on the haptic device pen), the respective Boolean variable is set to true, and the corresponding converted MATLAB routine is performed.

5. Results and discussion

Random digits and letters were selected to conduct analysis on the system performance regarding character classification and evaluation.

5.1 Digit classification

Digit classification, making use of HOG-SVM, was performed to the numbers zero, two and four.

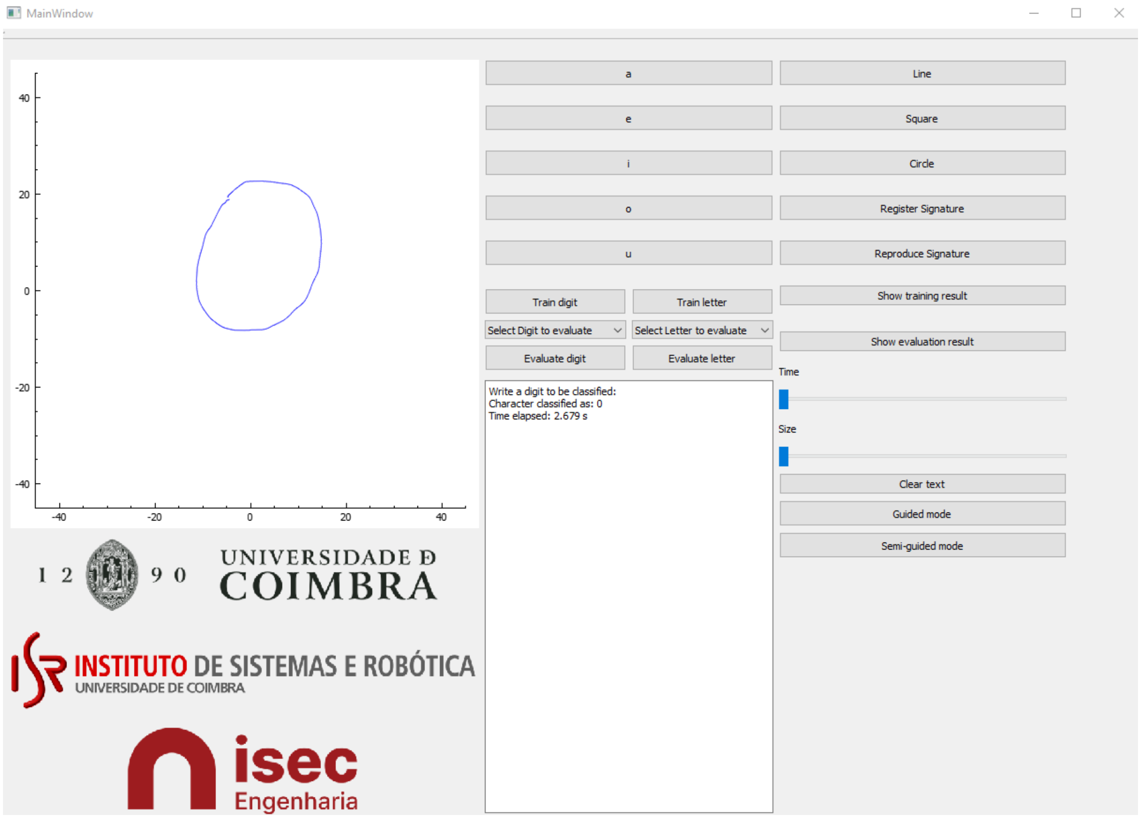


Figure 29 – Interface showing a written digit zero, its classification and time elapsed.

It is observable in the QTextBrowser (text box where the results are displayed) that the number zero was correctly classified. In the next images, only the region of interest (the plot area) is shown, as it was cropped to improve visibility, and the displayed results on QTextBrowser are briefly reported.

After number zero, number two was also correctly classified.

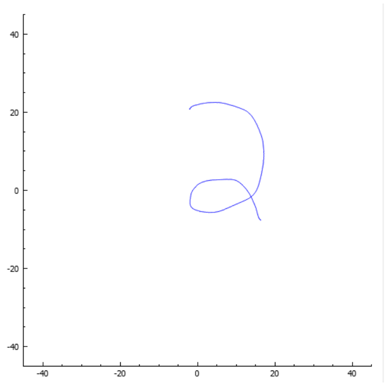


Figure 30 – Plot of number two performed in the user interface.

Number four was the last random digit evaluated. In this case, two different styles were written in order to perceive if the writing style affected the classification. Both cases were classified correctly.

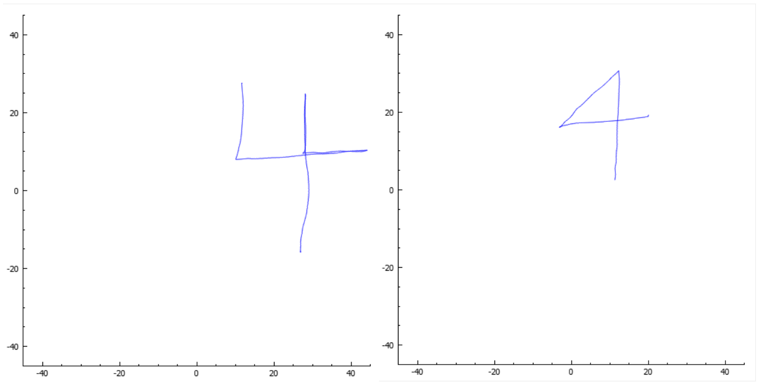


Figure 31 - Plots of two different styles of number four performed in the user interface.

5.2 Letter classification

In regard to letter classification, using a CNN, the first character evaluated was uppercase C, which was also correctly classified.

