

Final Report

Country-independent MRTD layout extraction and its applications

Eric Santiago Garcia

s2818337 - e.santiagogarcia@student.utwente.nl

September, 2022

Contents

1	Introduction	5
1.1	Problem Statement	6
1.2	Internal work at InnoValor	9
1.2.1	Information extraction	9
1.2.2	Document synthesis	10
1.3	Research Questions	10
1.4	Overview	11
2	Related work	12
2.1	Types of Information Extraction	12
2.2	Visually Rich Documents	14
2.2.1	Visually Rich Documents: Multi-step approaches	14
2.3	Summary	17
3	Design of MRTD Layout Finder	19
4	Method and Results	21
4.1	Datasets	21
4.2	Module 1: Layout extraction (RQ1)	22
4.2.1	Text Recognition and Preprocessing	24
4.2.2	Document cropping	28
4.2.3	Document representation	32

4.2.4	Template creation	40
4.3	Module 2: Privacy protection (RQ2.1)	41
4.4	Module 3: Fake image generation (RQ2.2)	45
4.4.1	Methods for Fake image generation	45
4.4.2	Results of automated evaluation of protected and synthesized documents (RQ3.1)	48
4.5	Human evaluation	53
4.5.1	Methods for Human evaluation	53
4.5.2	Results of Human evaluation (RQ3.2)	54
4.6	Correlation between metrics (RQ3.3)	59
4.7	Runtime of the tool	61
5	Conclusion	62
6	Future Work	64
6.1	Data preprocessing	64
6.2	Document representation	64
6.3	Applications of extracted templates	65
A	Appendix	71
A.1	Template example	71
A.2	Human Evaluation Form	73

Glossary

BSN Burgerservicenummer (Citizen Service Number).

DNN Deep Neural Network.

EU European Union.

GDPR General Data Protection Regulation.

ICAO International Civil Aviation Organization.

IE Information Extraction.

KYC Know Your Customer.

MRTD Machine Readable Travel Documents.

MRZ Machine Readable Zone.

NFC Near-Field Communication.

NLP Natural Language Processing.

NSS Natural Scene Statistics.

OCR Optical Character Recognition.

SVM Support Vector Machine.

VIZ Visual Inspection Zone.

VRD Visually Rich Documents.

1 Introduction

Passports and other travel documents are internationally accepted for identity verification. They can be used in different scenarios, for example in border control, where either an agent or a machine verifies that the travel document a person is carrying belongs to them.

Each country issues its own travel documents for its citizens in a specific format of their choice. Since 1980, many countries started issuing what are known as Machine Readable Travel Documents (MRTD), that contain information encoded in the Optical Character Recognition format. The International Civil Aviation Organization (ICAO) manages the standardisation process of these documents and periodically provides guidelines. The most recent ones are from 2016 and they are published in Doc 9303 [1]. In this document, the format of MRTDs is standardized in terms of size, ratio and mandatory data fields that must appear in a document. However, each country has still some freedom when designing their MRTD and different documents might vary in terms of color, layout, text format, or the exact data fields that are available, apart from the mandatory ones [2].

ICAO's standards made sure that identity documents could be read by machines. In order to facilitate this task, documents can incorporate an NFC (Near-Field Communication) chip that contains, as specified by the same standard, the most important information and can be read wireless with a simple antenna that can be included, for example, in smartphones. At the same time, it also allows visual inspection by having the information in two different zones: a Visual Inspection Zone (VIZ), where all data (including optional) is found, and is meant for humans to understand it; and a Machine Readable Zone (MRZ), that contains only essential information, and is meant to be read by machines. This can be seen in Figure 1.



Figure 1: Dutch passport, with VIZ and MRZ annotations. Source: Adapted from PRADO [3].

The adaptation of these documents to machines, together with current advances in AI, makes it possible to apply machine learning-based systems to automate tasks regarding the use of MRTDs that were previously performed by humans. For example, banks can do remote customer on-boarding by performing the identity-proofing online, extracting the information from the user’s identity document automatically from the VIZ and MRZ.

1.1 Problem Statement

As it was commented before, the applications for automated use of MRTD are numerous. However, designing these applications can be a very complex task, firstly because of the little data that is publicly available. Additionally, these tasks tend to be performed by mobile devices under uncontrolled conditions, providing bad-quality images [4]. Finally, because ID documents have a large variety in their layouts.

Any automating application needs examples during the design process, and this is especially true for machine learning-based approaches, that need an even larger quantity of data to be properly trained. Unfortunately, given the confidentiality of the personal data that is contained in identity documents, it seems not to be possible to gather a large dataset of real identity documents and make it available for research. As far as our understanding goes, there is only one publicly available set of identity documents created by the Council of the European Union, the PRADO register [3]. This register has different identity and travel documents from European countries, as well as some other countries. It contains different

versions of documents and offers a description of them, detailing some specifications like the fields available and security features. However, for each document, there is only one sample, a specimen document. Some research projects use the MIDV dataset [5], which will be commented on in Section 4. This dataset contains synthesized documents, not real ones. For many applications, there is the need to store ID documents. For example, financial institutions need to follow some Know Your Customer (KYC) guidelines. These detail how to verify a customer's identity, part of anti-money laundering policies. Thus, these institutions might have to store the data, including actual copies of the identity documents, needed for the business-customer relation or as a proof that this identity-proofing has been performed. In order to restrict how a company acquires and manages its customers' data, there exist some regulations like the General Data Protection Regulation (GDPR) from the EU, that give details on the storage of personal data. Under GDPR, the institutions are only allowed to store the essential data for their application but no other. For example, the Dutch social security number (BSN) shouldn't be stored by private companies. It is, however, part of the VIZ and MRZ. Some processing of raw images of real documents is needed to preserve the confidentiality and follow these restrictions. One possibility could be blurring or erasing the fields that are considered sensible for that application. Nevertheless, this task can be time-consuming if done by hand and no automated way is known for the full variety of documents.

As previously mentioned, the second aspect that complicates the design of applications for MRTDs is the capture conditions of the input data. Most systems use an image, often captured with a mobile device, under random conditions. These uncontrolled conditions mean that the scene's geometry, lighting conditions, or perspective vary for each image, they are unknown. This, together with the nature of mobile cameras, results in a, typically, low-quality image that might also include some blurs due to motion or other unexpected defects.

Thirdly, the format and variability of travel documents also play an important role. Since each country issues its own documents, in a particular fashion of its choice, layouts are non-uniform. For instance, each country might choose to include a different set of information fields and locate them in arbitrary positions, different for each country. A clear example can be found in Figure 2. In this picture of Swedish and Spanish passports, the displayed fields vary per country. For example, Sweden includes the field height, whereas Spain doesn't. However, even the common ones appear in different positions of the documents. Moreover, this variety in field location can also be seen in different versions of the same document and

country, see documents in the same row, e.g. the expiry date changes its location completely between the two Spanish passports.

Additionally, travel documents include some forgery protection: complex backgrounds with watermarks or holograms, special fonts or embossed printed elements; making each document style unique, and different from the others. These security features are essential for MRTDs, as identity-proofing must be safe and should not let any forged document through. In order to check the security features, highly trained humans are needed, no fully automated way is known, given the complexity of the features.

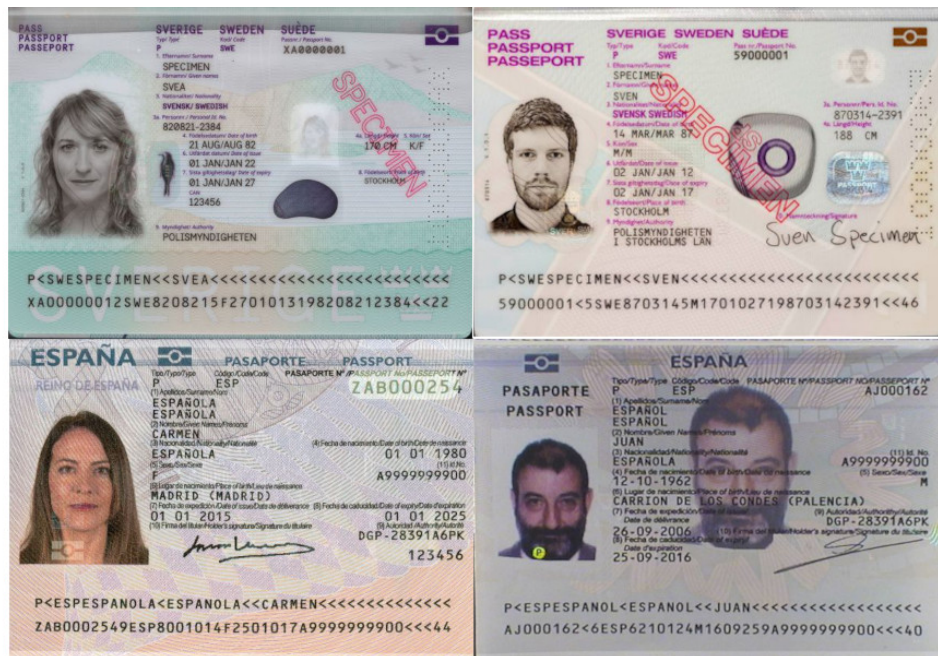


Figure 2: Swedish and Spanish passports, first and second rows, respectively, showing field variety. Source: Adapted from PRADO [3].

Hence, no matter the chosen approach for the system design, there are always some obstacles. If a system is rule-based, it needs prior knowledge about the documents it will process and it will also need constant adaptation to new templates or versions. These systems have poor generalization capacity and need constant updates for new documents. Alternatively, we might want to use a system that could hold more generalization capacity, using an algorithm that learns to extract the templates by itself. Unfortunately, it would need a large amount of data, not publicly available, and, for now, there are no known systems that are able to generalize for tasks involving MRTD.

These three characteristics are the reason why it is difficult to design systems that are able to generalize the information extraction task from identity documents, one of the main goals of this project. Bad capture conditions don't allow easy extraction of information that has big variability and is distorted by security features. Moreover, it's difficult to gather a large enough amount of data so all these phenomena can be overcome. Unlike some other information extraction tasks, as will be mentioned in the next section, MRTD documents miss a structure that helps the understanding of the different parts of the document, like the one given by grammar or a fixed tabular structure.

As previously mentioned, the information extraction is not the only task that can't be automated yet. There are two other related tasks with the same problem. These are the protection of sensitive data fields and generation of fake documents, which could benefit from an automated layout extraction from identity documents.

1.2 Internal work at InnoValor

1.2.1 Information extraction

This thesis was carried out at InnoValor, a company that, among other things, focuses on identity documents. In order to get a better understanding of why a solution to this problem is needed and being aware of the current situation, here we review their work. They developed the product ReadID that by means of reading the MRZ and NFC chip performs information extraction and verification of identity documents like passports, identity cards, or driving licenses. It is focused on the use of the NFC chip, because of the robustness that cryptography offers against forgery as well as the perfect accuracy of the extracted information. Thus, it only extracts the few fields that are available there, skipping some other fields that are available in the Visual Inspection Zone.

Moreover, for some use cases, real images of documents are required, rather than a cryptographically signed extract, and it is necessary to blur privacy-sensitive fields in these images. Innovalor had already created a tool for this task, but it requires manually-crafted templates for each kind of document, a time-consuming process that could be done faster if the template was automatically extracted.

1.2.2 Document synthesis

Another small project from Innovalor is SynthID, which had the goal of creating fake identity documents that could be used for testing machine learning modules with big amounts of data, avoiding the confidentiality constraint. No other work on document synthesis that specifies implementation details has been found in the company or the literature, so this is our only reference for fake document synthesis. This tool is also country and document specific and is based on human-crafted rules and templates, so it needs a prior study of the specific kind of document that needs to be generated. The first step is manually removing text fields from background images using photo editing software. Then, given the known coordinates of each data field, new information is inserted, using a set of predefined names and generating dates and pictures.

1.3 Research Questions

Having reviewed the main problems that come with the applications that manage identity documents or try their automated extraction, we can see how, despite similar work in similar fields, it is still a challenge how to automatically extract, in a generalized way, their information, especially from the Visual Inspection Zone. Thus, in order to find solutions to the general layout extraction problem from MRTDs, that would also allow an easy protection of sensitive data and fake document synthesis, we propose the next questions:

1. How can we detect the layout of a MRTD document so we can automatically extract the set of mandatory data fields, that appear in both VIZ and MRZ, in a country-independent fashion, without needing to adapt our tool to new versions of documents or new countries?
 - 1.1. How can the extracted layout of an MRTD be evaluated?
 - 1.2. How well does the layout extraction perform?
2. Can these templates be applied by a company like InnoValor in relevant circumstances?
 - 2.1. How can we use the templates to automatically censor sensitive data?

- 2.2. How can we use the templates to automatically clean text from the image and reinsert new data?
- 3. How can we evaluate the methods used for RQ2.1 and RQ2.2?
 - 3.1. How well the censor methods perform according to automated measures?
 - 3.2. How well the censor methods perform according to human juries?
 - 3.2. What is the correlation between automated and human measure?

1.4 Overview

Given this introduction to the project we will discuss related work on extracting information MRTD documents in Chapter 2. This knowledge allowed the design and implementation of our solution, detailed in a high-level in Chapter 3. Afterwards, in Chapter 4 we detail the dataset used and we describe each of the three modules that form our projects in a different section, aimed at a different research question. For each module, we describe the methods used, the different parts that compose it and the evaluation that we used. Each of these parts ends with a results and discussion section. Finally, we have the conclusion in Chapter 5 that synthesizes all important results and insights from the project and it is continued by a future work discussion in Chapter 6, detailing work directions that could be taken in the future.

2 Related work

Information extraction (IE) is the task of extracting structured information from unstructured documents [6]. As previously mentioned, most efforts have been aimed at other fields like extracting key information from newspapers and webpages or invoices: straightforward tasks thanks to the grammar of the text and the layout that HTML or tabular structures offer, respectively. Thus, in this section we will review some work that has been carried out in this field. First, with a more general point of view, reviewing the two kinds of approaches for the task of IE and classifying the task into 4 different types, the last one including MRTDs. This order corresponds, approximately, to the chronological order of their creation.

As presented in [6], there are two approaches for extracting patterns:

1. **Rule-based systems.** Systems able to write rules that extract the information for a particular task, given a small set of examples. Major drawbacks are small scalability and generalization to other tasks (or documents), they need constant adaptation to be able to process new kinds of documents.
2. **Machine learning models.** Extraction rules are automatically generated from training examples, using Machine Learning algorithms. The main drawback is the large amount of data that is needed to train the models. As mentioned, the nature of these documents makes this challenging. Moreover, despite further research on recent work, no model has been found to efficiently generalize for a broad number of documents. Thus, our goal is finding a Machine Learning model that is able to perform our extraction task.

2.1 Types of Information Extraction

Three types of extractions are presented in [6] and we will update them by adding a fourth one, more related to the identity document task.

1. **IE from Free Text.** Firstly, there is IE from free text, for those documents with plain texts, with grammar. The extraction rules are based on syntactic and semantic constraints, for example, AutoSlog [7] For example, we could try extracting writers

from Wikipedia articles. Given some text, we could build an extraction that looks for a triggering word (like *written*) and extracts the *agent compliment* of the passive sentence (what would be the subject performing the action in the active sentence), the word that follows *by*. A complete example would be looking for the writer of Don Quixote. Given "Don Quixote is a Spanish epic novel written by Miguel de Cervantes", our extraction pattern would be triggered because of the word *written* and would extract the agent compliment that is the word after the preposition *by*. Thus, we would extract *Miguel de Cervantes* as an author.

2. **IE from Online Documents.** The second kind is IE from online documents, which became a very popular topic thanks to the expansion of the Web. In this case, delimiters are added to the previous syntactic and semantic constraints. These delimiters bound the text to be extracted, in a fashion similar to regular expressions. One of the most famous examples is WHISK [8]. An example could be the extraction of a rent price given an announcement.

Flat to rent in Amsterdam

Number of rooms 4

Big garden

Price 1250€

We need an extraction rule that could look like: **Price (<Nmb>)€**. This patterns ignores everything until it finds the string Price, and stores a number until the euro symbol is found.

3. **Wrapper Induction Systems.** The third one is Wrapper Induction Systems that were born as an attempt to improve the generalization capacity of the previous systems. To do so, the HTML structure is used instead of linguistic constraints. HTML tags are used to build delimiter-based rules, as presented in WIEN [9]. This third approach has been improved in the future, as presented by Gogar et al. [10]. This more modern approach makes use of Deep Learning, in order to extract information from unseen websites. In this case, a Convolutional Neural Network learns this wrapper which has generalization power, thanks to the use of Machine Learning.
4. **IE from VRD.** Since the previous techniques are based on grammar or the specific structure of web pages (given by HTML), they are not effective enough when adapted

to other kinds of documents of another nature, like invoices or identity documents. Hence, we define a fourth task of IE from Visually Rich Documents (VRD). VRD are defined as those documents like invoices, receipts, declaration forms, or identity documents; that have a 2D structure that contains some information, allowing the proper understanding of the document and interpretation of the data that is contained. Although some classic techniques perform this task as a simple NLP task, extracting entities from plain text [11], their performance at classifying words for the information extraction from VRD is worse than that obtained with simple documents like books or articles. These techniques ignore the visual part of the document, essential by nature, so new techniques that also include Computer Vision are being developed, to exploit visual features of the documents. This, as it will be reviewed in the upcoming chapter, establishes the trend of using Machine Learning-based methods over rule-based ones for IE.

