Development of a mass drug administration monitoring system for a Humanitarian NGO

Rafael Ribeiro[1,2], Cláudio Teixeira[2], Pedro Vicente[2], and Paulo Baltarejo Sousa[1]

[1] School of Engineering/Polytechnic Institute of Porto, Porto, Portugal
{1190977,pbs}@isep.ipp.pt
[2] Mindera, Porto, Portugal
{rafael.ribeiro,claudio.teixeira,pedro.vicente}@mindera.com

Abstract. Mass drug administration is the process of distributing medication to people who are in areas with risk of neglected tropical diseases. The monitoring of this process is supported by the application of forms in the communities, which are then sent to the headquarters, where they are manually entered into governmental health information systems. This paper aims at finding a solution for automating the process, making it more efficient, reliable and scalable.
The proposed solution involves the usage of Optical Character Recognition (OCR) to extract the data, and is composed of a Backend, an Android Application and a Backoffice. It was evaluated using several metrics, achieving mean values of 0.8% Character Error Rate (CER), 2.5% Word Error Rate (WER), 3.7% Field Error Rate (FER) and 96.2% Precision, Recall and F1 score. Forms filled in English and forms filled using print handwriting style had significantly better results than forms filled in Portuguese and forms filled using cursive style, respectively. The application was used by two testers from the Nongovernmental Organization (NGO), resulting in positive feedback, with some improvements being suggested.

Keywords: Optical Character Recognition · Text extraction · Mass drug administration · Web Information System · Android development.

1 Introduction

The MENTOR Initiative is a Nongovernmental Organization (NGO) that works with the most vulnerable and hard to reach communities to reduce death and suffering from tropical diseases [20]. They cooperate closely with local communities, health workers, health authorities and other international organizations in insecure and high risk environments, such as conflict zones and other humanitarian emergencies [14].

Mass drug administration is one of the key interventions carried out by The MENTOR Initiative to prevent the risk of neglected tropical diseases, and its monitoring is essential for assessing the coverage and identify communities in need [11]. The monitoring process involves collecting data from the field using forms that contain information about the number of drugs distributed, side effects, inventory and supervision of the drug administration. The collected data is analysed and taken into account for decision-making, allowing the adoption of proper strategies and an effective resource allocation.

The monitoring process is based in the communication between the people appointed to distribute the medicine in the villages and the country headquarters. For each village, multiple paper forms with several pages are filled, which makes the data compilation difficult, inefficient and error-prone. The collected data is sent to the country headquarters, where a person is responsible for manually submitting the data into governmental health information systems. Besides the inefficiency, this process is both time-consuming and error-prone.

The main goal of the project is the development and evaluation of a tool that allows the digitization of mass drug administration forms, as well as the automation of data submission for different countries and NGOs.

This paper is structured as follows. Section 2 presents the state of the art about the text extraction technology and the available tools. Section 3 describes the analysis, design and implementation of the proposed solution. In Section 4, the developed solution is evaluated through experimentation. Finally, Section 5 presents the conclusions and future work about the project.

2 State of the Art

Optical Character Recognition (OCR) is a technology that extracts text from different types of documents, such as photos or scanned documents, so that it can then be indexed, stored or analysed. This technology serves as the foundational component of the proposed solution for the problem.

The concept of "reading documents by other than human means" appeared during World War II, motivated by the necessity of automating the transcription of documents [16]. During the early ages of OCR, it was also used to support visually impaired individuals [1], as well as reading musical notations and small words, even though it had a lot of limitations [2]. The commercially available OCR systems originated in early 1960s [1], and evolved into the latest generations, in which Machine Learning (ML) techniques, such as Artificial Neural Networks (ANN), were introduced, leading to a wider supportability and higher accuracy [1, 12].

Most of the modern OCR systems include multiple phases, such as preprocessing, segmentation, feature extraction, classification and post-processing [7, 19, 8, 13]. During the classification phase, several methods can be used, such as ML techniques, Template Matching and Structural Pattern Recognition [12].

The technology is widely used in multiple areas, such as document digitization, archiving, vehicle plate recognition, language translation and accessibility. However, there are some challenges to its implementation, like supporting multiple languages, recognizing handwritten text and dealing with low quality input images.