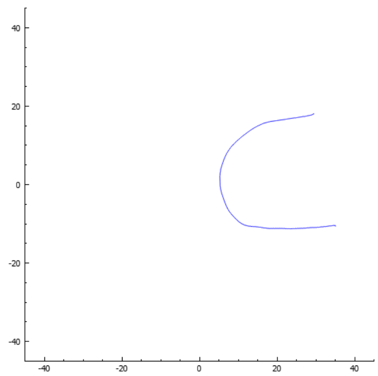


Figure 32 - Plot of capital letter C performed in the user interface.

The second letter evaluated, lowercase l, was wrongly classified as an uppercase D. Its uppercase version, however, was rightly labelled.

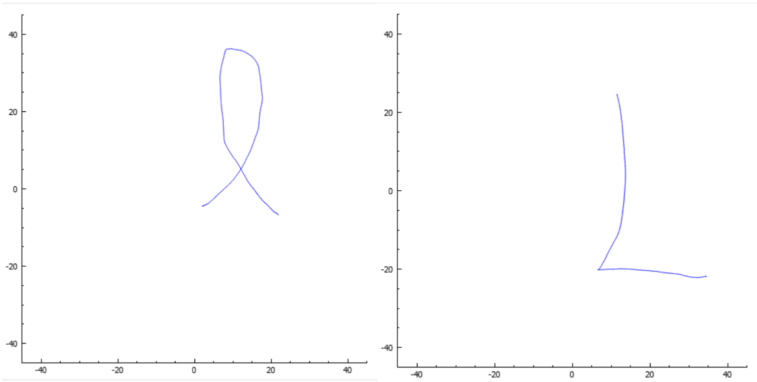


Figure 33 - Plots of lowercase (left) and uppercase (right) letter l performed in the user interface.

The last letter classified was lowercase g, which was also correctly labelled.

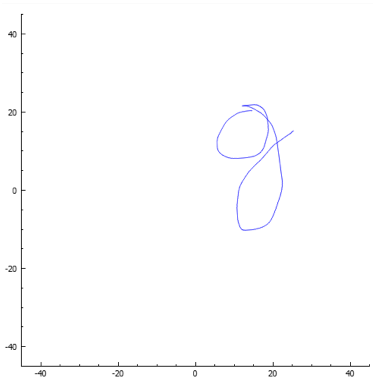


Figure 34 - Plot of lowercase letter g performed in the user interface.

5.3 Digit evaluation

After performing classification tests, digit evaluation using DTW was carried out. For this purpose, two versions (one well and another poorly written) of numbers one, three and five were analysed.

For number one, as well as the remaining characters, DTW was performed between the image and the ten reference images. The output distance is the minimum of the ten distances. The well written version had an output distance of 56.22, while the poorly written version a distance of 69.36. While the first obtained value was lower than the second, the reference value for this digit present in Table 5 is 7.86.

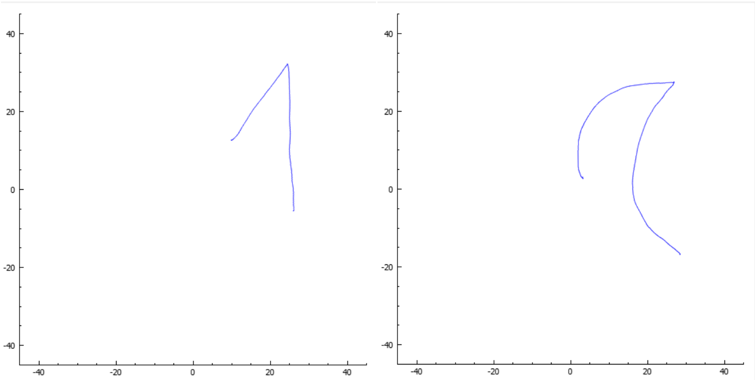


Figure 35 - Plots of a well written (left) and a poorly written (right) number one performed in the user interface.

For the second evaluation test, a well written number three and a mirrored version (along the vertical axis) were compared. The first distance was 71.54 and the latter 69.22, which contraries the desirable relative values. The reference value for number three present in Table 5 is 28.89.

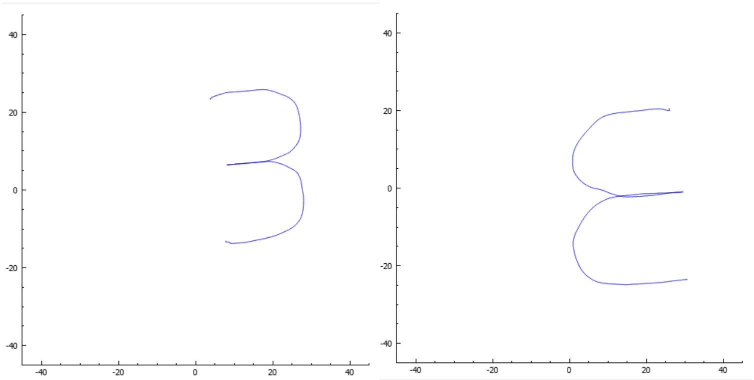


Figure 36 - Plots of a well written (left) and a mirrored (right) number three performed in the user interface.

In the last digit evaluation, a number five and a sinuous version of this digit were taken into consideration. DTW distance of the first was 80.60 and the second 84.99. The relative values are desirable (the first lower than the latter) but very distant from the reference value (27.75).