2.2 Visually Rich Documents

When it comes to IE models for VRD, we can see two different trends: end-to-end systems and multi-step approaches.

2.2.1 Visually Rich Documents: Multi-step approaches

Multi-step approaches split the task into three steps: text recognition, document representation and entity extraction. The main focus is on document representation, making sure both textual and visual information are used, so they can later perform the entity extraction task with the best possible input. One example can be found in [12], where the idea of keeping the semantic structure, determined by visual features like layout and font size, motivated the representation of each document as a graph containing both textual and visual information.

Text, firstly obtained with Optical Character Recognition (OCR) methods like Tesseract [13], is contained in nodes and the edges contain spatial information between the edges. Afterward, the entities are extracted using a very common Machine Learning model: BiLSTM-CRF [14]. Others have worked following this same idea, like [15], that represents a document using a graph, including a graph learning module, so the information contained in the graph is not just text and spatial coordinates but some more complex features that the network

learns by itself, yielding a richer semantic representation.

There is a second trend of document representation, that was started with Chargrid [16] where, in order to keep the 2D layout of documents, documents are encoded as a grid of characters. For each pixel, the corresponding character is encoded as a scalar or one-hot vector. Afterward, a convolutional encoder-decoder is used to perform a segmentation task at a pixel level and a further instance segmentation task that groups pixels and creates bounding boxes for each group of characters that belong to the same class. In the same paper, the use of a word-level grid is suggested, instead of the char-level one. Other works have been created, trying to expand this idea of char/word grid, like BERTgrid [17]. It improves upon the idea of Chargrid by representing a document as a grid of words, encoded with contextualized word piece embedding vectors, using a BERT model. The most recent work is VisualWordGrid [18], which using also a word-level grid, encodes also information about the visual layout by encoding the RGB channels together with the word embedding.

2.2.1.1 VRD: End-to-end approaches The second approach for IE from VRD are end-to-end models. A single model that performs different tasks in just one step, for example joining the text recognition from an image and the later entity extraction. An example is VIES [19], that splits the process of IE into three different independent steps: text detection/recognition, document representation and information extraction (or entity extraction), ignoring the correlation between them. In [19] they also state that if the correlation was used, it could be very beneficial for the optimization, by collecting both visual and semantic representations for the entity extraction task. As an example of making use of this correlation, VIES, a neural network model, is presented. It has a shared backbone for feature extraction, followed by three separated branches: two parallel text detection and recognition branches that are finally connected to the information extraction branch that performs the information extraction with the standard BiLSTM-CRF model. Another widely known end-to-end model is EATEN [20], standing for Entity-aware Attention Text Extraction Network. With a similar idea of a CNN-based backbone for high-level feature extractors, a sequence of decoders follow. Each decoder is dedicated to one (or a few) entities extraction and, despite being independent, they can have available semantic information from all the previous ones. The authors of EATEN created their own dataset, to extract information from train tickets, passports, and business cards. For passports, they obtained a 90.8% mean entity accuracy, with a time cost of 242ms. The passports part of their dataset is only formed by the same

version of the Chinese passport, with different identities, not showing any generalization capacity.

2.2.1.2 VRD: Focus on MRTD

Finally, we will focus on work related to MRTDs. In most of the works we found, only the information from the MRZ is extracted, because of its fixed format. If not, if they extracted information from the VIZ, they used template matching methods. In template matching, we look for the region contained in a sample image that is the most similar to our template image. Given the known coordinates of our regions of interest in the template, we can find these same regions in the sample image, once matched, and be able to extract the information inside those [21]. Thus, identification of the kind of document and prior knowledge of it is needed.

Next, we will introduce specific examples of work with MRTDS. In [22] the MRZs of 50 different passports are extracted and, afterward, cross-checked with the information obtained from the upper personal information area, to check fraud information. They have independent steps, like detecting the area of the passport, dividing the image into regions, and extracting MRZ and the personal information area. All of them are based on traditional methods that don't use machine learning techniques, like binarization, morphological filters, and histogram projection. Using template matching, they reach almost perfect extraction results, 99.8% for what they define as personal information extraction, in 2.3s per page.

Hartl et al. [23] created a tool for real-time detection and recognition of MRZ using mobile devices. Nowadays, as we can see from InnoValor's work or read in [24], the extraction of MRZ is performed with mobile phones, resulting in images of bad quality that suffer from perspective distortion, given the angle used for capture, making the recognition task more difficult. The system in [23] tries to cope with all these obstacles and automatically detects the document, fixes elastic distortions due to the capture angle, and extracts the MRZ by template matching. Instead of single-shot images, a sequence of them is used, giving better results. Despite reaching an MRZ-detection accuracy of 88.18% using multiple frames, only 63.40% reading accuracy is obtained, with a runtime of around 50ms for mobile devices.

Fang et al. [25] still use a classical approach for Chinese ID card identification from quality images, it focuses on the pre-processing part and highlights the use of Hough Transform to detect the area of an ID card in an image. Hough Transforms simplifies the process of de-

tecting lines (or other arbitrary shapes) in an image, by mapping the image into the Hough transform space where lines are easier to detect [26]. By template matching and an SVM model for complex fields like name and address, an accuracy of 90% for English characters is obtained. Given the improvement of using video sequences as input when a mobile device is used, [27] designed a system for Russian documents. It uses the Fast Hough Transform (a fast algorithm implementation of Hough Transform) [28] for detection of the document and morphological analysis to split the documents into the different fields, given a known template. However, it also introduces the use of Deep Learning techniques to detect and segment the characters inside the fields. Finally, the main innovation is a post-OCR post-processing that integrates results from different video frames. It obtains a 97,6% mean accuracy for Russian passports.

Tavakolian et al. [2] present the most used techniques for real-time information retrieval from ID cards, focusing on the preprocessing part for document detection and preparing the image for the best text recognition. Different methods are presented, but, in general, those based on Deep Learning techniques obtain the best results. Lastly, [29] present an end-to-end model for MRZ extraction. Given the fact that OCR usually fails to extract characters from digital images of passports, it is suggested that the text recognition task in identity documents might be more similar to text detection in real scenarios. Thus, they built their end-to-end model based on convolutional neural networks. Given an image, it extracts the MRZ bounding box and the text inside of it. A 100% MRZ detection rate is achieved and 99.25% character recognition macro-averaged f1 score. This system can process different image sizes and doesn't need prior knowledge about the document template. However, the runtime is around 15s, which can be too long for real-time applications.

2.3 Summary

In a nutshell, we saw how the task of information extraction has been exploited for more than 20 years and there has been a clear trend. While the first methods were always rule-based and only made use of textual features, like WIEN [9] or WHISK [8], more recent methods are Machine Learning based. New methods focus on generalization power, so they can be used also with unseen documents. One example is the CNN created by Gogar [10] that is able to extract information from websites, even if they haven't been seen before.

Efforts are focused on plain texts, websites or invoices, and the number of works on MRTD is clearly smaller. However, invoices are the closest kind of document to MRTDs so we can adapt the work from invoices to MRTDs and see how well they perform. We have multi-step approaches like PICK [15] or Chargrid [16], that have three individual stages: text recognition, document representation and entity extraction. On the other hand, we have end-to-end approaches that join these three stages into just one big one, that benefit from correlation between them.

3 Design of MRTD Layout Finder

The outcome of this project is a tool with three modules, described below in order of implementation, which is the same as importance/priority, as understood by InnoValor. In order to answer research question number one (RQ1) we need to find a way to extract the layout of an MRTD. We created a model that given an identity document, no matter the country or version, is able to automatically extract all the mandatory fields (the ones regulated by ICAO [1]) that appear in ID documents, like name, birth date and expiry date. It localizes (and classifies, to be able to interpret) the text fields in an identity document image, allowing an easy entity extraction. Thus, it has a good generalization capacity, extracting the fields in a country and version-independent fashion. It is described in detail in Section 4.2.

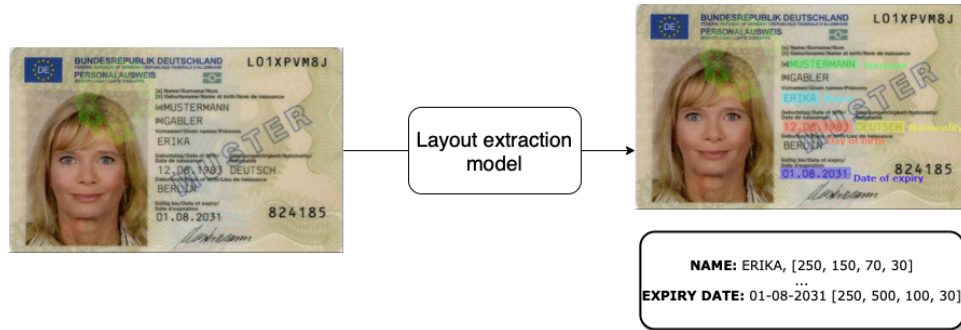


Figure 3: Example of layout extraction. Given an input image, the template is detected in the output image. Each field is detected in the image, giving its value and location. Source of ID image: PRADO [3].

Answering RQ2.1 we make use of the previous extracted template to protect private data. Some companies need to store identity documents respecting confidentiality constraints, following regulations like GDPR or any other that might apply, as part of KYC guidelines or similars. Thus, humans need to manually manipulate these images to protect this data. Our second module, a privacy protection module, will answer RQ2.1 by automatically erasing the specified fields that we would like to protect. More details can be found in Section 4.3. As an example, we can imagine a company that needs to store a copy of its customers' ID but it can't store the customer's doc number and name. In Figure 4, our module is fed with an input ID and a list containing the field "Document Number and First Name", and it outputs the same image but with the Document Number and First Name fields erased on the image, while all the other fields are kept as they were (we also keep picture and signature in this

case).

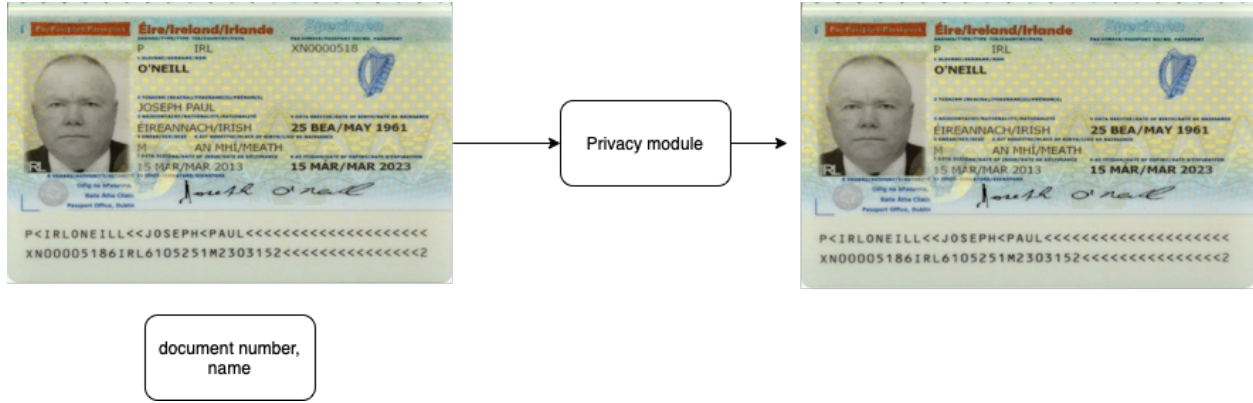


Figure 4: Example of the privacy module. Source of ID image: PRADO [3].

Finally, the third research question focuses on the synthesis of fake identities. Thus, the third module is a synthesis module of fake documents. Given a good layout of the document that is being processed, we create new samples of the same format. Knowing where each field is located, we can find the way to automatically substitute the text there with fake data, respecting the text format, so new ID documents are generated (skipping confidentiality problems allowing the use of a big set of documents for testing other applications). This module is described in Section 4.4. In Figure 5 we have an overview picture of our system and the result of each module.

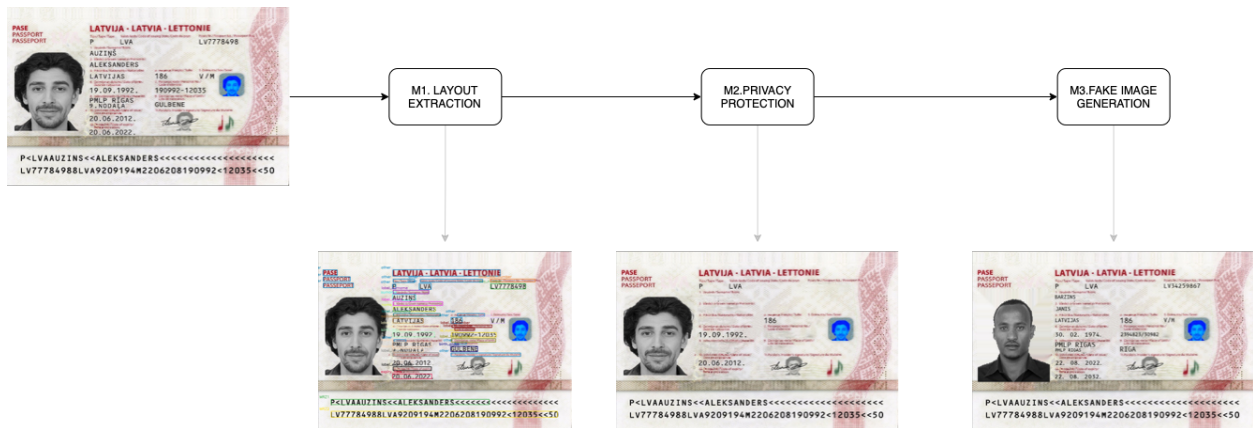


Figure 5: Overview of our system. Source of ID image: MIDV [3].

4 Method and Results

4.1 Datasets

To train and evaluate the models we will describe in this chapter, we need data, mostly pictures of different versions of MRTD documents and captured in different conditions. As presented in Chapter 2, most works used their own datasets. However, there seems to be one widely used dataset, known as MIDV-500 [5], and its more recent versions MIDV-2019 [24] and MIDV-2020 [30]. These, will be complemented with our sources, as summarized in Table 1.

Thus, we will be using the different versions of the MIDV datasets, mixed, so our dataset contains different images which have been captured under different conditions, trying to make our system more robust. The first two versions of this dataset contain 50 different kinds of synthesized documents, in both image and short video formats, with annotations of all text fields and their location. There are scans and images or videos taken in real-life scenarios, with different backgrounds and they are captured with mobile devices. The main change that the second version offers, is the use of newer mobile devices with better quality, and more recording conditions with more lighting variations [5] [24]. The newest version also uses better devices for capture but, instead of 50 different documents, it only has 10 kinds, with 100 identities per kind [30]. No details about the synthesis of documents process are given.

Since the main goal of our project is obtaining good generalization, we need a big variety of documents, which we can't obtain with the MIDV pictures. Hence, we decide to use document images from EU's database PRADO [3]. It is not possible to easily download pictures from the website, so we built a scraper tool to download all document images, filtering by the category of the document (passport, identity cards or others) and its kind (ordinary, diplomatic, temporary...). These 1.492 images are ideal ones, containing just the document in the center of the image. However, they don't have any annotations, so we manually labeled 131 different documents. For each experiment we used a different dataset and splitting, so they will be mentioned in each experiment's section.

Finally, in order to add some realness to our tool and continue the generalization ideas, for some small experiments (where it will be mentioned), we are going to use a set of images from real specimen documents owned by InnoValor, around 50 documents.

Table 1: Datasets summary

Dataset	Number of images	Number of different types of documents (or identities)
MIDV-500	500	50
MIDV-2019	500	50
MIDV-2020	1000	10
PRADO	1492 (131 labelled)	1492

4.2 Module 1: Layout extraction (RQ1)

In order to answer RQ1 we need to find a way to extract the layout of an MRTD. As we reviewed in Chapter 2, there are no other works that perform our information extraction task from MRTDs in a generalized way. Thus, the extraction of an MRTD’s layout in a generalized way (RQ1) is based on the adaptation of other works (on invoices) to our MRTDs.