The literature was reviewed in order to understand the feasibility of using OCR for handwritten text extraction, answering the research question "Are there case studies about solutions that use OCR to extract handwritten text with accuracy?".

Several case studies were found using OCR for handwritten text recognition in real-world scenarios. One of them is [15], in which the technology is being used for the digitization of child protection cards. A custom solution was built using ML, achieving accuracies between 93% and 98% [15].

Both [21] and [9] explored the application of OCR for historical documents with crowdsourced post-correction. The first study focus on ancient manuscripts and reported difficulty in maintaining a stable community of contributors. The second study focus on historical Dutch documents, in which recognizing person names was a challenge.

In [18], a solution was developed for automating mark entry in educational institutions. It significantly accelerated data processing workflows by efficiently converting and processing the marks of students. The main challenge was detecting decimal numbers [18].

The case study [17] used an OCR cloud service to develop a social pension management system. The solution had a high accuracy, but struggled at differentiating uppercase and lowercase letters, as well as struck-out text [17].

In the case study [5], the recognition of handwritten code from Computer Science students was explored. When combining an OCR cloud service with post-correction techniques, it proved to be accurate enough to be used in real learning environments [5]. One of the main challenges of the solution is dealing with struck-out text [5].

The literature was also reviewed to find the OCR tool with higher accuracy for handwritten text, answering the research question "What studies are there about OCR tools with the best accuracy for handwritten characters?".

The OCR tools can be divided into three categories: proprietary software (e.g., ABBYY FineReader and Transym OCR), open source engines (e.g., Tesseract and Calamari) and online services (e.g., AWS Textract, Google Document AI and Azure Document Intelligence) [6].

The study [3] compares Tesseract, AWS Textract and Google Document AI, using a dataset that includes scanned documents in English and Arabic. The results suggest that the cloud services perform significantly better than Tesseract, with Google Document AI achieving a slightly higher accuracy than Textract.

In [4], all the types of OCR tool were analysed, by comparing Google Vision AI (cloud service), Tesseract (open source), ABBYY FineReader and Transym OCR (proprietary). The dataset included key-frames extracted from lecture videos with variations in the text orientation, noise and skewness. Google Vision AI was the tool that performed better, with an average accuracy of 96.7%.

In the comparative study [10], the tools Tesseract, AWS Textract and Google Vision AI were compared using a dataset that includes printed and handwritten medical records. Both cloud services performed significantly better than Tesseract, with AWS Textract achieving a slightly higher accuracy for handwritten text recognition, with 88.89% accuracy.

In general, the application of OCR for recognizing handwritten text has proven to be accurate in many case studies, either by using commercially available OCR tools or training custom ML models. The main challenges found were struck-out text and noisy images. Regarding the commercially available tools, the results suggest that the cloud services have better performance compared to the other types of tool, with the services from AWS (Textract) and Google (Vision AI and Document AI) achieving higher accuracy for handwritten text.

3 Development

This section presents an analysis of the problem and outlines the design for the proposed solution, with its implementation being described.

In order to understand the problem and specify the requirements, the domain was modelled into entities and relationships, as represented by the diagram in Fig. 1.

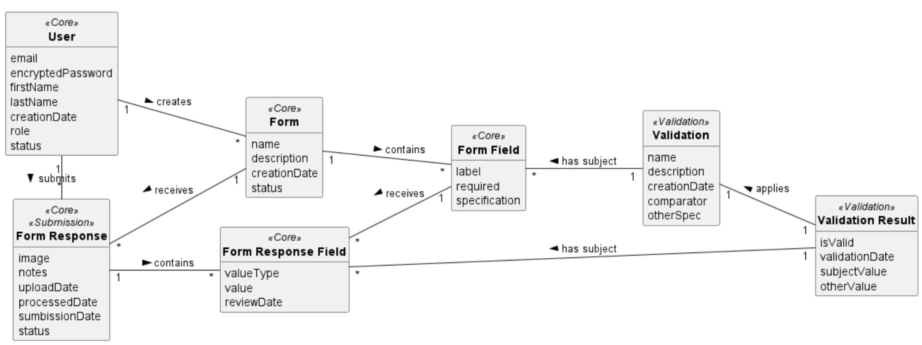


Fig. 1. Simplified domain model

The base concept of the domain is the Form entity, which represents the structure of a form that can be filled, being related to multiple Form Fields. The Form Response entity represents an uploaded filled response, being related to its respective Form Response Fields. As for the Validation entity, it specifies a comparison between Form Field values to validate their correctness. When it is applied to specific Form Response Fields, the Validation Result entity indicates whether it was successful or not. The User is a supporting entity that represents the user of the system.