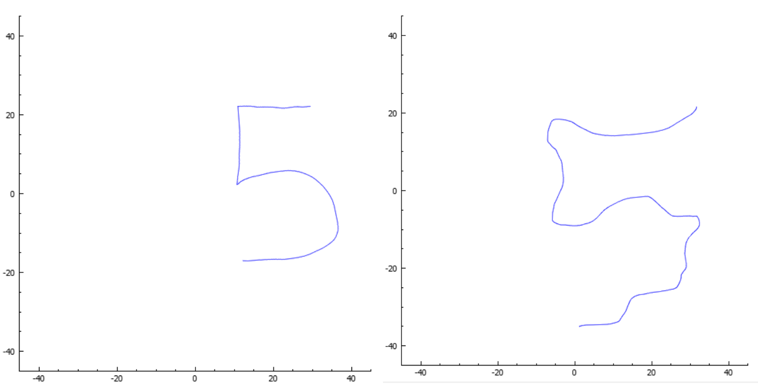


Figure 37 - Plots of a well written (left) and a sinuous (right) number five performed in the user interface.

5.4 Letter evaluation

DTW distances were calculated for the uppercase letters A and P and lowercase t.

The first test was performed with letter A and three and a mirrored version (along the horizontal axis). First obtained distance was 73.59 and the second 76.40. The reference distance is 27.87.

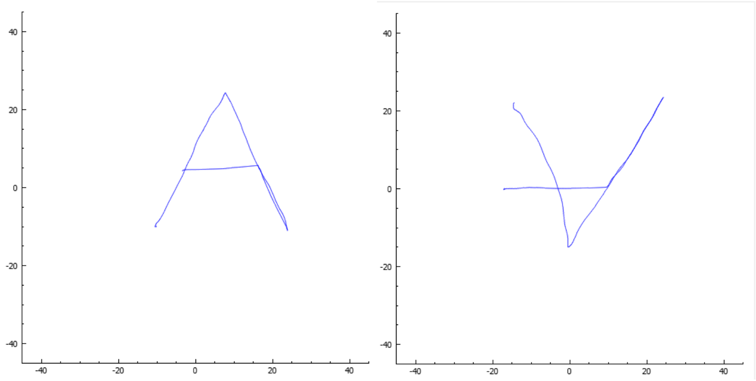


Figure 38 - Plots of a well written (left) and a mirrored (right) capital letter A performed in the user interface.

The next experiment included lowercase t and its uppercase form. Lowercase t evaluation was done in two attempts as the first was not written in a particularly skilfully way. However, the results are still presented since this could give insight of character evaluation performance using DTW. The first attempt achieved a distance of 72.64, the second 65.15. Uppercase T had a distance of 65.76. Reference distance of lowercase t is 21.30.

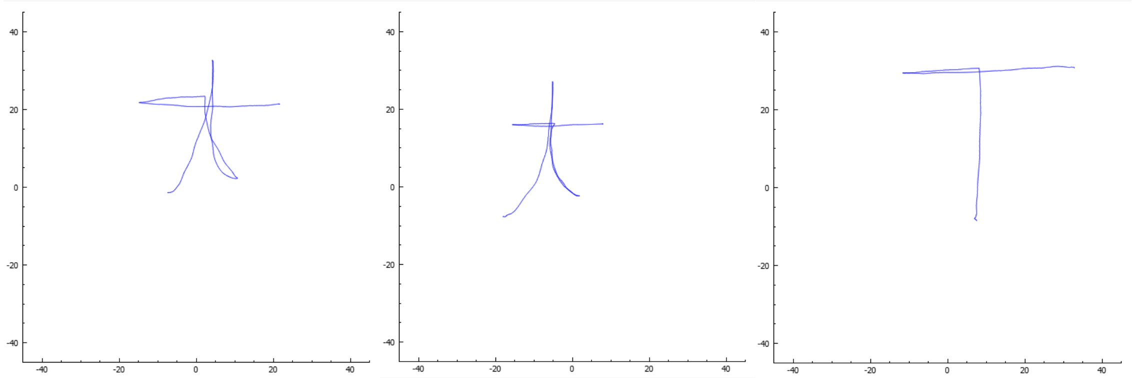


Figure 39 - Plots of an unsatisfactory lowercase t (left), a refined version (middle) and an uppercase type of the same letter (right) performed in the user interface.

The last letter tested was capital P. Evaluation of the good version wielded a result of 65.41, while an inclined version 76.06. The reference value for this letter is 23.39.

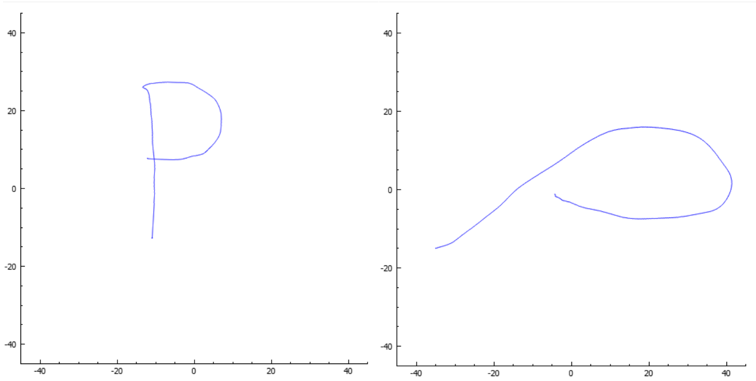


Figure 40 - Plots of a well written (left) and an inclined (right) capital letter P performed in the user interface.

5.5 Discussion

Through interpretation of the previous results, it is perceptible that both digit and letter classification achieved satisfactory results. From the randomly selected digits, all of them were correctly classified and only one of the letters was mislabelled.