Although some end-to-end strategies (like VIES [19] or EATEN [20]) were contemplated, we focused on multi-step approaches (like PICK [15] and CharGrid [16]). Given the structure of end-to-end approaches, a lot of data is needed for a proper training of a model and, in our case, that is a real constraint since we don’t have much varied data available. End-to-end strategies are a bit of a black box, while multi-step approaches allow a better analysis of the intermediate results.

We chose PICK [15] and CharGrid [16], because contrary to other multi-step approaches, these two use both textual and spatial relation together, in an explainable and simple way. Moreover, they extract features at a suitable level: just like a set of characters together. We wouldn’t benefit from more complex methods (like BertGrid [17] or VisualWordGrid [18]) that use word embeddings or take into account context, as our text is mostly composed by proper nouns, which can be ambiguous, or dates.

A general picture of the system, with the different steps of the pipeline, can be found in Figure 6. In our task, we have an image of an MRTD as input and our model will extract a template. We define a template as a text file containing a list of pairs entity-bounding box. This can be understood as a dictionary where the key is the name of the entity and

the value a list of bounding boxes defining their location inside the image. In pseudo-code, a template is defined as:

Template: [Key, [BoundingBox]]

A bounding box is an array containing 8 **double**: the x or y coordinate of the 4 corners, in the normalized range [0,1].

BoundingBox: List[double,8] = [top_left_x, top_left_y, top_right_x, top_right_y, bottom_left_x, bottom_left_y, bottom_right_x, bottom_right_y]

Next we find an example of the template that this Portuguese passport would have.

The three steps to answer RQ1 are:

1. **Preprocessing and text recognition.** As we will justify later, we need a first preprocessing step, to prepare the image to be properly processed in the next steps. We need the prior extraction of the text contained in an image. Traditionally, it has been done by OCR methods like Tesseract [13], but as suggested by [29] we will also try text recognition methods for real-life scenarios.
2. **Document representation.** Once the text has been recognized, we find the better representation that keeps the visual representation as well, so we facilitate the later extraction of the entities making use of both together. We try two methods for document representation: PICK that is graph-based and CharGrid that uses a grid of pixels.
3. **Entity extraction and Template Creation** In both cases, the process ends with the entity extraction performed with Bi-LSTM-CRF [14] layers, that allow the creation of a template.

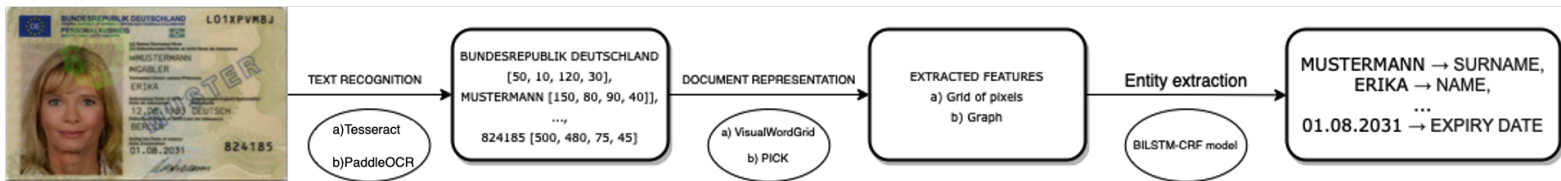


Figure 6: Diagram of the process of layout extraction for multi-step approaches. Source of ID image: PRADO [3].

4.2.1 Text Recognition and Preprocessing

We compared two different systems for the text recognition task: a traditional one which is Tesseract [13] and a state-of-the-art one which is PaddleOCR [31]. Tesseract OCR 4 is an LSTM-based engine. As any other text recognition tool given an image, it detects the bounding boxes containing text and recognizes the text inside of them. In order to get a good performance of Tesseract, it is necessary to get a clean black and white image (binarized). For this, we first remove noise using a median filter, deskew the image by means of finding the rectangular contour of the identity document and calculating the rotation matrix that turns it into a straight rectangle with no rotation. Finally, we turn the image into black and white, trying and evaluating different techniques. We have:

1. **Grayscale image.** We simply convert an RGB image into a gray scale one, with a standard method that calculates luminance based on the R, G and B values.
2. **Thresholded image.** Given a grayscale image, we binarize it using a determined threshold value. In our case, instead of an arbitrary one, we use the OTSU thresholding [32], an adaptive thresholding based on the histogram of the image.
3. **Open image.** Given the grayscale image, we perform an opening, a mathematical morphology operation, which consists of an erosion followed by a dilation [33].
4. **Canny image.** Given the grayscale image, the edges of the image are detected, using the Canny edge detector algorithm [34] which is based on the gradient of the image and two values that filter edges from non-edges.

An example of this image is found in Figure 7.



Figure 7: Examples of the different versions of preprocessed images. Source of ID: PRADO [3]

This preprocessing is implemented with functions from OpenCV, a Computer Vision library available for Python. Once the images are preprocessed we recognize the text using the official Tesseract implementation available in their github repository [35].

The second method we use for text recognition is PaddleOCR, a tool developed by PaddlePaddle based on, what they named, PP-OCRv2 [31]. It is a tool for OCR in real-life scenarios that uses several tricks to improve the training and system performance, allowing a high efficiency in a lightweight model. There is an official implementation of their tool available on github [36], with pretrained models. We used the most recent model called *Chinese and English ultra-lightweight PP-OCRv3*, as the authors suggest it should be the one used in multilingual problems.

In comparison with TesseractOCR, the authors say this method is more robust, as it is trained in real-life scenarios and it doesn't need the previously described color transformations preprocessing, they don't bring any improvement in performance.

To evaluate the text recognition system, we compared the performance of two OCR methods: PaddleOCR (*paddle*) and TesseractOCR. For the latter one, we created four different versions, depending on the kind of data that was used as input, one version for each different preprocessing method: grayscale image (T_{gray}), thresholded image ($T_{threshold}$), open image (T_{open}) and canny image (T_{canny}). The reason why PaddleOCR has a single version

is because we use as input the original files, because there is no need for preprocessing. In order to carry out this experiment, we use a selection of 50 ideal images from MIDV-2020 (those where the document is in the center and the only thing that appears). We decided to use these ideal images because these have the best annotations and they are taken in the best working conditions for both algorithms, the documents are cropped, so the image only contains the document in its center. We think this is the most fair choice, as Tesseract fails to detect text in real-life images, as we discovered during our first explorations. For each version of Tesseract, we use as input the same set of images, applying the corresponding preprocessing method.

4.2.1.1 Evaluation of text recognition

The annotations used are those offered by the MIDV dataset (what we call ground truth) against those obtained with the different OCR methods. MIDV datasets only contain some of the fields included in the ID, just the most basic ones and they don't include other text contained, like could be the sentence or label that introduces the value of the field below. On the other hand, OCR recognizes all text in the image. Thus, we will be mapping text in MIDV annotations to the OCR ones, ignoring other extra text. Given the nature of this experiment, we decided to come up with our own evaluation, with the parameters adjusted after some small experiments with trial and error. We calculate, limited by the amount of boxes predicted by the OCR, how many boxes are detected, how many boxes are correctly/wrongly predicted and incomplete (boxes that don't have a full overlap and are also missing some text). We complete this result with the precision value (true positives divided by number of predicted positive samples, constrained by the number of boxes detected).

To evaluate, we match each ground truth text box with the equivalent predicted box. In order to do so, we iterated through the ground truth boxes, finding the predicted box with maximum overlap (calculating the intersection over union). Two boxes are mapped if they have an overlap bigger than 1% and said to be equal if their texts have an edit distance lower than 3 (for strings longer than 3 characters), we consider the predicted box as correct.

4.2.1.2 Results of text recognition

First, we have the results of the comparison of the different recognition systems for our evaluation set, containing 50 images and a total of 325 boxes with text. This evaluation is defined in terms of number of detected boxes (*detected*), correctly classified (*correct*), wrongly classified (*wrong*) and precision. It can be found in Table 2.

Table 2: Comparison of different text recognition systems

	Detected	Correct	Wrong	Incomplete	Precision
Gray	217	210	2	5	97%
Threshold	222	214	2	6	96%
Open	205	192	5	8	94%
Canny	142	107	15	20	75%
Paddle	324	322	0	2	99%

In this table we can see that there is a system that performs clearly better, PaddleOCR, not only its precision is higher than any Tesseract system but it also detects more boxes (50% more than the best Tesseract system), holding a greater recall. We could expect this result as PaddleOCR is a more recent method that uses state-of-the-art features and it is meant for real-life scenarios. Although these images are not in real-life scenarios, the complexity of MRTDs backgrounds mimics this complexity that Tesseract doesn't know how to handle, failing to detect boxes and doing it with worse quality. These worse performance is probably also due to the fact that the binarization obtained is never perfect (due to the complex backgrounds) and that can easily affect the text recognition. Since PaddleOCR doesn't need this previous imperfect preprocessing step, its performance is better.

However, we had to add a new feature to the preprocessing step. As discovered during a later experiment where we had already decided to work with PaddleOCR, in order to get the best performance without leaving out many text boxes undetected, we need the image to contain the document as the main part of the image. If the document is not the main focus, the text recognition fails, as we can see in Figure 8, where it improves substantially by just cropping the image.



Figure 8: Example of an original document image with no recognized text and the cropped version on the right where text is recognized. Source: Adapted from MIDV-2020 [30]

Consequently, we had to build a tool that can automatically detect and crop a document inside the image, so we can use all the images available in our datasets and be ready for any scenario in real life.

4.2.2 Document cropping

Document cropping, as reviewed by [2] is typically based on contour detection. This can be done in a traditional way, based on Canny edge detection or using the Hough Transform, or in a more recent way, like HoughEncoder [37], where they use a neural network that performs an image segmentation task, mimicking the Fast Hough Transform.

Inspired by the latter one, we will crop our document by detecting the document with image segmentation so we can extract 4 corners used to warp the perspective, to get rid of any distortion. In Figure 9 we can see the general process, described below.

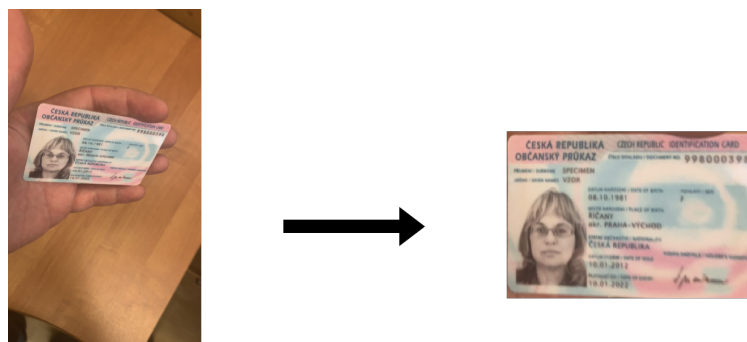


Figure 9: Example of document cropping using our tool based on image segmentation. Source: Adapted from MIDV-2020 [30]

Among a long list of models for semantic segmentation, we chose to use the SegFormer [38] model, as it is a simple one that doesn't need a big training set and had an available implementation.

1. **Mask detection.** The first step is to calculate the mask containing the identity document. We use a SegFormer model [38], an encoder-decoder network that unifies transformers and multi-layer perceptron, it was chosen because it is a simple model that doesn't need a big training set and it already had an available implementation. Another model, U-net [39] was tried and we compare its performance with SegFormer in the results sections.
2. **Corners detection.** Given the mask, we need to extract the 4 corners coordinates. These are calculated by means of the Hough Transform of the image. Starting from all lines in the images, these are classified into two groups: horizontal and vertical, so we can calculate the intersection of each horizontal line with the vertical ones. These intersections are clustered with the k-means algorithm into 4 groups, which should correspond to the four corners of the documents. The centroid of each cluster is used as the corner coordinates.
3. **Perspective Warp** Given these four points and the height/width of the document, we can construct the perspective transformation.

For evaluating this task we built a training dataset containing 1000 images from MIDV-2019 and 5000 images from MIDV-2020. Afterwards, we evaluate it with a set of 1000 never seen images from MIDV-500. All images contain identity documents in different real-life scenarios, captured with different phone devices and lighting conditions.

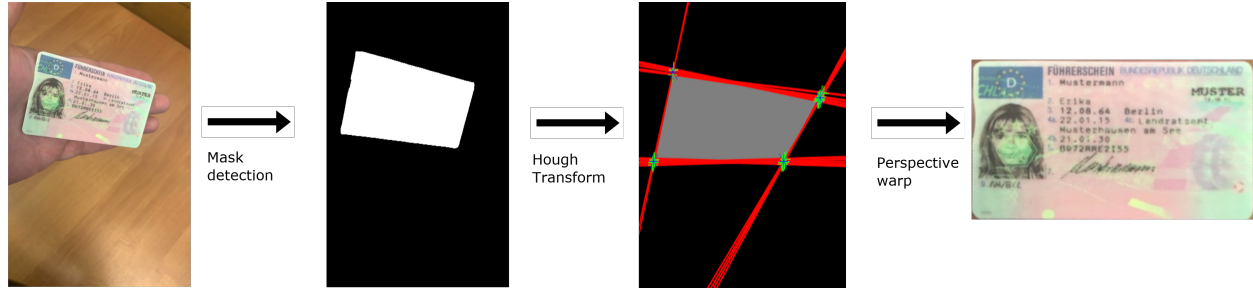


Figure 10: Step-by-step example of document cropping. Given an image, we detect the mask (white) using a SegFormer model. Later we obtain the Hough Transform, to calculate the lines (red) and their intersections (green), which are clustered into four groups with a centroid, chosen as a corner (blue). Finally with a transformation matrix we warp the image. Source: Adapted from MIDV-2020 [30]

Apart from SegFormer [38], we also used U-net [39] and compared their performance. In order to do so, we realized that there was no use in trusting the Intersection over Union (IoU) of the models, as they were both very close to 1, on average. The IOU is an evaluation metric typically used to measure the accuracy of object detectors, by calculating the overlapping between the ground truth bounding box and the predicted one. Thus, after some manual evaluation of small experiments, we saw that a reliable and numerical measure was the accuracy of mask detection (number of images where a mask is detected, divided by the total, keeping in mind that all images contain a document). We consider an extraction correct if we can detect 4 corners inside the image. These points, that will correspond to corners are points that are reasonably separated in space (all corners have an euclidean distance of at least 100 pixels between them). If any corner is less than 100 pixels away from another there is probably some error in the corner extraction, it is not right, or, simply, the document inside the image is too small to obtain a clean image with high enough resolution. These parameters were chosen manually after some small adjustments with trial and error.

4.2.2.1 Results for Document Cropping

We compare the performance of U-net model with the SegFormer one, in terms of percentage of good quality masks, those that allow extracting 4 corners. We can find the results in Table 3.

Table 3: Comparison of different mask detection models

	Correct	Wrong	Accuracy
U-Net	605	395	60,5%
U-net_normal	541	459	54,1%
U-net_augmented	640	360	64,0%
SegFormer	905	95	90,5%

From the table we can see that, with no need for normalization or data augmentation, SegFormer is the best model to perform this task, as it reaches an accuracy of 90.5% for quality mask extraction, 26% higher than the best U-net model. Moreover, this model does not need normalization nor benefits from data augmentation, as it always held similar results. Although both models create masks that detect most of those parts of the image containing the document (they have indeed a great IoU of the ground truth mask and the predicted one), SegFormer's mask are more useful in our case because they learnt the shape of the document, they are more similar to rectangle, as we see in most examples. Although they might not cover the whole image, and contain holes inside, the contour is always close to a rectangle like the one of an MRTD, making easy drawing straight lines. On the other hands, the masks of U-net have more irregular shapes, made of curves instead of straight lines, complicating the detection of lines with Hough Transform. In Figure 11 we have examples of extracted masks that are not perfect, seeing how different in quality and shape the masks of both models are.