Considering the domain model, the functional requirements were specified as use cases. A regular user can upload photos of filled forms, review the extracted data, apply validations and submit form responses. Besides that, an administrator can also register new users and define the structure of forms and validations.

The architecture of the system was designed as shown in Fig. 2.

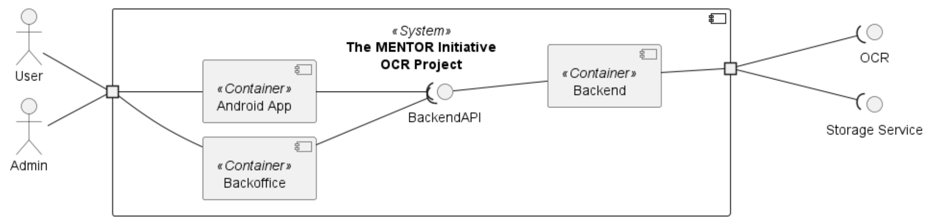


Fig. 2. Component diagram of the architecture of the system

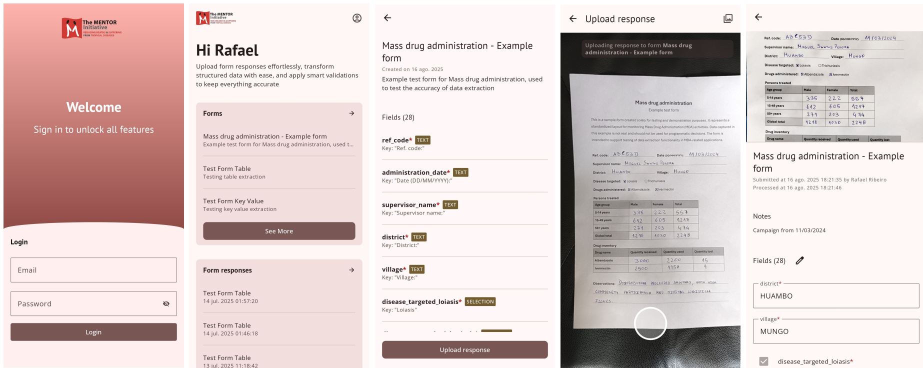
The system is composed of three components: Backend, Android App and Backoffice. The Backend is responsible for the business logic and data management, serving as a foundation for the frontend components by exposing a Representational State Transfer (REST) Application Programming Interface (API). It interacts with the external services, for data extraction and image storage. The Android App component allows the users to interact with the system and perform the main actions, such as uploading photos and applying validations. As for the Backoffice component, it allows the administrators to manage the system, such as creating new users and editing the structure of forms or validations.

The architecture of the Backend is based on a combination of Model-View-Controller (MVC) and layered architecture. It was built using NodeJS with the programming language TypeScript and the framework Express, which is responsible for exposing the REST API. PostgreSQL was used for the database, with the Object-Relational Mapping (ORM) framework Prisma being used to handle the database operations and migrations. The uploaded images were stored using AWS S3. For the data extraction, AWS Textract was used as the main OCR tool, considering the results of the literature review, with Google Gemini being used as a fallback tool to improve the extraction accuracy. The output from both tools is processed according to the form fields defined and their respective types.

A Model-View-ViewModel (MVVM) architecture was followed in the Android App, which was built natively using the programming language Kotlin. The framework Jetpack Compose was used to build the User Interface (UI) in a declarative style, with Jetpack Navigation being responsible for handling the navigation of the application. Retrofit was used for the network requests, while Jetpack DataStore was used for internal storage. The Fig. 3 includes some screenshots from the Android App.