In relation to character evaluation, in most of the performed tests the well written version of the character achieved a lower DTW distance than the poorly written version, thus validating DTW as a quality assessment method. The fact that the respective reference distance was far from the obtained ones was also recurrent amongst all the carried-out tests, which makes comparison with the right-most reference tables column erroneous. The distant reference values could be explained due to images of the same character of both EMNIST datasets similarity and the fact that reference images were handpicked from the same dataset.

Although only a limited number of characters were analysed, it is visible that the developed system, despite being improvable, can be used to assess handwriting skills through classifying and evaluating characters.

6. Conclusions and future work

The development of the system enabled learning in numerous new areas and fields, such as Machine and Deep Learning, Computer Vision (and the use of the reported methods), dealing with peripheral devices, using a previously new programming language (C++), managing code conversion, using previous work to help implementation of new functionalities, and adjusting graphical interfaces (using Qt) to fit the desired needs.

In sum, the system achieved the main purpose of this project, which was to perform handwriting analysis through classification and evaluation.

For digit classification purposes, the used dataset was EMNIST Digits and the employed method was HOG-SVM. This method achieved great results, partially because, in most cases, numbers are distinguishable from one another (i.e., have a different gradient orientation) and because there is generally just one way of writing numbers.

Letter classification method made use of EMNIST Letters and consisted in a CNN. The results were also satisfactory but slightly worse than in the digit case. This can be justified by a bigger number of possible outputs and by the fact that the dataset included both uppercase and lowercase versions of each letter in a single class, thus increasing intraclass variation and decreasing the method's accuracy. Training two networks, one for each version of the letter, would limit letter classification to be made separately but could be an advantageous approach in terms of labelling accuracy.

Regarding handwriting evaluation, DTW was used for both digits and letters. In most results of the previous chapter, the well-written version achieved a lower distance than the poorly written one, deeming DTW as a good method to perform this task. However, when considering the values in the built reference tables as comparison metric, the test values diverged from the calculated reference minimums.

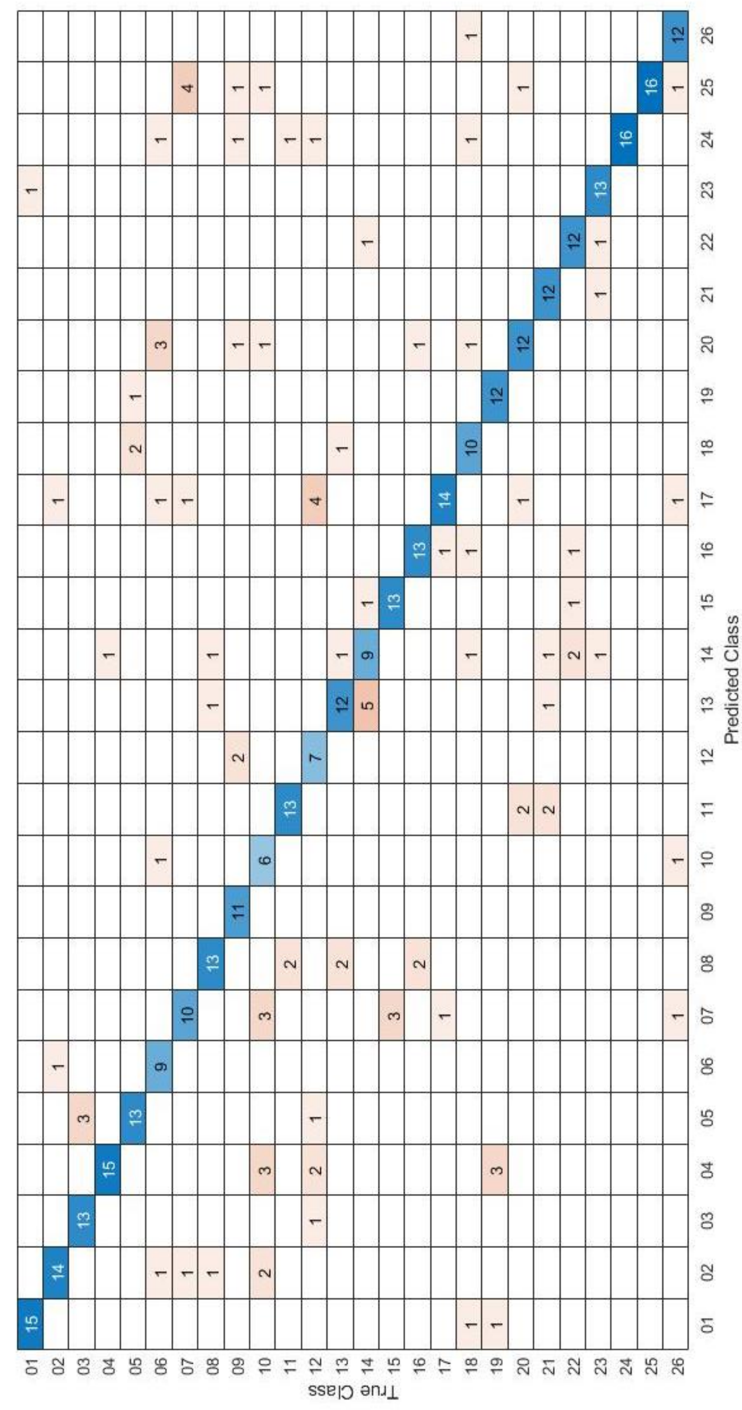
Outside of the intraclass variety imposed by the presence of both uppercase and lowercase letters in EMNIST Letters dataset, the two datasets present a substantial intraclass similarity. Although images obtained from joystick coordinates underwent the same procedure of centring (leaving a border of two pixels) and thickening to minimize the differences between the final image and those present in the dataset, converting vectors of hundreds of points (since coordinates are obtained in a high acquisition rate) to a 28 by 28 pixels image might induce relevant differences to those in the dataset.