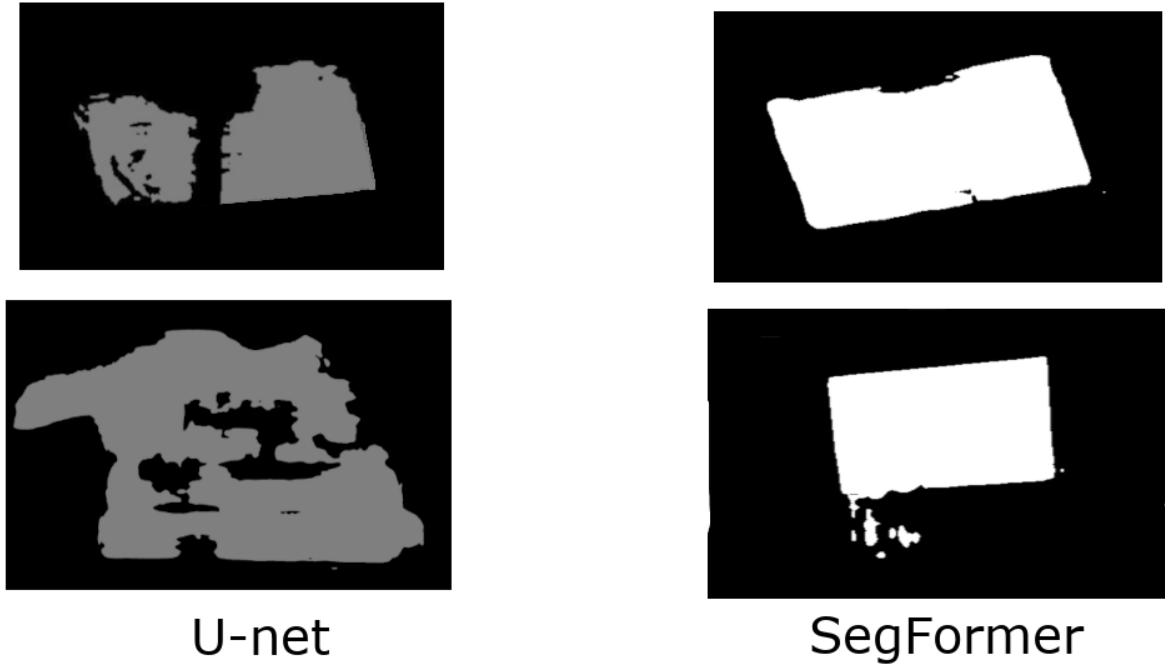


Figure 11: Different examples of masks extracted with U-net (gray) and Segformer (white)

4.2.3 Document representation

Once we have the text and image, we look for the best representation that is able to exploit their correlation in order to perform the best entity extraction. Thus, we try two different methods: PICK [15] and CharGrid [16].

PICK is a framework that combines graph learning with graph convolution operation, to obtain a richer semantic representation of both textual and visual features. As we can see in Figure 12, PICK contains 3 modules:

1. **Encoder.** Encodes text segments with a Transformer (to create text embeddings) and image segments (for image embeddings). These are combined into a representation that becomes the input for the graph module.
2. **Graph module.** Performing a graph learning convolutional operation, catches the best latent relations between nodes, to get richer graph embeddings.
3. **Decoder.** Using the graph embeddings, the module performs sequence tagging using

BiLSTM and CRF layers. This model includes the key information extraction (as a sequence tagging task) as part of the document representation task.

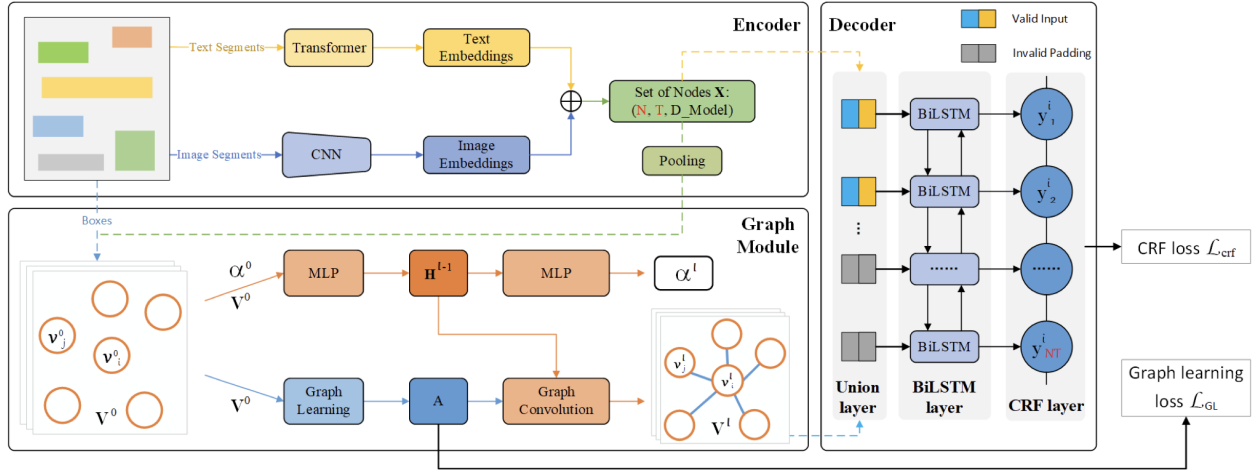


Figure 12: PICK's overview [15]

The parameters of the model are the ones suggested by the authors in [15], as the model converged fast enough and performed well. Thus, we respect the dimensions of the encoder/decoder, the dropout rates and learning rates. We only changed the number of epochs to 100 with an early stopping of 40 epochs. We also set a fixed image input size of 760×480 .

The overall dataset used is composed by two different kinds of data: 900 images from MIDV-2020 (9 different kinds of documents times 100 identities per kind, leaving out the Russian passport that used the Cyrillic alphabet and made the evaluation and comparison of strings more complex) and 131 randomly selected images from PRADO's database (containing 131 different versions of identity documents, mostly specimens). These are randomly split into 65% for training and 35% for testing.

During the experimental preparation, we noticed an annotation mismatch between MIDV and PaddleOCR. MIDV training annotations differed too much from the ones obtained with PaddleOCR at testing. MIDV annotations only contained the value of the mandatory fields in the document, whereas PaddleOCR extracts all text inside the image: thus, it also extracts irrelevant text or the labels that introduce each data field. Thus, we decided to use PaddleOCR transcripts, which are the kind of transcripts that would be used in production in a real-life application. We manually updated the MIDV annotation to include all text in an image, like PaddleOCR does. For example, given a German Identity card like the one in Figure 3, the MIDV annotations missed non-relevant text like the *BUNDESREPUBLIK*

DEUTSCHLAND on top of the image, or all the labels on top of the data values, like the *[a] Name/Surname/Nom* above *MUSTERMANN*. We think these label texts, can be very beneficial for our model, as they give hints of which data fields are next to them and that is why we decided to include the labels as part of the entities we want to extract.

We built the list of entities to extract based on the mandatory fields for MRTD that ICAO defines in its document 9303 part 5. Thus we have 11 value entities, their 11 labels (text that introduces them), 3 entities that don't have labels, each of the lines in the Machine Readable Zone and a last entity for *other* fields. These are all listed in Table 4.

Table 4: List of entities.

Value entities	Labels
Issuer (institution)	Label issuer
Name	Label Name
Surname	Label Surname
Gender	Label Gender
Nationality	Label Nationality
Birth date	Label Birth date
Document number	Label Document number
Expiry date	Label Expiry date
Birth place	Label Birth place
ID number	Label ID number
Issue date	Label Issue date
MRZ1	
MRZ2	
MRZ3	
other	

The second method for document representation is CharGrid [16], that encodes each document as a two-dimensional grid of characters, in order to preserve the 2D layout of a document. A document of size $H \times W$ is represented by a grid with $H \times W$ cells, where each of them contains the encoding of the character in that pixel, or 0, if there is no character in that position. Given this grid, a fully convolutional neural network (with an encoder-

decoder structure) performs semantic segmentation of the grid, predicting the class label for each character pixel, further performing an instance segmentation to merge different characters into the same entity, by predicting bounding boxes. This architecture is shown in Figure 13.

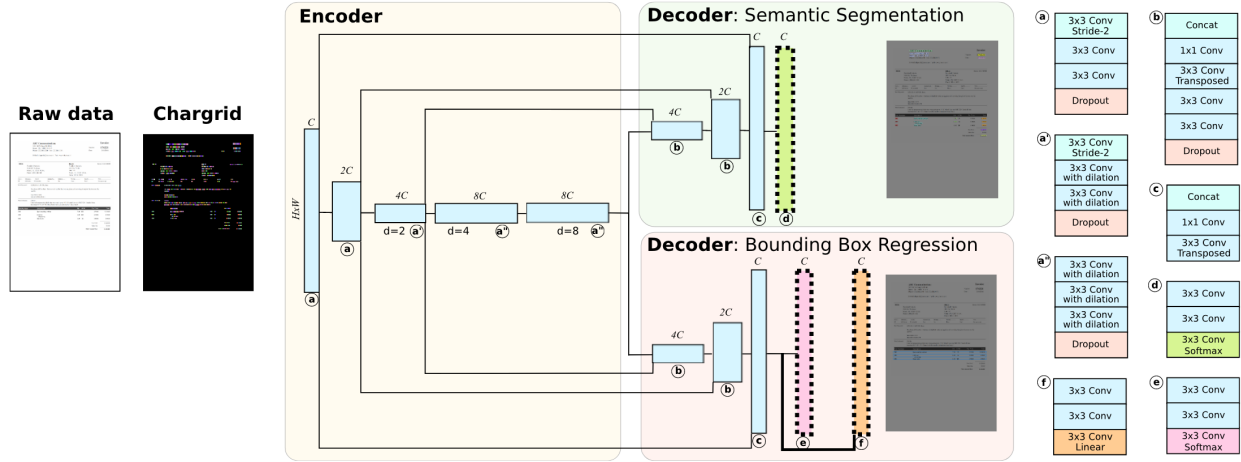


Figure 13: Chargrid's architecture. Source: [16]

In order to answer RQ1.1 we need to evaluate our extracted templates and this is based on the evaluation of the performance of PICK [15] and Chargrid [16], in terms of entity extraction. Thus, for each of the methods we tried, we compare confusion matrices (to see which entities create problems) and we calculate the precision, recall, F1-score and number of entities extracted (support). This way, we are able to compare different strategies. For this task, we use a dataset of 1031 pictures, of which 65% (670) are for training and 35% (361) for testing. These images are a random mix between MIDV-2020 (900) and PRADO (131) images, with pictures which hadn't been seen by the model before. Although some images might contain the same version of a document, all identities are unique, there are 1031 different identities.

4.2.3.1 Results of entity extraction (RQ1.2)

By evaluating the performance of each model's entity extraction we are answering RQ1.2 as this will be the way to assess MRTD templates. Unfortunately, we will only review the results from PICK as no matter how we played with the parameters of the CharGrid model, it never converged and no results were possible to extract as no defined bounding boxes were created. It could be worth the try, building our own implementation or adaptation of

CharGrid, to verify whether the model is not suitable for this task or there is some error in the used implementation.

For PICK, we have 3 different models, based on 3 different kinds of IOB (Inside-Outside-Beginning) tagging, which is the way of tagging tokens (or words in our case) and that can be: box level (*box*), document level (*doc*) or a mix between the two of them (*band*). This IOB tagging describes the level at which words with the same tag are merged together during the entity extraction step. Thus, *box* level won't allow the merging of two words from two different boxes, so we could have multiple instances of the same entity. On the other hand, *doc* level does allow merging words from different boxes into a single instance, so we only find one instance per entity. Finally, the *band* level is a mix of the two, allowing multiple instances and merging between different boxes.

This merging can really affect our results, as we post-process the results from PICK, which only consist of the entity name of the value, with no bounding box information. We need to map (in a way we designed) the output entities to the input text, to find their bounding boxes, and different merges in output can result into different mappings, with wrong entities locations.

In Figure 18 we can see a complete review of the test metrics containing precision, recall f1-score and support (number of entities extracted) vary along epochs for each of the 3 models). We recorded the values at epochs 20, 40, 60 80, 100 and what the algorithm detected as the epoch with best f1-score for training. We should note that in case of the *doc* model, the model trained for a 100 epochs, while the other two only trained until epoch 80, we used the early stopping as no improvement was obtained (this is clear if we take a look at the yellow and orange graphs, they reach a peak at epoch 40 and never improve substantially after that point).

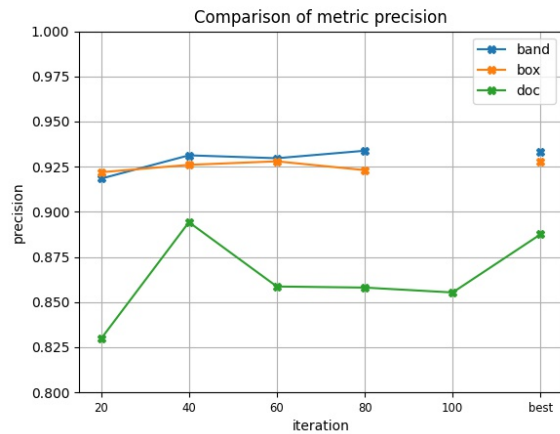


Figure 14: Precision of the 3 PICK models

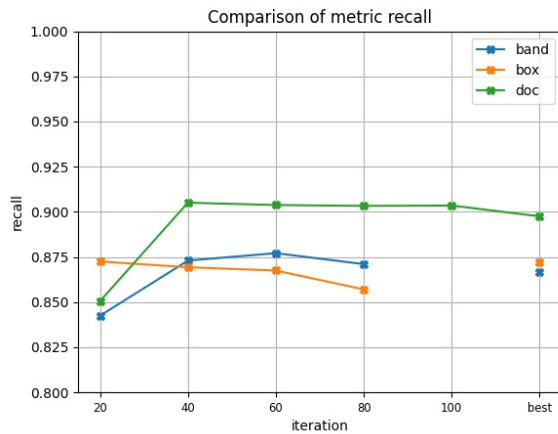


Figure 15: Recall of the 3 PICK models

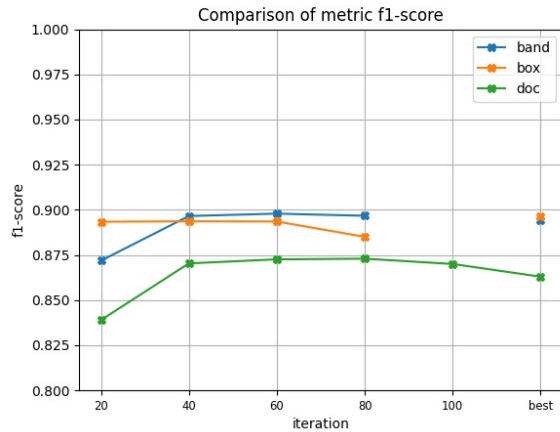


Figure 16: F1-score of the 3 PICK models

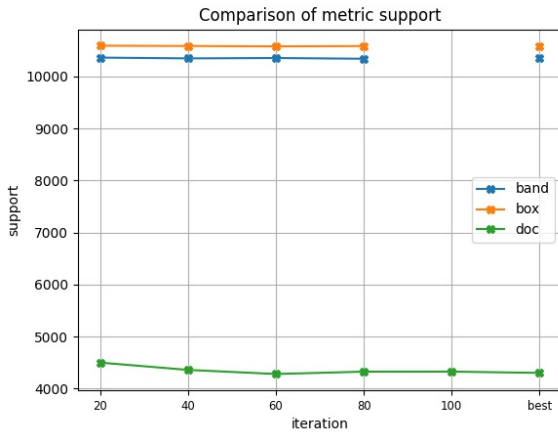


Figure 17: Support of the 3 PICK models

Figure 18: Results of PICK model

The first thing to highlight from these plots is that the support (number of entities extracted) in the case of the document-level IOB tagging, is clearly lower than for the other two models (a 50% lower). This is important to take into account, as we would like to extract as many fields as possible, always making sure that an increase in the number of entities extracted doesn't result into a decrease in accuracy. Given the nature of our task, the Precision metric is the most important one. We want to know the proportion of predicted positives that is truly positive. For instance, if we classify a field as *document number* it better not be a *personal number* as, despite having similar format, they should not be confused. However,

if we don't extract a class, if we can't find the *personal number*, it is fine, (if we have high precision and we can trust the other decisions) we will know that it was not possible to label it automatically and we just need to review it manually.

Since the *band* PICK model (with mixed box and document IOB tagging), has the best results we will now compare the evolution of its training and testing precision and support throughout the epochs.