After entering the credentials in the login screen, Fig. 3 (a), the user is redirected to the home screen, Fig. 3 (b). By accessing the details screen of a form, Fig. 3 (c), the user can upload a photo using the camera or gallery, Fig. 3 (d). When the upload is successful, the user is redirected to the details screen of the form response, Fig. 3 (e), with the extracted data.

The Backoffice follows a component-based architecture, being built using React with the programming language TypeScript, in a NodeJS environment.



(a) (b) (c) (d) (e)

Fig. 3. Android App screenshots of the main screens

This component was implemented in a partnership with Mindera School, which is a learning programme that introduces young people to the world of technology and programming.

4 Experimentation & Evaluation

Two experiments were conducted to evaluate the accuracy of the text extraction in the developed solution, as well as its usability and efficiency.

The first subsection will focus on measuring the accuracy of the text extraction, as well as testing the conditions that may affect it. The second subsection includes the analysis of the analytics data and the direct feedback from the testers.

4.1 Text extraction accuracy

In this context, the accuracy represents the quality and classification performance of the text extraction, which will be evaluated with a set of appropriate metrics. The accuracy of text extraction was tested using an example test form that was filled by multiple people, following a non-probability convenience sampling technique. Some forms were filled in English, while others were filled in Portuguese. Different styles of handwriting were also used, such as cursive or print handwriting. Each form response was captured in two different ways: by taking a picture with the camera inside the application or by scanning the form and picking the scanned image from the gallery.

Several metrics were collected at different depths. The Character Error Rate (CER), Word Error Rate (WER) and Field Error Rate (FER) were used to quantify the percentage of characters, words and fields that were incorrectly extracted, respectively. Besides that, the Precision was measured to determine the proportion of correctly extracted fields out of the fields that were extracted, while the Recall determines the proportion of correctly extracted fields out of the number of fields that should have been extracted. The F1 score represents the harmonic mean between the Precision and Recall.

The form was filled by 8 individuals, with 16 images being uploaded to the application. Table 1 summarizes the results that were achieved.

Table 1. Summary table for accuracy tests dataset

Metric	**Mean**	**Standard deviation**	**Minimum**	**Maximum**
CER	0.008	0.009	0.000	0.026
WER	0.025	0.026	0.000	0.063
FER	0.037	0.040	0.000	0.107
Precision	0.962	0.040	0.892	1.000
Recall	0.962	0.040	0.892	1.000
F1 score	0.962	0.040	0.892	1.000

The results indicate that the text extraction accuracy is generally high for the forms used, with low mean error rates (**0.8%** CER, **2.5%** WER and **3.7%** FER) and high performance metrics (**96.2%** Precision, Recall and F1 score). The higher error rates at word and field level can be explained by their lower granularity, which causes an incorrect character to have a higher impact on the value. The parity of the field-level metrics can be explained by the lack of fields in which the system could not extract a value.

To further analyse the results, the data was grouped by the method of capturing the form, language and handwriting style. Statistical tests were performed in order to compare the distributions for each group and determine if their differences are statistically significant. Given the small sample size and the non-normal distribution of the data, non-parametric tests were used.

In the performed tests, the null hypothesis states that there are no differences between the distributions. Considering the mean values for each group, the alternative hypothesis states that the error rates are lower and the field-level performance metrics are higher for scanned images, forms filled in English and forms filled with print handwriting style. Table 2 includes the results of the statistical tests.

Considering a significance level of 5%, there is not enough evidence to conclude that the scanned images allow for significantly better results than taking the picture with the camera inside the application. Conversely, there is enough evidence to conclude that the forms filled in English had significantly better results than the forms filled in Portuguese. For the handwriting style, it is also possible to conclude that the forms filled with print handwriting style had significantly better results than the forms filled with cursive handwriting style.