This makes comparison between the obtained image and the ten reference images picked from EMNIST datasets different from the comparison between the same reference images and random ones of the dataset (the latter were used to build the respective reference table). Thus, the minimum of the 100 DTW distances carries less value than originally expected. Therefore, comparison should be performed with another parameter (or with reference images created using the haptic joystick) so that the reference table wields more meaningful values and a better evaluation metric.

Future work could not only focus on the previously mentioned aspects, but also on others, such as keeping track of the user evolution over time (e.g., using a database), creating a score measure that considers not only the DTW distance, but also the time taken to write the character (using, for example, reference times for each character). The system, as it stands, only uses haptic functionalities of the device for training lowercase vowels and some geometric shapes. Extending the use of these haptic functionalities to all the characters (e.g., guiding the user through a character upon achieving a low score on the evaluation of the same character) would increase the user independence in the rehabilitation (at the time, this process can only be done if a therapist uses the Register Signature button, and the user follows what the therapist wrote using the Reproduce Signature button).

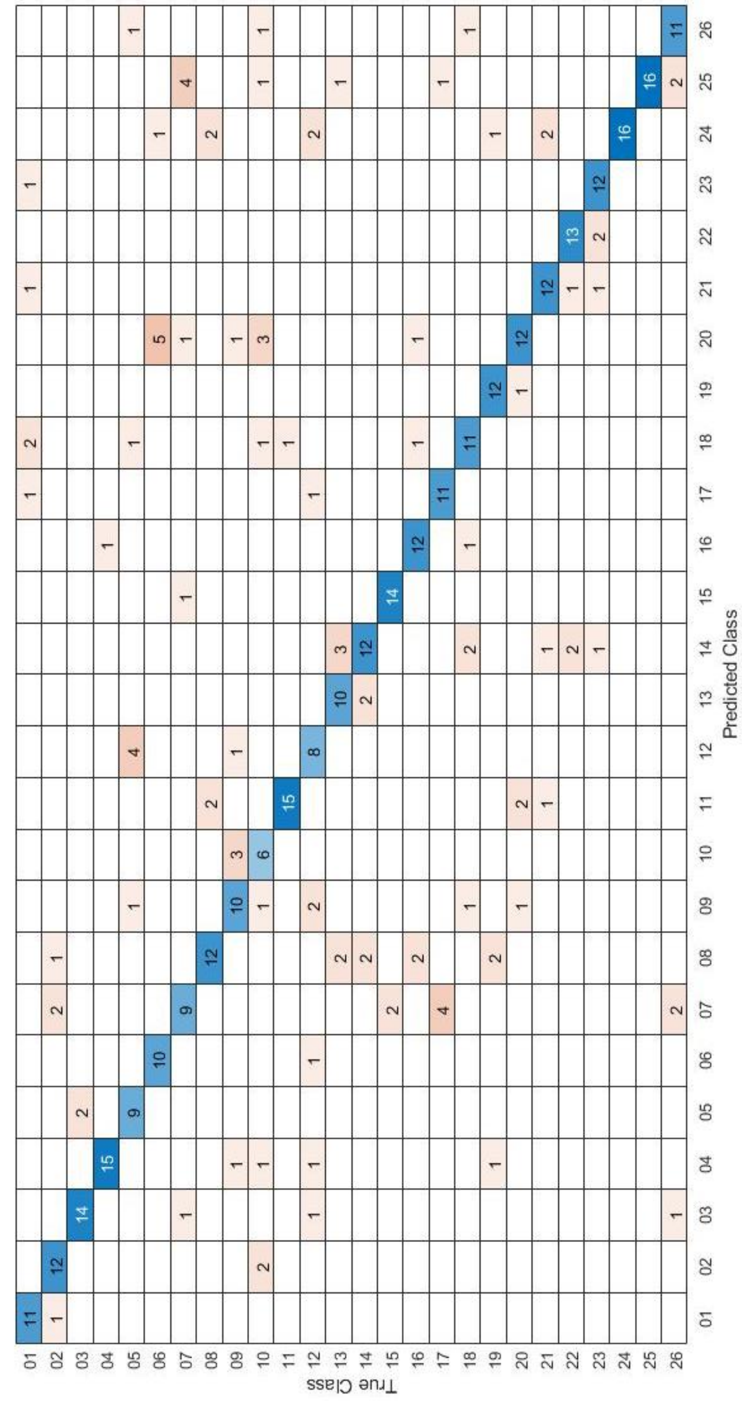
In summary, the system developed herein exhibited positive results in handwriting analysis by classifying and evaluating user input, can already be used to assess handwriting skills, and has proven to be scalable on a handwriting and hand therapy standpoint.

ANNEX 1



Confusion chart for handwritten letter classification by a Convolutional Neural Network (CNN).
The class names represent the index of the letter in the alphabet.

ANNEX 2



Confusion chart for handwritten letter classification with Histogram of Oriented Gradients (HOG) and Support Vector Machine (SVM). The class names represent the index of the letter in the alphabet.

ANNEX 3

Uppercase letters reference Dynamic Time Warping distances table built using MATLAB.