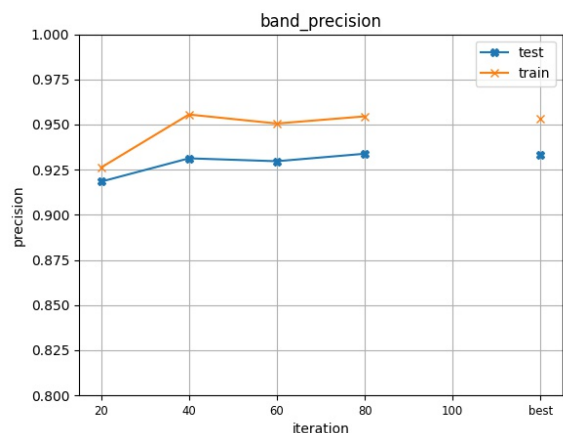


Figure 19: F1-score for PICK band_model

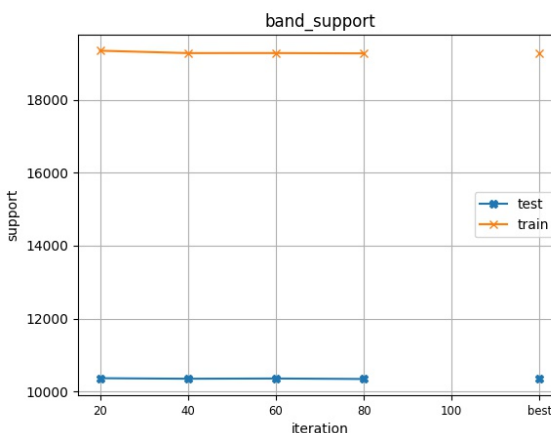
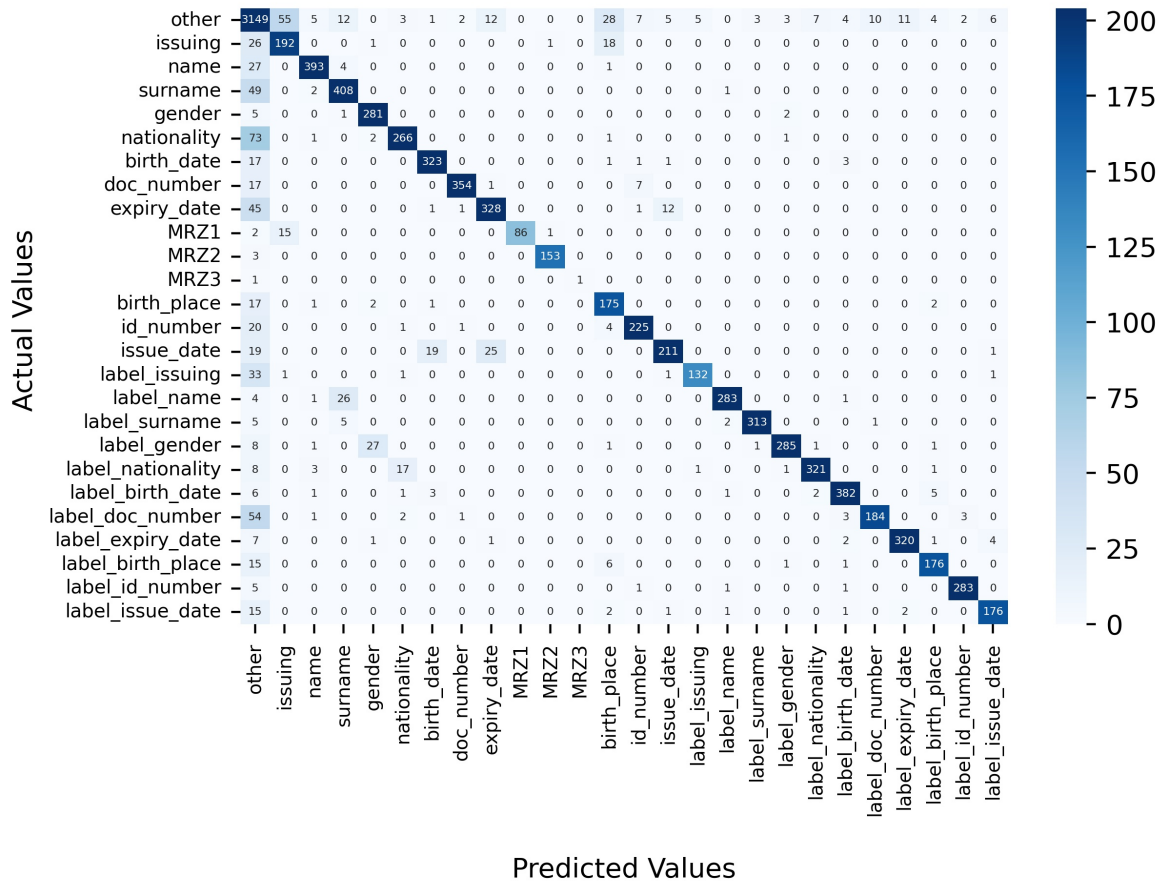


Figure 20: Support for PICK band_model

It is worth noting that, despite having a clear difference in train and test support, this difference stays the same. The same happens with the precision, despite being small for epoch 20 (because the model is still underfitting), the rest of the epochs have a similar gap between train and test precision. Thus, we can conclude that we are not overfitting as this small gap of 2% doesn't increase with epochs. Finally, if we were to choose a specific model from the *band*, we are probably better off selecting the model of epoch 40, to continue with the early stopping strategy and make sure that our model is not overfitting our data. The difference in performance between epoch 40 and the best one (for precision, number 80), is really small so ensuring that there is no overfitting is more beneficial than earning a 0.2% improvement in precision. The precision of the model at epoch 40 is 93%.

The next step in the evaluation of the model is analyzing errors. For this, we will be using the confusion matrix of the epoch 40 *band* model, which can be found in Figure21.

Figure 21: Configuration matrix of the *band* model in epoch 40.

We should first note that the diagonal of the matrix is full of high values, in line with the results seen before, which promised a high precision of 93% (a good value for a task with 27 different classes). We should ignore the fact that the field *MRZ3* has a value of 1 in its diagonal, as if a field that doesn't appear often (only two times in our dataset), as it only appears in the back of some identity documents (and for most documents we don't have a back picture). We can also see that most errors happen for the field *other*: different values are predicted as *other*, the opposite direction is also true, but in a smaller magnitude: values that are truly *other* are predicted as different fields. Apart from this, the main error we have is *other* values predicted as *birth_place*, for which we couldn't find a clear explanation, yet. Afterwards, we have *label_gender* and *gender*, which could be explained because of their close location and the fact that the text for the label (usually sex) is a short word, like the values of the gender field. We also have *label_name* and *surname* fields that are in close-by locations and that are nouns that can change a lot depending on the language. On a similar note,

we have the problem with *label_nationality* and *nationality*. Finally, we should comment a very well-defined error, those with dates. Although the systems seems capable of learning the format to represent dates (it detects dates), it doesn't always give the class the correct label. For example, we see that 25 *issue date* were predicted as *expiry date*, the system detected it was a date but is not capable of learning a higher level relation between them, it is probably not aware of the fact that the issue date is always more recent that the expiry date. Fortunately, this is a mistake that could be easily fixed with some postprocessing. Given the fields that are dates, they can all be reviewed and ensure that the first in time is the birth date, then the issue date and finally the expiry date.

4.2.4 Template creation

The last step to answer RQ1 is the creation of a template. Once an image is processed by Module 1, we obtain a list of: contained entities, their bounding box (location inside the image) and the transcription or value of the field. In order to build the layout of an image we can drop the transcription, as it will vary from one document to another, and just keep the name of the entity and its bounding box. We decide to store each of these templates (one for each image) as a plain-text file: containing a set of tuples with the name of the entity and its bounding box. A visual example of an extracted template can be found in Figure 22.



Figure 22: Example of template extraction, showed on top of a document. Source original ID: MIDV-2020 [30]

4.3 Module 2: Privacy protection (RQ2.1)

To answer RQ2 about using extracted templates, in RQ2.1 we look for the way to use the template from RQ1 in a possible application: protection of sensitive data. Thus, we answer RQ2.1 by using the image of an identity document and its textual template, to remove (with inpaintings) selected value data fields.

We can easily conceal the fields from the image by retrieving its bounding box from the template and inpainting that part of the image with OpenCV. Inpainting is the process of restoring missing or damaged areas in an image, in an unnoticeable way [40]. In our case, we will inpaint those areas containing text to look like the background, as if the text was the missing part and had to be covered in a way similar to the background. An example can be found in Figure 24, where the fields *nationality*, *document number*, *place of birth*, *date of birth*, *sex*, *date of issue* *date of expiry* and *id number* have been concealed with inpaintings.

We tried different kinds of inpaintings, with different combinations of 3 parameters: inpainting algorithm, mask type and mask dilation.

1. **Algorithms.** They define how the new values are computed, we use three different algorithms: one based on Fast Matching Method, created by Telea in 2004 [41] (that we will name **telea**); a second one based on fluid dynamics and partial differential equations, created by Sapiro in 2000 [42] (**sapiro**) and a third one based on the Fourier Transform of the image, Frequency Selective Reconstruction (**FSR**), created by Seiler et al. in 2015 [43].

2. **Mask type.** The mask defines the area of the image that will be inpainted. With manual inspection, we saw that 3 kinds of masks (the ones described below) held better results than any other mask we had tried (like masking the whole bounding box). In our selection of 3 masks we have 2 types of mask depending on the area they define to be inpainted (what we call *charmin* and *charbox*) and a third one that additionally restricts the area of the image that can be checked to complete the image (*restricted*). *Charmin* is the minimum possible mask, as it only selects the lines that form the character. By thresholding the image, we keep all text in black and get rid of the background that turns white. We will mask all black pixels, that should belong to the text characters. *Charbox* selects all the boxes that contain a character. Given the previous mask, we draw a contour for each character and we fit it into a square, so for each character a whole square will be selected. *Restricted* uses the mask from *charmin* but restricts the area of the image that can be checked out to compute the inpaintings. We perform the inpainting using a portion of the image, defined by the bounding box of the text, so just a small surrounding area and not the whole picture is used. This ensures that undesired elements do not contaminate the inpainting, as can happen in the previous mask types. Choosing the mask type is a bit of a trade-off, as although we might desire a very strict mask, to get an unnoticeable inpainting, we might be leaking sensitive data that was originally present.

3. **Mask dilation.** To ensure that we inpaint all text and no text pixels are left out, we can perform the morphological operation of dilation to the mask, so it grows in size and more probably covers all text. We try no dilation (dilation 0) and a dilation with a square kernel of 3×3 pixels.

An example of these types of mask is found in Figure 23. We have a branch for each type

and each branch contains: a mask image that with white pixels indicates the part of the image that will be inpainted (the text to be removed) and the image used to inpaint (the image that will be used as reference to fill the pixels indicates in the mask, with information from the others). In the case of *charmin* the mask is just the inverted thresholded image, for *charbox* we draw contours into this and fit them into squares that are later filled and for *restricted* the difference is the image used to inpaint, we cover all image except for the bounding box of the word, so no external elements are used to inpaint.

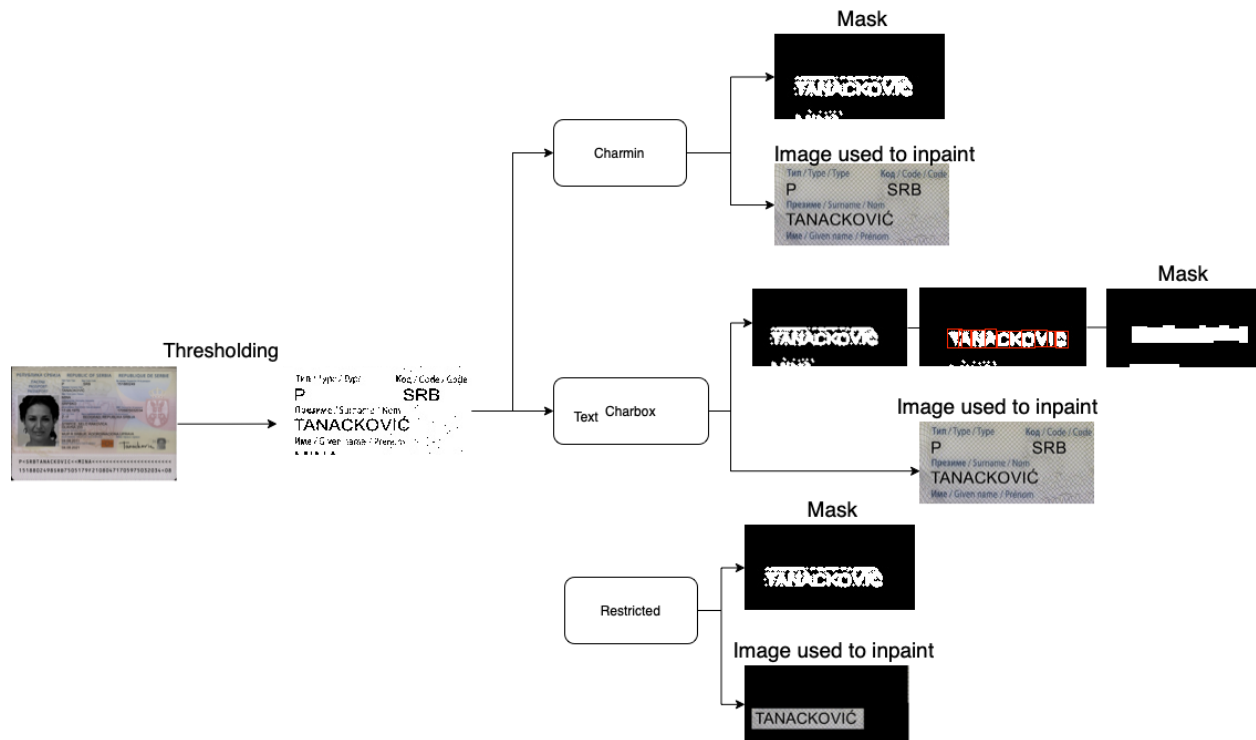


Figure 23: Different types of masks used in inpaintings. Source original ID: MIDV-2020 [30]

To evaluate this Module 2, and find an answer to RQ3 (to evaluate this application) we try to rank the different inpainting methods, check which one performs better. Unfortunately, this answer is not straightforward. Thus, we did some research on image similarity and image quality evaluation, to find the best possible answer.

Although we have no reference images for these inpaintings, there is a model called NIQE (Natural Image Quality Evaluator) [44] that is able to automatically predict the quality of distorted image, as perceived by an average human. This model assumes that only the distorted image is available, it needs no reference and it is also opinion unaware (there is no need to train it to fit some human-based scores). This model is based on a collection of

quality-aware features fitting a multivariate Gaussian model, derived from a natural scene statistic (NSS) model. These Natural Scene Statistics include several measures like means or correlations. The quality of a test image is a measure of the distance between the model statistics and those of the test image.

We will evaluate the inpaintings and the later document synthesis with a human jury that will answer a questionnaire, containing 36 images with different inpainting settings, as explained in detail in Section 4.5. Additionally, they will also be evaluated explicitly with the document generation task and an automatically calculated metric, as explained in Section 4.4, trying to find an answer for RQ3.1.



Figure 24: Example of privacy protection. Source original ID: MIDV-2020 [30]

4.4 Module 3: Fake image generation (RQ2.2)

4.4.1 Methods for Fake image generation

To obtain a more complete answer to RQ2, we look for a second application: synthesis of fake documents. We are answering RQ2.2 to see how to automatically synthesize the best possible fake documents, given a template and a sample image. We created a third module that allows the automatic creation of fake identity images, given an example document, the template extracted from RQ1 and the set of fake text values.

Given an input image, we retrieve the location of all fields from the template, so we can inpaint the areas belonging to data fields (erasing any trace of the original text) and keeping areas with no text or label fields, using Module 2. This results in what we call a clean image, used as background images.

We will need to superpose a new foreground, containing the text of the fake identity. In a nutshell, Module 3 is an extension of Module 2. Given an image with all data fields inpainted (clean image), we add new text in the best possible way, as similar as possible to the original one and with no obvious traces of manipulation.

This added text must be as similar to the original as possible, in terms of format. To do this, we estimate the font size given the size of the bounding box and we use the most similar possible font, as identified by the tool provided by the API of [whatfontis](#) [45], a website that identifies the font of the text in an image. In general, fake identities will be used for the context of this text, by using common names and surnames or random dates.

However, not only the content of the text defines an identity, but also the portrait picture included. It needs to be erased, so it can be later substituted. To erase a portrait, we detect the area of the image containing it and inpaint all the area, to ensure we don't leak the original portrait. In order to detect the portrait we use the method suggested by Tavakolian in [2] which first detects the face (with a traditional method like Haar Cascade or a more modern one, like a DNN), giving a small face boundary box, which needs to be expanded to cover the whole portrait, respecting the image ratio of $4/3$, a round number obtained

from the suggested portrait dimensions suggested by ICAO in their document 9303 [1]. An example of this can be found in Figure 25.



Figure 25: Example of expansion of the face box. Source of ID image: PRADO [3]

In order to expand the face boundary box into the portrait bounding box, we use the following equations, adapted from the ones suggested by Tavakolian:

$$W_{portrait} = 1.6 \cdot W_{face}$$

$$x_{left_portrait} = x_{left_photo} - 0.3 \cdot W_{photo}$$

$$x_{right_portrait} = y_{right_photo} + 0.3 \cdot W_{photo}$$

$$H_{portrait} = 1.33 \cdot W_{portrait}$$

$$y_{top_portrait} = y_{photo} - 0.25 \cdot H_{portrait}$$

$$y_{bottom_portrait} = y_{photo} + 0.85 \cdot H_{portrait}$$

We expand the box width a 30% to each side and the height 25% up and 85%down. We inpaint the area of the portrait and overlay it with the new portrait image, turning it into grayscale, as most documents contain grayscale images rather than color ones. This method is very simple and generalizes over the picture size and ratio a lot. Thus, it fails sometimes to cover the whole portrait.