Table 2. Statistical tests for differences in grouped distributions

	CER	**WER**	**FER**	**Precision**	**Recall**	**F1 score**
Scanned vs Camera (upload type)						
Skewness	-1.132	-1.627	-1.564	1.564	1.564	1.564
p-value	0.0625*	0.25*	0.25*	0.25*	0.25*	0.25*
English vs Portuguese (language)						
p-value	0.0018**	0.0017**	0.0023**	0.0023**	0.0023**	0.0023**
Print vs Cursive (handwriting style)						
p-value	0.0307**	0.0178**	0.0195**	0.0195**	0.0195**	0.0195**

*Sign test; **Mann-Whitney U test

4.2 User testing

The user testing session was conducted in a partnership with two persons from the NGO, with the objective of gathering feedback from the target users and evaluate the extraction of fields. Before starting the testing, a demonstration meeting was held to showcase the application and provide instructions. The tests were performed during the first week of August 2025, with official forms being used, filled with fake data due to confidentiality reasons.

After performing the tests on the application, a meeting was held with the testers. The feedback gathered from them during the meeting was generally positive, with some annotations being shared about their experience using the application. Even though the solution is missing the capability to submit the form response directly to the health information system of the partner, the testers saw the potential of the application to improve their workflow.

Some points for improvement were mentioned, such as the ordering of the fields in the form response details screen, the capability to delete an uploaded form response and the size of the numeric fields in the form response details screen. Besides that, the testers mentioned that validations could be more intuitive and be applied automatically after uploading an image.

To monitor the usage of the application during the tests, Firebase Analytics was integrated with custom events, tracking the extraction rate, which indicates the proportion of fields that the system managed to extract a value, the review rate, indicating the proportion of fields that were reviewed, and the result of the validations. The extraction and review rates do not represent real accuracy metrics, as the fields can be extracted incorrectly or be left empty.

A total of 10 form responses were uploaded and had their fields extracted during the testing session. The mean extraction rate was **82.8%**, which means that, on average, 17.2% of the fields could not be extracted from the uploaded forms. Through observation, it was possible to notice that the forms were partially filled, leading to the system not being able to extract a value from fields that were empty. The mean review rate was **25%**, which could have been caused by the empty fields. During the tests, 18 validations were applied, with **10 (56%)** of them returning a valid result.

5 Conclusions

This paper aimed at finding a solution to improve the efficiency of the paper-based mass drug administration process. A solution with a Backend, an Android App and a Backoffice was designed and implemented, using AWS Textract and Google Gemini for the text extraction. It achieved a mean value of 0.8% CER, with 96.2% Precision at field level, which is similar, or even better in some cases, compared to the results found in the literature. It was possible to conclude that filling forms in English and with print handwriting style resulted in significantly better results than filling in Portuguese and using cursive style, respectively. On the other hand, there were no significant differences for the upload type. The proposed solution showed potential to improve the efficiency of the process, which can lead to a faster decision-making and, consequently, a more timely and accurate response to neglected tropical diseases.

The main goal was mostly completed, except for the automation of data submission. The lack of time and the complexity of the project were the biggest challenges during the project. Another limitation was the small sample size in the experimentation. The main piece of future work is the integration with the external health information systems to automate the submission of form responses.

References

1. Berchmans, D., Kumar, S.S.: Optical character recognition: An overview and an insight. 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies, ICCICCT 2014 pp. 1361–1365 (12 2014). https://doi.org/10.1109/ICCICCT.2014.6993174
2. Dubey, M.R.: Machine learning in the field of optical character recognition (ocr). International Journal of Trend in Scientific Research and Development (IJTSRD) **4** (2020)
3. Hegghammer, T.: Ocr with tesseract, amazon textract, and google document ai: a benchmarking experiment. Journal of Computational Social Science **5**, 861–882 (5 2022). https://doi.org/10.1007/S42001-021-00149-1/FIGURES/10
4. Hukkeri, G.S., Goudar, R.H., Janagond, P., Patil, P.S.: Machine learning in ocr technology: Performance analysis of different ocr methods for slide-to-text conversion in lecture videos. International Journal of Advanced Computer Science and Applications **13** (2022). https://doi.org/10.14569/IJACSA.2022.0130839
5. Islam, M.S., Doumbouya, M.K.B., Manning, C.D., Piech, C.: Handwritten code recognition for pen-and-paper cs education. L@S 2024 - Proceedings of the 11th ACM Conference on Learning @ Scale pp. 200–210 (7 2024). https://doi.org/10.1145/3657604.3662027
6. Jain, P., Taneja, K., Taneja, H.: Which ocr toolset is good and why : A comparative study. Kuwait Journal of Science **48**, 1–12 (4 2021). https://doi.org/10.48129/KJS.V48I2.9589
7. Karthick, K., Ravindrakumar, K.B., Francis, R., Ilankannan, S.: Steps involved in text recognition and recent research in ocr; a study. International Journal of Recent Technology and Engineering (IJRTE) **8**, 2277–3878 (2019)