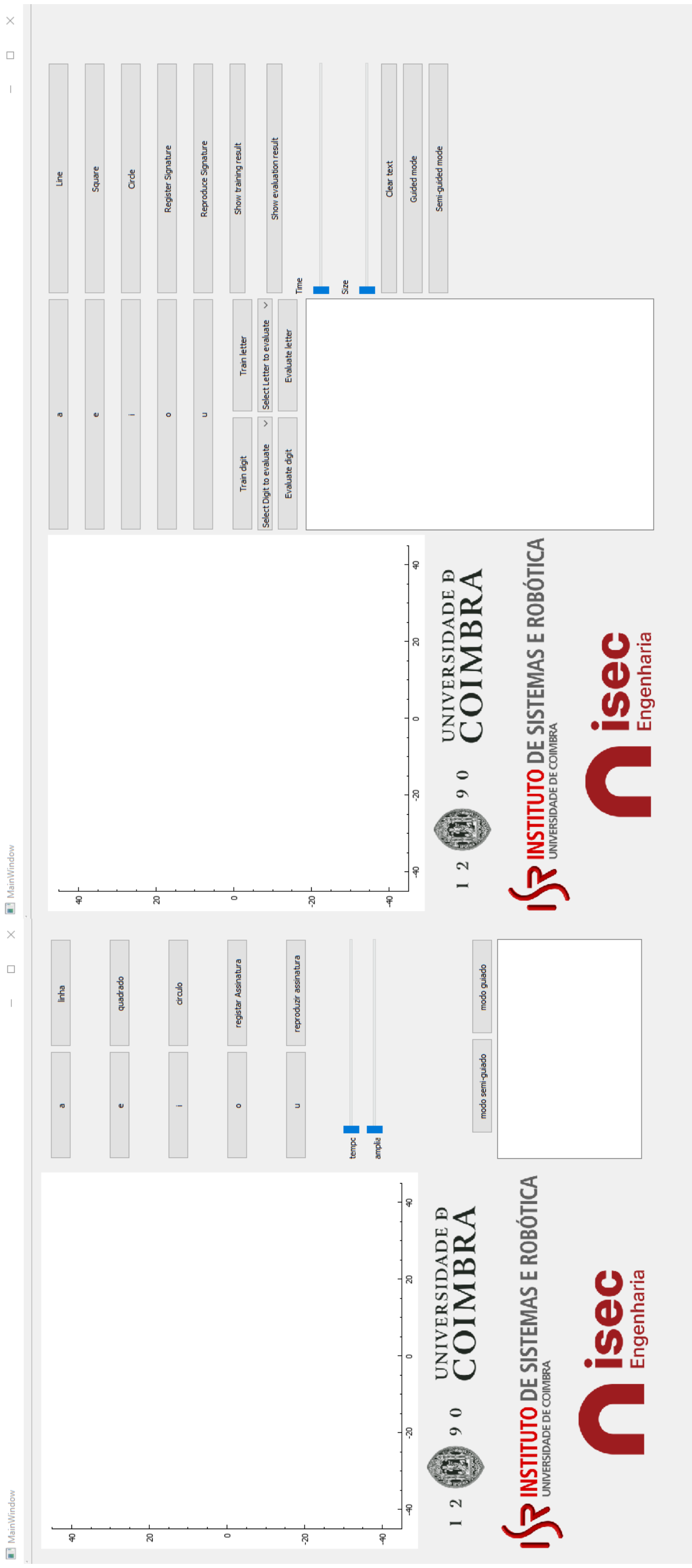
Letter	Mean	Standard deviation	Minimum
A	48.85	8.81	27.87
B	46.10	6.63	28.74
C	45.65	10.39	23.63
D	51.03	8.59	27.47
E	58.97	7.79	37.59
F	41.82	8.47	23.20
G	50.28	7.62	32.66
H	39.18	7.24	22.69
I	43.90	6.48	24.76
J	38.49	9.98	18.67
K	44.60	6.93	30.17
L	32.75	9.16	13.30
M	53.28	5.97	35.32
N	52.50	8.74	27.48
O	52.37	10.23	29.78
P	38.01	7.86	23.39
Q	52.94	6.96	39.32
R	55.43	5.62	40.21
S	44.98	9.38	24.86
T	47.88	12.01	15.56
U	45.51	11.75	21.00
V	39.77	8.55	21.54
W	46.70	6.55	30.46
X	41.92	8.77	24.19
Y	30.73	10.08	15.62
Z	48.87	8.88	24.41

ANNEX 4

Lowercase letters reference Dynamic Time Warping distances table built using MATLAB.

Letter	Mean	Standard Deviation	Minimum
a	52.84	10.01	31.79
b	39.54	11.13	18.50
c	46.77	8.87	29.17
d	41.52	11.18	22.96
e	53.73	8.09	33.15
f	36.95	8.00	22.46
g	48.10	10.88	28.43
h	33.82	10.63	17.39
i	22.60	11.05	8.12
j	34.72	12.92	10.83
k	41.13	7.83	20.35
l	23.85	14.03	4.74
m	45.41	7.73	26.02
n	51.50	7.61	34.97
o	50.61	10.05	28.08
p	32.86	7.91	19.14
q	46.56	10.76	23.81
r	43.17	10.69	22.43
s	47.54	8.32	26.92
t	41.05	8.87	21.30
u	48.22	7.42	28.04
v	38.98	8.87	23.91
w	47.09	6.46	32.48
x	43.21	8.47	27.06
y	32.04	8.73	15.88
z	50.44	7.04	32.85

ANNEX 5



User interface layout before (left) and after (right) addition of QPushButtons and QComboBoxes.

Bibliography

[1] Klimova, Blanka. (2012). The Importance of Writing. Paripex - Indian Journal of Research. 2. 9-11. 10.15373/22501991/JAN2013/4.

[2] Feder KP, Majnemer A. Handwriting development, competency, and intervention. Dev Med Child Neurol. 2007 Apr;49(4):312-7. doi: 10.1111/j.1469-8749.2007.00312.x. PMID: 17376144.