Since the task performed by this module is very similar to the previous one their evaluation goes together.

Since the generation of fake documents strongly depends on the previous module of document protection, because we inpaint all fields in the image and later simply insert text, they will be evaluated together. We should also take into account that our text insertion has fixed settings and there is no variance or parameters we can adjust. Thus, we include the same kind of questions for inpaintings adapted to fake documents. We ask people to give a numeric score from 1 to 5 to automatically synthesized fake documents, based on overall quality and information leak.

To answer RQ3.1, we found a possible automated measure to evaluate synthesized documents. We calculate image similarity between an original image and our synthesized one that recreates the original, using the same identity, forgetting about privacy protection for this experiment. Since we are not using real identities, this is not a problem for the task. We will use the Multi-Scale SSIM (Structural Similarity Index Measure) [46]. MS-SSIM is an advanced measure of SSIM that performs the calculations over multiple scales by sub-sampling the image. All these measures are calculated per channel and averaged afterward. The equation to calculate the SSIM between two (windows of) images is:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

With μ_x the average of x , μ_y the average of y , σ_x^2 the variance of x , σ_y^2 the variance of y , σ_{xy} the covariance of x and y , $c_1 = (k_1L)^2$, $c_2 = (k_2L)^2$ two variables to stabilize the division with weak denominator, L the dynamic range of the pixel values and $k_1 = 0.01$ and $k_2 = 0.03$ by default.

Additionally, we will also be calculating the NIQE score of recreated images, so the perfor-

mance can be compared to that of MS-SSIM and that of NIQE of protected images.

To automatically evaluate the recreation of documents, we evaluate the same inpainting settings that are used during the final human jury evaluation, which is described in Section 4.5. This is a set of 6 different configurations by adjusting: mask type (*restricted*, *charmin* and *charbox*) and mask dilations (0 or 3). As explained later, the inpainting algorithm is set as *telea*. Each of these settings is evaluated using six pictures, each of them coming from a different dataset (MIDV, PRADO or a picture taken by us of a specimen document) and being a different kind of document (ID or passport).

4.4.2 Results of automated evaluation of protected and synthesized documents (RQ3.1)

We now check the performance of automated measures, which is the answer to RQ3.1. We have two cases: privacy or data protection (simple deletion of text) and document synthesis (protection and later addition of text). For both cases we use a set of 36 pictures, changing: mask type (*restricted*, *charmin* or *charbox*), mask dilations (0 or 3), the kind of picture (*id* or *passport*) and the database they come from (*MIDV*, *PRADO* or *specimens*). We will compare their performances in terms of the mean score of the images for each configuration (for a set inpainting algorithm, mask type and dilation).

First we present the results from the privacy protection part. For each of the 36 images we calculate their NIQE score, and we average the results, grouping by inpainting algorithm, mask type and dilation, averaging for different kind and databases. The results, with the best configuration highlighted in blue, are presented in Table 5.

Table 5: NIQE scores of images with data protection

inp_algorithm	mask	dilation	text	NIQE mean	NIQE std
telea	charbox	0	0	9.756	3.417
telea	charbox	3	0	9.732	3.492
telea	charmin	0	0	9.916	3.135
telea	charmin	3	0	9.824	3.250
telea	restricted	0	0	9.903	3.147
telea	restricted	3	0	9.839	3.191

Next, we have the NIQE metrics for the images that recreate fake identities in Table 6

Table 6: NIQE scores of images with recreation of fake identities

inp_algorithm	mask	dilation	text	NIQE mean	NIQE std
telea	charbox	0	1	9.398	2.959
telea	charbox	3	1	9.444	3.015
telea	charmin	0	1	9.433	3.003
telea	charmin	3	1	9.454	3.038
telea	restricted	0	1	9.417	3.003
telea	restricted	3	1	9.429	3.069

Finally, we have the results of the MS-SSIM score between our synthesized fake images and their originals in Table 7.

Table 7: NIQE scores of images with recreation of fakes identities

inp_algorithm	mask	dilation	text	MS-SSIM mean	MS-SSIM std
telea	charbox	0	1	0.929 6	0.021 1
telea	charbox	3	1	0.928 9	0.021 3
telea	charmin	0	1	0.933 7	0.020 5
telea	charmin	3	1	0.930 3	0.021 3
telea	restricted	0	1	0.934 1	0.020 4
telea	restricted	3	1	0.931 0	0.021 3

First, we will discuss the results from the NIQE score, a distance metric. Thus, the lower

the score, the better the image quality is. In both cases, simple data protection and fake image recreation, the best score is that obtained with the mask *charbox*, the one that inpaints a whole square for each of the characters. The only difference between protection and fake recreation is that for protection, dilation 3 holds better score than 0, that gets the best score for recreation. However, we must keep in mind that these results have a large standard deviation (around a 30% of the mean). This could mean that the specific inpainting algorithm, mask type and dilation we are using aren't really affecting how the overall picture is evaluated, but just other factors like could be the kind of image or its database. Another explanation could be that the standard statistical model used for NIQE is trained with natural scenes (like could be landscapes or pictures of humans or animals) so it is not suitable for identity documents. In order to use it properly, we could train our own model, based on identity documents, but that would need a score reference.

We can see the best pictures according to NIQE in figure 30

Example of protection *charbox* dilation 0Example of protection *charbox* dilation 3Example of recreation *charbox* dilation 0Example of recreation *charbox* dilation 3

Figure 30: Examples of images inpainted with *telea* algorithm, showing why the configuration gets the best NIQE score.

Figure 30 shows why dilation 0 gets a worse score in protection. For example, the background for dilation 3 is more smooth, there are no color changes, whereas dilation 0 has some of them, which makes the image look a bit less natural. However, when we insert text on top of it, we can't really see those strokes of different colors, and the score becomes better for dilation 0, although the difference with dilation 3 is not big.

In terms of MS-SSIM it's the *restricted* mask with dilation 0 that gets the best score, almost the same as the *charmin* with dilation 0, examples can be found in 35.

*restricted* dilation 0 (1st place)*charmin* dilation 0 (2nd)*charmin* dilation 3 (bad score)

Original Serbian passport

Figure 35: Examples of images and their MS-SSIM qualitative performance

MS-SSIM score is a bit more explainable, as it gives a score of similarity in the range 0 to 1, where 1 means equal images. We can check it given the bottom right image in Figure 35, containing the original picture. The two pictures in the top row, that have a higher MS-SSIM, are similar to the original one than the bottom left one. That image has a dark blur on the *Place of issue* field, making it look more different.

However, as we will discuss further in Section 4.6, these two metrics pick different inpainting configurations as the best ones.

4.5 Human evaluation

4.5.1 Methods for Human evaluation

The answer to RQ.3.2 is obtained by asking a human jury to evaluate different images that have been inpainted with different methods. This evaluation has as main method an online form format, where each participant can answer the questions independently, using their own device, with no supervision. This form can be found in the appendix. The main focus of this form is to obtain a single numerical score for each kind of inpainting, based on overall quality (how good it looks and how obvious it is that the inpainting has happened) and information leak (whether it is possible to guess the original text, despite trying to protect it), combined together. For this reason, questions present an image and ask the person to evaluate the quality of the inpainting with a numerical score from 1 to 5, for the overall quality combined with the information leak. A table showing how these questions were designed can be found as well in the appendix, in Table 12. Similarly to the previous section about automated metrics, we will be picking the best configuration by grouping them in terms of inpainting algorithm, mask type and dilation, so we average different kinds and databases.

As explained before, in order to perform the inpainting of images (for a later possible document synthesis) we can choose a configuration among 18, based on 3 parameters. Apart from these 3 configurable parameters, we detected 3 factors that can impact the evaluation of the image: insertion of text (whether only the inpainting happened or also text was inserted, to synthesize a fake document), dataset of the image (MIDV, PRADO or real image of specimen) and kind of document (ID card or passport). Thus, we could define 216 different combinations by considering all the different values that these parameters can have.

We would like to have an image for each of the 216 configurations. However, it is not possible to get a big population to evaluate such a big set. Thus, before getting the final mean scores, we need to run some statistical tests to find redundancy and reduce the size of the factorial experiment design. We did a pre-screening where 3 people familiar with identity documents evaluated the 216 images in terms of the two separate scores: overall quality of the image and data leak (from the original identity, due to a bad inpainting). With the scores of this experiment, we performed an Analysis of Variance (ANOVA) method in 8 that creates a model that infers all means and variances of the parameters, to see the effects that each of

them has individually and in combination with others. In general, this test tells whether different groups have similar means or not [47]. Thus, we used its results to find non-significant parameters with equal mean for their different groups. We dropped the parameters that were not significant, reducing the number of combinations to try and reducing the overall size of the form that the general jury filled out.

4.5.2 Results of Human evaluation (RQ3.2)

We analyzed the ANOVA from the pre-screening with the full set of 216 images, considering significant those parameter interactions with a P-value smaller than 5% and F-test statistic greater than 1. The result of the different parameter interactions' P-values and F-test statistics for the overall and leak score, given the evaluations of the 3 experts for the 216 images, can be found in Table 8.

Table 8: ANOVA

interaction	F leak	p-value leak	F overall	p-value overall
dilation	93.487	$1.653 \cdot 10^{-20}$	87.472	$2.237 \cdot 10^{-19}$
mask	35.720	$2.676 \cdot 10^{-15}$	25.449	$2.731 \cdot 10^{-11}$
database	31.638	$1.010 \cdot 10^{-13}$	23.124	$2.307 \cdot 10^{-10}$
mask:dilation	30.681	$2.384 \cdot 10^{-13}$	9.527	$1.893 \cdot 10^{-07}$
text	47.878	$1.284 \cdot 10^{-11}$	12.359	$5.640 \cdot 10^{-06}$
dilation:database	22.063	$6.142 \cdot 10^{-10}$	10.570	$3.140 \cdot 10^{-05}$
mask:database	11.770	$3.566 \cdot 10^{-09}$	10.443	$3.547 \cdot 10^{-05}$
kind	30.744	$4.606 \cdot 10^{-08}$	6.188	$7.126 \cdot 10^{-05}$
mask:kind	17.200	$5.724 \cdot 10^{-08}$	4.948	0.007
kind:database	14.907	$4.988 \cdot 10^{-07}$	3.243	0.012
database:text	13.017	$3.009 \cdot 10^{-06}$	6.108	0.014
mask:dilation:database	6.938	$1.885 \cdot 10^{-05}$	3.243	0.040
mask:text	9.552	$8.375 \cdot 10^{-05}$	3.572	0.059
dilation:text	15.171	0.000	1.735	0.141
dilation:kind	13.105	0.000	1.825	0.162
dilation:kind:database	7.121	0.001	1.712	0.191
mask:kind:database	3.744	0.005	1.457	0.214
dilation:database:text	4.727	0.009	1.510	0.222
mask:database:text	2.963	0.019	1.496	0.225
mask:dilation:text	2.396	0.092	1.275	0.280
inp_algorithm:dilation	1.993	0.137	0.848	0.358
inp_algorithm:database	1.621	0.167	0.946	0.389
mask:kind:text	1.161	0.314	0.937	0.392
kind:text	0.758	0.384	0.768	0.464
inp_algorithm:text	0.922	0.398	0.626	0.644
mask:dilation:kind	0.695	0.499	0.622	0.647
inp_algorithm:database:text	0.739	0.566	0.615	0.652
inp_algorithm:mask:database	0.760	0.639	0.573	0.682
inp_algorithm:mask:kind	0.582	0.676	0.115	0.735
inp_algorithm:kind:text	0.367	0.693	0.242	0.785
inp_algorithm:dilation:text	0.330	0.719	0.228	0.796
kind:database:text	0.317	0.728	0.059	0.809
dilation:kind:text	0.103	0.749	0.143	0.867
inp_algorithm:mask	0.468	0.759	0.267	0.899
inp_algorithm	0.267	0.766	0.434	0.901
inp_algorithm:mask:dilation	0.267	0.899	0.210	0.933
inp_algorithm:kind	0.065	0.937	0.049	0.952
inp_algorithm:mask:text	0.147	0.964	0.045	0.956
inp_algorithm:dilation:database	0.134	0.970	0.016	0.984
inp_algorithm:dilation:kind	0.027	0.973	0.007	0.993
inp_algorithm:kind:database	0.084	0.987	0.059	0.994
Residual				

From this table we can see that, especially for the leak score, the *dilation* parameter is very significant, with a p-value below 0.01. This dilation has a big impact on concealing the original text (or failing to do so), thus its different values hold different means. In a similar way, the *text* parameter is also, in a weaker way with also $p < 0.01$, significant. As we could expect, some text can cover the original text, improving a bit the leak score, but not completely, as it doesn't cover the whole original text.

Focusing on the overall quality score, we see that *inpainting algorithm*, with p-value 0.766 is not a significant factor for image quality determination, meaning that different inpainting algorithms have similar means, they perform similarly. Further, whether text is inserted is very significant as $p < 0.01$. This clashes with the goal of the experiment: to evaluate the performance of the inpainting and not the performance of the text insertion. Therefore, we decide to always use the inpainting algorithm *telea* and to always insert text, as it is the final task that we would need to assess (fake documents) and we don't really care about the individual performance of inpainting algorithms, but the final result in combination with the text insertion. Thus, we reduced the factorial experiment design by 6, from 216 to 36 different combinations.

Taking those impacts into account, we decided to prepare the final form with just 36 different images/configurations. We use the *telea* inpainting algorithm that had the greatest mean (still very similar to the others), and always text *inserted*, we will vary the mask type (*restricted*, *charmin* or *charbox*), the dilation (0 or 3), the kind of document (*id* or *passport*) and database (*MIDV*, *PRADO* or *specimens*). These 36 configurations are the ones that were also used in the automatic evaluation of images, in Section 4.4.2. The form was answered by 32 people.

Before picking the best configuration based on mean scores, we wanted to check which parameters affected the quality in this short form with 36 images. Thus, we performed a second ANOVA test using the answers from the short form. The ANOVA can be found in Table 9.

Table 9: ANOVA of the human evaluation of the 36 images form

interaction	sum of squares	degrees of freedom	F statistic	p-value
kind:database	184.335 1	2.0	74.861 0	$3.573 4 \cdot 10^{-31}$
database	101.829 7	2.0	41.354 4	$4.948 7 \cdot 10^{-18}$
mask:database	59.035 8	4.0	11.987 6	$1.554 6 \cdot 10^{-09}$
dilation:database	39.388 9	2.0	15.996 4	$1.424 9 \cdot 10^{-07}$
mask:kind:database	33.595 0	4.0	6.821 7	$2.013 4 \cdot 10^{-05}$
mask	24.200 7	2.0	9.828 2	$5.888 8 \cdot 10^{-05}$
kind	18.068 1	1.0	14.675 4	0.000 1
mask:dilation	20.953 4	2.0	8.509 5	0.000 2
mask:kind	19.727 6	2.0	8.011 6	0.000 4
mask:dilation:database	15.390 7	4.0	3.125 2	0.014 4
dilation:kind:database	6.087 8	2.0	2.472 3	0.084 9
dilation:kind	3.014 3	1.0	2.448 3	0.117 9
mask:dilation:kind	4.222 2	2.0	1.714 7	0.180 5
mask:dilation:kind:database	6.143 4	4.0	1.247 5	0.289 1
dilation	0.702 5	1.0	0.570 6	0.450 2

This table show that the interaction between kind and database (which basically defines the document shown in the image) has the most significant results. As we could expect, this tells us that the quality score people give is mostly based on the document image they are evaluating: for the different values of kind and database (different document images) means are different. Different scores are given to different document images, as they can have backgrounds of different complexities, making the inpainting process more or less obvious or smooth.

The mask and dilation interaction is also very significant. This means that some combinations of mask and dilation can give us an inpainting configuration that creates an image with an obvious inpainting that doesn't look good. However, individually, the dilation parameter is not a very significant variable. On its own, it doesn't really impact the score people give.