8. Kaur, A., Baghla, S., Kumar, S.: Study of various character segmentation techniques for handwritten off-line cursive words: A review. International Journal of Advances in Science Engineering and Technology **3**, 2321–9009 (2015)
9. Koolen, M., Hoekstra, R., Oddens, J., Sluijter, R., Koert, R.V., Brouwer, G., Brugman, H.: The value of preexisting structures for digital access: Modelling the resolutions of the dutch states general. Journal on Computing and Cultural Heritage **16**, 2019–2024 (6 2023). https://doi.org/10.1145/3575864/ASSET/D262D520-E752-4D06-8012-329CCFB74EEF/ASSETS/GRAPHIC/JOCCH-21-0143-F08.JPG
10. Kumar, A., Singh, P., Lata, K.: Comparative study of different optical character recognition models on handwritten and printed medical reports. International Conference on Innovative Data Communication Technologies and Application, ICIDCA 2023 - Proceedings pp. 581–586 (2023). https://doi.org/10.1109/ICIDCA56705.2023.10100213
11. Mass drug administration - the mentor initiative, https://mentor-initiative.org/activity/neglected-tropical-diseases/mass-drug-administration/
12. Memon, J., Sami, M., Khan, R.A.: Handwritten optical character recognition (ocr): A comprehensive systematic literature review (slr). IEEE Access **8**, 142642–142668 (2020). https://doi.org/10.1109/ACCESS.2020.3012542
13. Nguyen, T.T.H., Jatowt, A., Coustaty, M., Doucet, A.: Survey of post-ocr processing approaches. ACM Computing Surveys (CSUR) **54**, 1 – 37 (2021). https://doi.org/10.1145/3453476
14. Our vision - the mentor initiative, https://mentor-initiative.org/our-vision/
15. PantDevesh, TalukderDibyendu, SethAaditeshwar, PantDinesh, SinghRohit, DuaBrejesh, PandeyRachit, MaruthiSrirama, JohriMira, AroraChetan: Robust ocr pipeline for automated digitization of mother and child protection cards in india. ACM Journal on Computing and Sustainable Societies **1**, 1–24 (9 2023). https://doi.org/10.1145/3608114
16. Robertson, A.: Optical character recognition: A study of success and failure in innovation. Management Decision **9**, 213–223 (1971). https://doi.org/10.1108/EB000971
17. Samonte, M.J.C., Bejar, A.M.L., Bien, H.C.L., Cruz, A.M.D.: Senior citizen social pension management system using optical character recognition. ICTC 2019 - 10th International Conference on ICT Convergence: ICT Convergence Leading the Autonomous Future pp. 456–460 (10 2019). https://doi.org/10.1109/ICTC46691.2019.8940013
18. Shaju, A.T., Jo, J.T., Abraham, E.S., Vishnuprasad, K.G., Deepa, V., Mathew, J.: Document digitization using optical character recognition - a case study of mark entry automation in educational institutions. 2024 1st International Conference on Trends in Engineering Systems and Technologies, ICTEST 2024 (2024). https://doi.org/10.1109/ICTEST60614.2024.10576116
19. Thanki, J.D., Davda, D., Swaminarayan, P.: A review on ocr technology **8** (2021)
20. The mentor initiative, https://mentor-initiative.org/
21. Wecker, A.J., Raziel-Kretzmer, V., Kiessling, B., Ezra, D.S.B., Lavee, M., Kuflik, T., Elovits, D., Schorr, M., Schor, U., Jablonski, P.: Tikkoun sofrim: Making ancient manuscripts digitally accessible: The case of midrash tanhuma. ACM Journal on Computing and Cultural Heritage (JOCCH) **15** (4 2022). https://doi.org/10.1145/3476776