[3] Reynolds, C. (2007). Encyclopedia of special education: A reference for the education of children, adolescents, and adults with disabilities and other exceptional individuals (3rd ed.). New York: John Wiley & Sons.

[4] 1. Deuel RK. Developmental Dysgraphia and Motor Skills Disorders. Journal of Child Neurology. 1995;10(1_suppl):S6-S8. doi:10.1177/08830738950100S103

[5] Chung, P., & Patel, D. R. (2015). Dysgraphia. International Journal of Child and Adolescent Health, 8(1), 27.

[6] Rodrigues, J. C., da Fontoura, D. R., & de Salles, J. F. (2014). Acquired dysgraphia in adults following right or left-hemisphere stroke. Dementia & neuropsychologia, 8(3), 236–242. https://doi.org/10.1590/S1980-57642014DN83000007

[7] Allison, R., Shenton, L., Bamforth, K., Kilbride, C., & Richards, D. (2016). Incidence, Time Course and Predictors of Impairments Relating to Caring for the Profoundly Affected arm After Stroke: A Systematic Review. Physiotherapy research international: the journal for researchers and clinicians in physical therapy, 21(4), 210–227. https://doi.org/10.1002/pri.1634

[8] Lopes, J. F. P. (2020). Sistema Háptico de Treino da Escrita (Handwriting Training Haptic System). University of Coimbra, MSc Project report.

[9] NIST: https://www.nist.gov/srd/nist-special-database-19

[10] Antonio Di Carlo, Human and economic burden of stroke, Age and Ageing, Volume 38, Issue 1, January 2009, Pages 4–5, https://doi.org/10.1093/ageing/afn282

[11] Hable, Richard; Pergler, Elisabeth; and Ram, David, "Prototyping a Tablet Application for the Rehabilitation of Stroke Patients" (2013). 2013 International Conference on Mobile Business. 18. https://aisel.aisnet.org/icmb2013/18

[12] Jesús Blanquero, María-Dolores Cortés-Vega, Pablo Rodríguez-Sánchez-Laulhé, Berta-Pilar Corrales-Serra, Elena Gómez-Patricio, Noemi Díaz-Matas, Alejandro Suero-Pineda, Feedback-guided exercises performed on a tablet touchscreen improve return to work, function, strength and healthcare usage more than an exercise program prescribed on paper for people with wrist, hand or finger injuries: a randomised trial, Journal of Physiotherapy, Volume 66, Issue 4, 2020, Pages 236-242, ISSN 1836-9553, https://doi.org/10.1016/j.jphys.2020.09.012.

[13] (Jennifer Curtis, Loes Ruijs, Maartje de Vries, Robert Winters, and Jean-Bernard Martens. 2009. Rehabilitation of handwriting skills in stroke patients using interactive games: a pilot study. CHI '09 Extended Abstracts on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, 3931–3936. DOI:https://doi.org/10.1145/1520340.1520596.

[14] Kilbride C, Scott DJM, Butcher T, et alRehabilitation via HOMe Based gaming exercise for the Upper-limb post Stroke (RHOMBUS): protocol of an intervention feasibility trialBMJ Open 2018;8:e026620. doi: 10.1136/bmjopen-2018-026620

[15] E. J. Koeneman, R. S. Schultz, S. L. Wolf, D. E. Herring and J. B. Koeneman, "A pneumatic muscle hand therapy device," The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2004, pp. 2711-2713, doi: 10.1109/IEMBS.2004.1403777.

[16] P. Polygerinos et al., "Towards a soft pneumatic glove for hand rehabilitation," 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 1512-1517, doi: 10.1109/IROS.2013.6696549
[17] M. Mulas, M. Folgheraiter and G. Gini, "An EMG-controlled exoskeleton for hand rehabilitation," 9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005., 2005, pp. 371-374, doi: 10.1109/ICORR.2005.1501122.
[18] Behle, Carina R. (2020) Interventions for cognitive impairments and mental health by the means of virtual reality among stroke survivors.)
[19] Ben-Pazi, H., Ishihara, A., Kukke, S., & Sanger, T. D. (2010). Increasing viscosity and inertia using a robotically controlled pen improves handwriting in children. Journal of child neurology, 25(6), 674–680. https://doi.org/10.1177/0883073809342592
[20] Shire, K. A., Hill, L. J., Snapp-Childs, W., Bingham, G. P., Kountouriotis, G. K., Barber, S., & Mon-Williams, M. (2016). Robot Guided 'Pen Skill' Training in Children with Motor Difficulties. PloS one, 11(3), e0151354. https://doi.org/10.1371/journal.pone.0151354
[21] Gargot, T., Asselborn, T., Zammouri, I., Brunelle, J., Johal, W., Dillenbourg, P., Archambault, D., Chetouani, M., Cohen, D., & Anzalone, S. M. (2021). "It Is Not the Robot Who Learns, It Is Me." Treating Severe Dysgraphia Using Child-Robot Interaction. Frontiers in psychiatry, 12, 596055. https://doi.org/10.3389/fpsyt.2021.596055
[22] Feder, K.P. and Majnemer, A. (2007), Handwriting development, competency, and intervention. Developmental Medicine & Child Neurology, 49: 312-317. https://doi.org/10.1111/j.1469-8749.2007.00312.x
[23] James KH. The Importance of Handwriting Experience on the Development of the Literate Brain. Current Directions in Psychological Science. 2017;26(6):502-508. doi:10.1177/0963721417709821
[24] I. Shakra, M. Orozco, A. E. Saddik, S. Shirmohammadi and E. Lemaire, "Haptic Instrumentation for Physical Rehabilitation of Stroke Patients," IEEE International Workshop on Medical Measurement and Applications, 2006. MeMea 2006., 2006, pp. 98-102, doi: 10.1109/MEMEA.2006.1644470.
[25] Danna, J., & Velay, J. L. (2015). Basic and supplementary sensory feedback in handwriting. Frontiers in psychology, 6, 169. https://doi.org/10.3389/fpsyg.2015.00169
[26] J. Mullins, C. Mawson and S. Nahavandi, "Haptic handwriting aid for training and rehabilitation," 2005 IEEE International Conference on Systems, Man and Cybernetics, 2005, pp. 2690-2694 Vol. 3, doi: 10.1109/ICSMC.2005.1571556.
[27] Mansour, Mohammed & Eid, Mohamad & El Saddik, Abdulmotaleb. (2021). A Multimedia Handwriting Learning and Evaluation Tool.
[28] P. P. Shinde and S. Shah, "A Review of Machine Learning and Deep Learning Applications," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018, pp. 1-6, doi: 10.1109/ICCUBEA.2018.8697857.
[29] N. Prameela, P. Anjusha and R. Karthik, "Off-line Telugu handwritten characters recognition using optical character recognition," 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), 2017, pp. 223-226, doi: 10.1109/ICECA.2017.8212801.
[30] P. Dhande and R. Kharat, "Recognition of cursive English handwritten characters," 2017 International Conference on Trends in Electronics and Informatics (ICEI), 2017, pp. 199-203, doi: 10.1109/ICOEI.2017.8300915.
[31] N. Arica and F. T. Yarman-Vural, "Optical character recognition for cursive handwriting," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 6, pp. 801-813, June 2002, doi: 10.1109/TPAMI.2002.1008386.