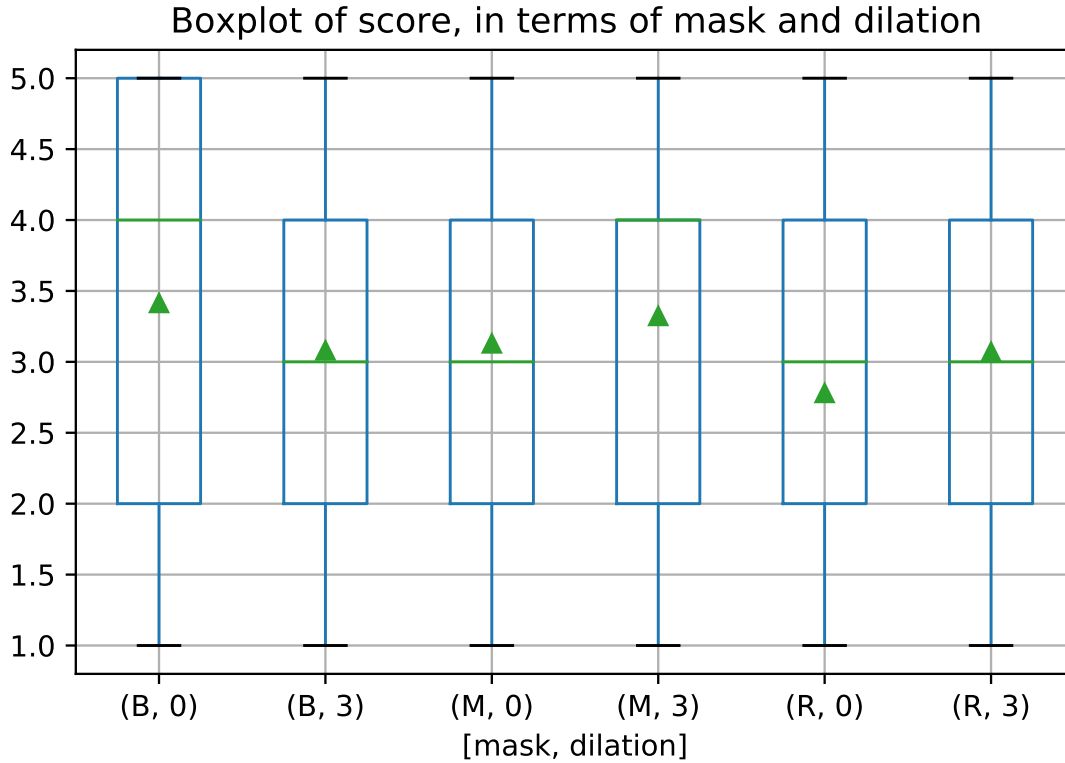


Figure 36: Boxplot of the quality score for different mask and dilation configurations

In Figure 36, we plot the distribution of the different scores for each of the 6 configurations we can get by changing mask type and type. This way we can find the best inpainting configuration, according to our human jury. As before, we fixed the inpainting algorithm to *telea* and we grouped by different mask types (*charmin* (M), *charbox*, (B) and *restricted* (R)) and dilations (0 and 3). We iterate over the different scores that people give, averaging for the different kinds and databases.

For each configuration we find a box describing the distribution of first and third quartile, a green line for the median and a triangle for the mean. Although most means have similar values, the configuration (B,0) (*charbox* and dilation 0) gets the higher mean (and also median). It is closely followed by (M,3) (*charmin* and dilation 3), with similar mean, same median but with a smaller box. Moreover, the median of (M,3) is right at the top end of the box, which means it is positively-skewed, with a higher frequency of high scores than (B,0) that has a line more in the middle of the box. (B,0) has a big box compared to the others, so its distribution of data is more uniform, the result might not be as reliable as the others,

as the trend for high scores is not as clear.

Finally, we can show an example of a fake id automatically generated using our tool and the configuration with highest mean according to the human jury: *telea*, *charbox* and dilation 0. This can be seen in Figure 39, on the left side we see the original document and on the right our synthesized one. We see that some aspects that could be improved is the size of some texts, especially the surname or the fields on the right side. The sharpness and intensity of text is also different from the original, it should be blended with the background in a smoother way. Also, the portrait background can be improved, we had to inpaint that area and now it appears blurred, not continuing the hologram that should be there, making it obvious that it is fake. The portrait picture is also too sharp, compared to the rest of the document. As mentioned before, should find a better way, more refined, to substitute the image.



Original Slovakian ID card. Source of ID image: PRADO [3]



Automatically generated Slovakian ID card. Source of ID image: PRADO [3]

Figure 39: Automatically generated document

4.6 Correlation between metrics (RQ3.3)

Finally, we will compare the correlation between the automated metrics and the manual metrics used for the evaluation of the inpaintings. First of all, we will get a general overview of the scores that each evaluation metric offers to each configuration in Table 10. We will remember that MS-SSIM is a similarity score in the range $[0,1]$. The human score is also a positive measure in the range $[1,5]$. On the other hand, NIQE is a distance score that ranges from 7 to 15, for our set. Contrary to the other measures, a high NIQE scores means worse

quality.

Table 10: Scores for each configuration and metric

configurations	msssim	niqe_rec	niqe_prot	human
telea_restricted_0	0.934	9.417	9.903	2.780
telea_restricted_3	0.931	9.429	9.839	3.070
telea_charbox_3	0.929	9.444	9.732	3.081
telea_charmin_0	0.934	9.432	9.916	3.129
telea_charmin_3	0.930	9.454	9.824	3.323
telea_charbox_0	0.930	9.398	9.756	3.414

We can see that *charbox* with dilation 0 is the best configuration according to the human score and the NIQE when recreating pictures (and for NIQE protecting it also takes a second place). Thus, we could say that the configuration with *telea*, *charbox* and dilation 0 is the best one. However, the MS-SSIM score disagrees and its means have similar values, so it is not clear that the configuration performs much better than the others.

Table 11: Correlations of each of the metrics with the others.

	msssim	niqe_rec	niqe_prot	human
msssim	1.000	-0.146	0.962	-0.636
niqe_rec	-0.146	1.000	0.016	-0.035
niqe_prot	0.962	0.016	1.000	-0.526
human	-0.636	-0.035	-0.526	1.000

Next, we take a look at the correlations in Table 11. Although we said that different methods agreed on a winner, taking a look at the correlations, we can see that ranking the same configuration as first is bit of a coincidence, as there is no big correlation between metrics. Despite having the same configuration as favorite, the order for the rest of configurations is different for each metric, there is no relation. We should highlight that any correlation of NIQE with others should be negative (as it is a distance and the others are positive scores), which is not the case. We can see no strong correlation as the only ones strong in magnitude (ms-ssim and niqe_prot) have the wrong sign. The only correlation worth-mentioning is that of human with niqe_prot, which is negative and has a magnitude of -0.5. This shows that

there is some correlation but is not strong.

The fact that there are no strong correlations, means that each metric bases its score in disjoint aspects. Thus, different scores don't correlate between each other. Moreover, this can also mean that there is not just one perfect configuration, the choice for the best configuration depends on what specific aspect we are looking at and since each metric is looking for a different thing (similarity, naturalness or overall good visual quality) the ranking of configurations varies. It is also worth noting that none of the configurations had a much better score than the others in any of the cases, so we can't really pick one configuration as best, we could just stick to one per default, assuming that other configurations won't have a much different impact.

4.7 Runtime of the tool

Although this tool is just a research project and it has not been optimized for production, we will comment on the running times of each of the modules. We either run the tools in our CPU Quad-Core Intel Core i5 with 1,4GHz or a GPU hosted in the cluster of the University of Twente, using a Nvidia GeForce RTX 2080 TI. First, we have the document cropping of an image resized to 512x512 pixels that in our CPU Segformer took around 60ms per image. From now on, all steps use an image of size 760x480 pixels. Afterward we have the text recognition, PaddleOCR took around 2.5s. Next step is document representation and entity extraction with PICK, that takes 2ms. In case of inpainting, our CPU took 200ms to erase all data fields in an image. Finally, the insertion of new text and substitution of the portrait takes 2.4s.

This means that just the template extraction, as it is, takes around 2.6s, being the text recognition part the slowest one. For now, even though it is a task that needs a very quick response (it needs to be processed in less than a second) it wouldn't be possible to perform it on-the-fly with a mobile device and we would need to do it offline, as it takes too long.

If done online, we would be creating templates that need to be stored and named. When we need to process a document, we will have to identify the document (find its name) and fetch the template of that kind of document, so it can be applied. On the other hand, the data protection and fake generation are tasks that don't need to be performed quickly, so we don't mind the long time of the fake generation part.

5 Conclusion

The main question for this project was how to automatically extract the layout of any MRTD document, in a country-independent way (RQ1). This split into two questions: how to evaluate that layout (RQ1.1) and how good is the performance of our extraction (RQ1.2). RQ1 directly led to another question, how can these templates be used in a real-life application? (RQ2). The answer is found by answering two other questions: how to censor private data in MRTDs (RQ2.1) and how to automatically generate fake data (RQ2.2). We followed up by evaluating its performance, so we asked how can we evaluate the previous methods (RQ3). In order to answer this, we answered three other questions: How well do these methods perform according to automated measures (RQ3.1), how well they perform according to humans (RQ3.2) and what is the correlation between these two approaches (RQ3.3).

To answer RQ1, we extracted templates from MRTDs using the tool PICK and it looks like it is a suitable tool to perform the template extraction of MRTDs in a country-independent fashion. It is able to learn how to classify each extracted field and this is justified by its precision. Around 93% of the times it correctly classified the extracted field from the document, providing us with a positive answer to RQ1.2 about the performance of the layout extraction.

We saw that a bottleneck for this task is the previous text recognition step, which, despite having improved a lot in the last years (the new method PaddleOCR overturns the traditional OCR), still has a long way to go to detect all the text in an image, correctly. To improve this detection, we found out that SegFormer is a good choice to perform image segmentation and crop the document inside an image, simplifying the image that the text recognition system processes.

The answer to RQ2 is that it is indeed possible to use the templates extracted from MRTDs to protect sensitive data contained in MRTDs (RQ2.1) and automatically create fake documents (RQ2.2). We protected the data using basic inpainting algorithms, like the Fast Marching by Telea. Although the sensitive data is protected perfectly using simple inpaintings algorithms, most of the times in a very smooth way, some documents with complex backgrounds might make this removal of data look not as pleasant. This last point also affects the fake document generation, a decent process that, despite being decent, can clearly be improved. We should improve the portrait detection and make easier the process of adapting the font type and size. For the rest, the quality of fake documents is good enough,

as the newly inserted text helps covering not so good-looking inpaintings.

RQ3 has as an answer that it is possible to evaluate these applications both automatically and manually. Unfortunately, their results differ a bit, giving a negative answer to RQ3.3 about their correlation. Even though they don't correlate, we think that this result can be expected and doesn't mean a bad performance of automated metrics answering RQ3.1 nor manual ones answering RQ3.2. These metrics base their evaluation on different aspects of images and that creates different scores. We should also keep in mind that despite not correlating, all means have very close values, no big differences are shown. Again, this proves that there is no single good configuration for inpainting text in MRTDs, all configurations have similar performance and depends a lot on the specific document being used.

To sum up, with this project we proved that it is possible to extract the template of an MRTD document in a country-independent fashion (RQ1) and that it can be later used in some real-life applications (RQ2) like data protection and fake document generation, with decent quality. However, the evaluation of these applications (RQ3) is subjective and will vary depending on what aspect we are looking at, as most configurations have similar scores and none performs clearly better.

As it will be detailed in the next Chapter 6, this project can easily be improved by getting a bigger and more complete dataset that allows a better training of the models for text recognition and entity extraction. We should also improve the fake document generation by getting a more robust and accurate way to detect the portrait pictures within the image and font adaptation. Finally, as a conclusion for InnoValor, we can say that this projects shows that it is possible to automatically extract and classify data fields from MRTDs, but this accuracy is still not high enough to be put into production, although that could be possible if we created a confidence score that tells whether we can rely on a result or not.

6 Future Work

6.1 Data preprocessing

A first line of work can be improving the text recognition systems. This could be easily done by adapting a system to the domain of MRTDs, given a completely annotated dataset of MRTDs (as now we only transcribed the fields in the VIZ and MRZ, but not other irrelevant text in the document). We could fine-train our text recognition model for MRTDs, so it performs the recognition better, having a positive impact in the layout extraction (RQ1). Also, we could work on the document cropping part. Although we obtained good results, there is still place for improvement. We could try other models for image segmentation (like a Dynamic-Structured Semantic Propagation Network [48] or DeepLabv3 [49]) or refine our algorithm for corner detection and perspective warp, making it more robust for line detection and fine-tuning the corner estimation.

6.2 Document representation

We saw that PICK as the method for document representation can be improved with a postprocessing step that verifies PICK's decision, for example in the case of dates that were wrongly classified. Their labels could be fixed by just sorting the different dates in time: birth date is the oldest one, then issue date comes, and finally expiry date. All these tasks could benefit if more data became publicly available. A publicly available big MRTD dataset would be very useful for research similar to ours. For example, a dataset like MIDV-2020, containing fake identities which look as good as real ones, could be good enough. Although it would be better if it were bigger in size (with more than just 10 different identity documents). In a nutshell, a next step that could help any work on MRTDs, will be the creation of a big publicly available dataset, with different documents and identities for each of them (in the magnitude of 100 documents and 100 identities), that contains full transcription of the text contained and also gives each text field their label. A possible way to obtain it would be following a similar method to MIDV using a bigger team that can generate more identities for more countries.

6.3 Applications of extracted templates

When it comes to RQ2.1, we saw that the main part of inpaintings is the mask selection. There are many possibilities and although the restricted version seems to have the most potential, it can be still improved, for example, fine-tuning the bounding box of the text, so no other elements are contained. The fake synthesis has also a big margin of improvement. We don't get perfect results in font recognition, so a method that doesn't depend on external APIs like Deepfont [50] can be a good idea (RQ 2.2). Another aspect to be improved is the color/contrast of the added text, which is very clean and has intense colors, while the background or other text in the document might be more faded or blurred. We could filter these new added fields, to make them look more similar to the original ones, by means of a blurring filter like a Gaussian one or other state-of-the-art methods. Finally, there is the face detection and removal. Our algorithm for face detection works decently for most documents, but sometimes it fails to cover the whole portrait completely. We should find a way of refining this, maybe a different approach based on edge detection or image binarization and contouring once we have an approximate idea of the region where the portrait is located.

Finally, there is a general remark on this project. It fulfills the requisite of extracting a template in a country-independent fashion. However, it would be nice if we could improve these templates iteratively, if we could update the template for a specific kind of document when the same document is processed again (with a different identity). It would be nice to get as accurate templates as possible but improving the stored template every time we process a new instance of the same document. For example, if we process for the first time a Dutch passport from the 2020 version, we would create a template and store it. If later in time we saw the same kind of document again (Dutch passport from 2020) but with a different identity, we could use the results of its extracted template to update the one we stored, by finding the right way to merge them. In order to update a template, we would need a way to detect the exact kind of document and another way to name each version of a document so they can be uniquely identified.

References

- [1] ICAO. (2021, Jul) Doc series - doc 9303, machine readable travel documents. [Online]. Available: <https://www.icao.int/publications/pages/publication.aspx?docnum=9303>
- [2] N. Tavakolian, A. Nazemi, and D. Fitzpatrick, “Real-time information retrieval from Identity cards,” *arXiv:2003.12103 [cs]*, Mar. 2020, arXiv: 2003.12103. [Online]. Available: <http://arxiv.org/abs/2003.12103>
- [3] C. of the European Union. Council of the european union - prado. [Online]. Available: <https://www.consilium.europa.eu/prado/en/prado-start-page.html>
- [4] A. Castelblanco, J. Solano, C. Lopez, E. Rivera, L. Tengana, and M. Ochoa, “Machine learning techniques for identity document verification in uncontrolled environments: A case study,” *Pattern Recognition*, vol. 12088, pp. 271 – 281, 2020.
- [5] V. V. Arlazarov, K. Bulatov, T. Chernov, and V. L. Arlazarov, “MIDV-500: A Dataset for Identity Documents Analysis and Recognition on Mobile Devices in Video Stream,” *Computer Optics*, vol. 43, no. 5, pp. 818–824, Oct. 2019, arXiv: 1807.05786. [Online]. Available: <http://arxiv.org/abs/1807.05786>
- [6] I. Muslea, “Extraction radoattners for information extraction tasks: A survey,” *In AAAI-99 Workshop on Machine Learning for Information Extraction*, pp. 1–6, 1999.
- [7] E. Riloff and W. Phillips, “An introduction to the sundance and autoslog systems,” *University of Utah School of Computing Technical Report #UUCS-04-015*, 10 2011.
- [8] S. Soderland, “Learning information extraction rules for semi-structured and free text,” *Mach. Learn.*, vol. 34, no. 1–3, p. 233–272, feb 1999. [Online]. Available: <https://doi.org/10.1023/A:1007562322031>
- [9] N. Kushmerick, D. S. Weld, and R. B. Doorenbos, “Wrapper induction for information extraction,” *IJCAI*, 1997.
- [10] T. Gogar, O. Hubacek, and J. Sedivy, “Deep neural networks for web page information extraction,” *Artificial Intelligence Applications and Innovations*, pp. 154–163, 2016.
- [11] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, “Neural architectures for named entity recognition,” *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics:*