[32] Yoshihiro Shima, Yumi Nakashima, and Michio Yasuda. 2018. Handwritten Digits Recognition by Using CNN Alex-Net Pre-trained for Large-scale Object Image Dataset. In <i>Proceedings of the 3rd International Conference on Multimedia Systems and Signal Processing</i> (<i>ICMSSP '18</i>). Association for Computing Machinery, New York, NY, USA, 36–40. DOI:https://doi.org/10.1145/3220162.3220163.
[33] Ebrahimzadeh, Reza, and Mahdi Jampour. "Efficient handwritten digit recognition based on histogram of oriented gradients and SVM." International Journal of Computer Applications 104.9 (2014).
[34] L. Bottou et al., "Comparison of classifier methods: a case study in handwritten digit recognition," Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3 - Conference C: Signal Processing (Cat. No.94CH3440-5), 1994, pp. 77-82 vol.2, doi: 10.1109/ICPR.1994.576879.
[35] Baldominos, A., Saez, Y., & Isasi, P. (2019). A Survey of Handwritten Character Recognition with MNIST and EMNIST. Applied Sciences, 9(15), 3169. doi:10.3390/app9153169
[36] Ziliotto, A., Cersosimo, M. G., & Micheli, F. E. (2015). Handwriting Rehabilitation in Parkinson Disease: A Pilot Study. Annals of rehabilitation medicine, 39(4), 586–591. https://doi.org/10.5535/arm.2015.39.4.586
[37] Carlo Di Brina, Ralph Niels, Anneloes Overvelde, Gabriel Levi, Wouter Hulstijn. Dynamic time warping: A new method in the study of poor handwriting, Human Movement Science, Volume 27, Issue 2, 2008, Pages 242-255, ISSN 0167-9457, https://doi.org/10.1016/j.humov.2008.02.012.
[38] MNIST: http://yann.lecun.com/exdb/mnist/
[39] EMNIST: https://www.nist.gov/itl/products-and-services/emnist-dataset
[40] Mendes, P. A. S., Ferreira, J. P., Coimbra, A. P., Crisóstomo, M. M., & Bouças, C. (2020). Hand Exercise Using a Haptic Device. Communications in Computer and Information Science, 1194 CCIS, 449–461. https://doi.org/10.1007/978-3-030-42520-3_36
[41] J. Schmidhuber, Deep learning in neural networks: An overview, 2015, Volume 61, Pages 85-117, ISSN 0893-6080, https://doi.org/10.1016/j.neunet.2014.09.003
[42] https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/)
[43] https://www.mathworks.com/help/signal/ref/dtw.html
[44] https://www.mathworks.com/matlabcentral/answers/447387-matlab-coder-how-do-i-build-the-intel-mkl-dnn-library-for-deep-learning-c-code-generation-and-dep
[45] https://www.mathworks.com/help/deeplearning/ug/create-simple-deep-learning-network-for-classification.html
[46] Hinz, Tobias & Barros, Pablo & Wermter, Stefan. (2016). The Effects of Regularization on Learning Facial Expressions with Convolutional Neural Networks. 80-87. 10.1007/978-3-319-44781-0_10. Image: https://www.researchgate.net/profile/Tobias-Hinz/publication/306081277/figure/fig1/AS:403580726071296@1473232551816/CNN-architecture-our-CNN-consists-of-three-convolutional-layers-with-10-15-and-20.png
[47] https://www.mathworks.com/help/vision/ug/digit-classification-using-hog-features.html
[48] Barbhuiya, A.A., Karsh, R.K. & Jain, R. CNN based feature extraction and classification for sign language. Multimed Tools Appl 80, 3051–3069 (2021). https://doi.org/10.1007/s11042-020-09829-y
[49] Chen, J. Deep Learning for Handwritten Digits Recognition using MATLAB Toolbox, 2018. https://dspace.library.uvic.ca/handle/1828/11353