- Human Language Technologies*, pp. 260–270, Jun. 2016. [Online]. Available: <https://aclanthology.org/N16-1030>
- [12] X. Liu, F. Gao, Q. Zhang, and H. Zhao, “Graph convolution for multimodal information extraction from visually rich documents,” *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*, pp. 32–39, Jun. 2019. [Online]. Available: <https://aclanthology.org/N19-2005>
- [13] R. Smith, “An overview of the tesseract ocr engine,” *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 2, pp. 629 – 633, 10 2007.
- [14] R. Panchendrarajan and A. Amaesan, “Bidirectional LSTM-CRF for named entity recognition,” *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, 1–3 Dec. 2018. [Online]. Available: <https://aclanthology.org/Y18-1061>
- [15] W. Yu, N. Lu, X. Qi, P. Gong, and R. Xiao, “Pick: Processing key information extraction from documents using improved graph learning-convolutional networks,” *25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event / Milan, Italy, January 10-15, 2021*, pp. 4363–4370, 2020. [Online]. Available: <https://doi.org/10.1109/ICPR48806.2021.9412927>
- [16] A. R. Katti, C. Reisswig, C. Guder, S. Brarda, S. Bickel, J. Höhne, and J. B. Faddoul, “Chargrid: Towards understanding 2d documents,” *EMNLP*, 2018.
- [17] T. I. Denk and C. Reisswig, “Bertgrid: Contextualized embedding for 2d document representation and understanding,” *Document Intelligence” workshop of 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada*, vol. abs/1909.04948, 2019. [Online]. Available: <http://arxiv.org/abs/1909.04948>
- [18] M. Kerroumi, O. Sayem, and A. Shabou, “VisualWordGrid: Information Extraction From Scanned Documents Using A Multimodal Approach,” *Document Analysis and Recognition - ICDAR 2021 Workshops*, Sep. 2021.
- [19] J. Wang, C. Liu, L. Jin, G. Tang, J. Zhang, S. Zhang, Q. Wang, Y. Wu, and M. Cai, “Towards robust visual information extraction in real world: New dataset and novel solution,” *CoRR*, vol. abs/2102.06732, 2021. [Online]. Available: <https://arxiv.org/abs/2102.06732>

- [20] H. Guo, X. Qin, J. Liu, J. Han, J. Liu, and E. Ding, “Eaten: Entity-aware attention for single shot visual text extraction,” *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 254–259, 2019.
- [21] T. Mahalakshmi, R. Muthaiah, and P. Swaminathan, “An overview of template matching technique in image processing,” *Research Journal of Applied Sciences, Engineering and Technology*, vol. 4, pp. 5469–5473, 01 2012.
- [22] Y.-B. Kwon and J.-H. Kim, “Verification of the Document Components from Dual Extraction of MRTD Information,” *Graphics Recognition. Recent Advances and New Opportunities*, pp. 235–244, 2008.
- [23] A. Hartl, C. Arth, and D. Schmalstieg, “Real-time Detection and Recognition of Machine-Readable Zones with Mobile Devices,” *VISAPP 2015 - 10th International Conference on Computer Vision Theory and Applications; VISIGRAPP, Proceedings*, vol. 3, pp. 79–87, Jan. 2015.
- [24] K. Bulatov, D. Matalov, and V. V. Arlazarov, “MIDV-2019: Challenges of the modern mobile-based document OCR,” *Twelfth International Conference on Machine Vision (ICMV 2019)*, p. 64, Jan. 2020, arXiv: 1910.04009. [Online]. Available: <http://arxiv.org/abs/1910.04009>
- [25] X. Fang, X. Fu, and X. Xu, “Id card identification system based on image recognition,” *2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 1488–1492, Jun. 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/8283074/>
- [26] D. Ballard, “Generalizing the hough transform to detect arbitrary shapes,” *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0031320381900091>
- [27] K. Bulatov, V. V. Arlazarov, T. Chernov, O. Slavin, and D. Nikolaev, “Smart IDReader: Document Recognition in Video Stream,” *2017 14th IAPR International Conference on Document Analysis and Recognition ICDAR*, pp. 39–44, Nov. 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/8270294/>
- [28] H. Li, M. A. Lavin, and R. J. Le Master, “Fast hough transform: A hierarchical approach,” *Computer Vision, Graphics, and Image Processing*, vol. 36, no. 2, pp. 139–161, 1986. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0734189X86900733>

- [29] Y. Liu, H. James, O. Gupta, and D. Raviv, “Mrz code extraction from visa and passport documents using convolutional neural networks,” *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 25, no. 1, pp. 29–39, Mar 2022. [Online]. Available: <https://doi.org/10.1007/s10032-021-00384-2>
- [30] K. B. Bulatov, E. Emelianova, D. V. Tropin, N. Skoryukina, Y. S. Chernyshova, A. Sheshkus, S. A. Usilin, Z. Ming, J. Burie, M. M. Luqman, and V. V. Arlazarov, “MIDV-2020: A comprehensive benchmark dataset for identity document analysis,” *CoRR*, vol. abs/2107.00396, 2021. [Online]. Available: <https://arxiv.org/abs/2107.00396>
- [31] Y. Du, C. Li, R. Guo, C. Cui, W. Liu, J. Zhou, B. Lu, Y. Yang, Q. Liu, X. Hu, D. Yu, and Y. Ma, “Pp-ocrv2: Bag of tricks for ultra lightweight OCR system,” *CoRR*, vol. abs/2109.03144, 2021. [Online]. Available: <https://arxiv.org/abs/2109.03144>
- [32] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [33] K. A. Mat Said, A. Jambek, and N. Sulaiman, “A study of image processing using morphological opening and closing processes,” *International Journal of Control Theory and Applications*, vol. 9, pp. 15–21, 01 2016.
- [34] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.
- [35] R. Smith, Z. Podobny *et al.*, “Tesseract ocr,” <https://github.com/tesseract-ocr/tesseract>, 2021.
- [36] PaddlePaddle, “Paddleocr,” <https://github.com/PaddlePaddle/PaddleOCR>, 2022.
- [37] A. Sheshkus, D. Nikolaev, and V. L. Arlazarov, “Houghencoder: Neural network architecture for document image semantic segmentation,” *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 1946–1950, 2020.
- [38] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, “Segformer: Simple and efficient design for semantic segmentation with transformers,” *CoRR*, 2021. [Online]. Available: <https://arxiv.org/abs/2105.15203>
- [39] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: <http://arxiv.org/abs/1505.04597>

- [40] C. Guillemot and O. Le Meur, “Image inpainting : Overview and recent advances,” *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 127–144, 2014.
- [41] A. Telea, “An image inpainting technique based on the fast marching method,” *Journal of Graphics Tools*, vol. 9, no. 1, pp. 23–34, 2004.
- [42] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, “Image inpainting,” *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, p. 417–424, 2000. [Online]. Available: <https://doi.org/10.1145/344779.344972>
- [43] J. Seiler, M. Jonscher, M. Schöberl, and A. Kaup, “Resampling images to a regular grid from a non-regular subset of pixel positions using frequency selective reconstruction,” *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4540–4555, 2015.
- [44] A. Mittal, R. Soundararajan, and A. C. Bovik, “Making a “completely blind” image quality analyzer,” *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 209–212, 2013.
- [45] W. F. I. C. SRL. Api identify fonts from image. [Online]. Available: <https://www.whatfontis.com/API-identify-fonts-from-image.html>
- [46] D. M. Rouse and S. S. Hemami, “Understanding and simplifying the structural similarity metric,” *2008 15th IEEE International Conference on Image Processing*, pp. 1188–1191, 2008.
- [47] A. Gelman, “Analysis of variance. why it is more important than ever,” *The Annals of Statistics*, vol. 33, no. 1, feb 2005.
- [48] X. Liang, H. Zhou, and E. P. Xing, “Dynamic-structured semantic propagation network,” *CoRR*, vol. abs/1803.06067, 2018. [Online]. Available: <http://arxiv.org/abs/1803.06067>
- [49] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” *CoRR*, vol. abs/1802.02611, 2018. [Online]. Available: <http://arxiv.org/abs/1802.02611>
- [50] Z. Wang, J. Yang, H. Jin, E. Shechtman, A. Agarwala, J. Brandt, and T. S. Huang, “Deepfont: Identify your font from an image,” *Proceedings of the 23rd ACM International Conference on Multimedia*, p. 451–459, 2015. [Online]. Available: <https://doi.org/10.1145/2733373.2806219>

A Appendix

A.1 Template example

```
{
'doc_number': [[0.7737, 0.1437, 0.8882, 0.1437, 0.8882, 0.1792, 0.7737, 0.1792]],
'id_number': [[0.6263, 0.4292, 0.7974, 0.4292, 0.7974, 0.4708, 0.625, 0.4708]],
'other': [[0.1447, 0.0333, 0.2461, 0.0333, 0.2461, 0.0625, 0.1447, 0.0625], [0.7737,
0.0875, 0.8592, 0.0875, 0.8592, 0.1083, 0.7737, 0.1083], [0.4118, 0.1062, 0.6855,
0.1062, 0.6855, 0.1271, 0.4118, 0.1271], [0.7724, 0.1083, 0.9039, 0.1104, 0.9039,
0.1313, 0.7724, 0.1292], [0.2842, 0.1354, 0.3197, 0.1354, 0.3197, 0.1688, 0.2842,
0.1688], [0.1408, 0.1583, 0.2118, 0.1583, 0.2118, 0.1875, 0.1408, 0.1875], [0.6211,
0.3208, 0.75, 0.3208, 0.75, 0.3479, 0.6211, 0.3479], [0.6263, 0.4729, 0.7105,
0.4729, 0.7105, 0.4958, 0.6263, 0.4958], [0.6303, 0.5021, 0.9026, 0.5083, 0.9026,
0.5437, 0.6303, 0.5375], [0.6316, 0.5771, 0.8895, 0.5854, 0.8895, 0.6208, 0.6316,
0.6125], [0.6316, 0.6708, 0.9145, 0.6708, 0.9145, 0.7208, 0.6316, 0.7208], [0.1434,
0.0625, 0.2276, 0.0625, 0.2276, 0.0917, 0.1434, 0.0917], [0.2776, 0.1062, 0.3553,
0.1062, 0.3553, 0.1292, 0.2776, 0.1292], [0.4145, 0.1354, 0.4671, 0.1354, 0.4671,
0.1729, 0.4145, 0.1729], [0.1395, 0.1875, 0.2105, 0.1833, 0.2118, 0.2167, 0.1395,
0.2208], [0.6276, 0.35, 0.7211, 0.3542, 0.7211, 0.3917, 0.6276, 0.3875], [0.6263,
0.55, 0.7211, 0.55, 0.7211, 0.5708, 0.6263, 0.5708], [0.625, 0.6229, 0.8013, 0.625,
0.8013, 0.6542, 0.625, 0.6521], [0.1408, 0.0896, 0.2368, 0.0938, 0.2368, 0.1229,
0.1408, 0.1187], [0.1408, 0.1229, 0.2211, 0.1271, 0.2211, 0.1625, 0.1408, 0.158]]3,
'label_surname': [[0.2737, 0.1729, 0.4395, 0.1729, 0.4395, 0.2, 0.2737, 0.2]],
'surname': [[0.2816, 0.2104, 0.4829, 0.2104, 0.4829, 0.2396, 0.2816, 0.2396]],
'label_name': [[0.2737, 0.2458, 0.5526, 0.2479, 0.5526, 0.275, 0.2737, 0.2729]],
'name': [[0.2803, 0.2812, 0.5132, 0.2812, 0.5132, 0.3104, 0.2803, 0.3104]],
'label_doc_number': [[0.2763, 0.3208, 0.425, 0.3208, 0.425, 0.3417, 0.2763, 0.3417],
[0.6237, 0.3958, 0.7776, 0.3958, 0.7776, 0.4229, 0.6237, 0.4229]],
'label_nationality': [[0.4197, 0.325, 0.5053, 0.3208, 0.5053, 0.3438, 0.4197,
0.3479,
0.3479,
'nationality': [[0.2816, 0.3438, 0.4592, 0.3438, 0.4592, 0.3875, 0.2816, 0.3875]],
'label_birth_date': [[0.2724, 0.3917, 0.5092, 0.3958, 0.5092, 0.4229, 0.2724,
0.418]]8,
'birth_date': [[0.2816, 0.4292, 0.425, 0.4292, 0.425, 0.4646, 0.2816, 0.4646]],
```



	Passaporte Passport Passeport Especial Special Spécial	República Portuguesa / Portuguese Republic / République Portugaise	
		Tipo/Type/Type PS PRT	Código do País/Code of issuing State/Code de l'État émetteur V111659
		[01] Apetido(s)/Surname/Nom NUNES MENDES	
		[02] Nome(s) próprio(s)/Given name(s)/Prénom(na) TIAGO FERNANDO	
		[03] Nacionalidade/Nationality/Nationalité PORTUGUESA	
		[05] Data de nascimento/Date of birth/Data de naissance 25.03.1987	
		[07] Sexo/Sex/Sexe M	
		[09] Data de emissão/Date of issue/Data de délivrance 08.06.2009	
		[11] Válido até/Date of expiry/Data d'expiration 08.06.2013	
		[04] Altura/Height/Taille 1.74 m	
		[06] Número de identificação pessoal/Personal identifying number Identifiant personnel BI13438070	
		[08] Local de nascimento/Place of birth/Lieu de naissance C SE NOVA*COIMBRA	
		[10] Autoridade/Authority/Autorité SECR. GERAL DO MAI	
		[12] Assinatura do titular/Holder's signature/Signature du titulaire Tiago Fernando Nunes Mendes	
		P<PRTNUNES<MENDES<<TIAGO<FERNANDO<<<<<<<<<<<<<	
		V111659<<4PRT8703259M1306086BI13438070<<<<16	

Figure 40: Portuguese Passport and its template

A.2 Human Evaluation Form

Table 12: Distribution of questions for the form

Question	INP_ALG	MASK	DILATION	KIND	DATABASE	ALB_id	EST_pas	LVA_id	PRT_pass	svk_id	srb_pass
1	T	B	0	I	P	x					
2	T	B	3	I	P	x					
3	T	M	0	I	P	x					
4	T	M	3	I	P	x					
5	T	R	0	I	P	x					
6	T	R	3	I	P	x					
7	T	B	0	P	P		x				
8	T	B	3	P	P		x				
9	T	M	0	P	P		x				
10	T	M	3	P	P		x				
11	T	R	0	P	P		x				
12	T	R	3	P	P		x				
13	T	B	0	I	S			x			
14	T	B	3	I	S			x			
15	T	M	0	I	S			x			
16	T	M	3	I	S			x			
17	T	R	0	I	S			x			
18	T	R	3	I	S			x			
19	T	B	0	P	S				x		
20	T	B	3	P	S				x		
21	T	M	0	P	S				x		
22	T	M	3	P	S				x		
23	T	R	0	P	S				x		
24	T	R	3	P	S				x		
25	T	B	0	I	M					x	
26	T	B	3	I	M					x	
27	T	M	0	I	M					x	
28	T	M	3	I	M					x	
29	T	R	0	I	M					x	
30	T	R	3	I	M					x	
31	T	B	0	P	M						x
32	T	B	3	P	M						x
33	T	M	0	P	M						x
34	T	M	3	P	M						x
35	T	R	0	P	M						x
36	T	R	3	P	M						x

Table 13: Abbreviations used for the parameters

INP_ALG	F (FSR)	T (Telea)	N (NS)
MASK	M (Minimal)	B (box)	R (restricted)
DILATION	0	2	
TEXT	0 (just erased)	1 (inserted)	
KIND	I	P	
Database	Prado	Midv	Specimen

Image Quality Evaluation

Welcome to this questionnaire! This questionnaire, made by the University of Twente student Eric Santiago Garcia (contact mail: e.santiagogarcia@student.utwente.nl), is part of his master thesis: *Extraction of layout from MRTDs in a country-independent fashion*. In this project we extract and analyze all the fields contained in Machine-Readable Travel Documents (MRTD) so we can create a layout, to know the location of each field in that specific kind of MRTD.

In the project we automatically generate fake identities and there are different methods to erase the text so it can be later substituted. Thus, the goal of this questionnaire is to evaluate the different methods so we can find the best one.

* Required

1. You are invited to take part of a questionnaire, in which you will be asked to answer some questions about a number images. You will be shown some images of ID documents, these might be either “real images” (corresponding to fake identities, created manually with almost-perfect results, so we consider them real) or “fake images” (automatically generated using the main tool of this project). You will be asked to grade them quantitatively. This will take you around 10 minutes. There are no risks or benefits that you can reasonably expect from this. No personal information about you will be collected. * 0 pc

If you decide to participate in the questionnaire, please understand your participation is voluntary and you may withdraw from it at any time without giving any reasons. If you have any further questions, you can contact the researcher via an e-mail to e.santiagogarcia@student.utwente.nl.

If you are not satisfied with how this study is being conducted, or if you have any concerns, complaints, or general questions about the research or your rights as a participant, please contact the Ethics Committee, Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS), University of Twente, at +31534896719, or email ethicscommittee-cis@utwente.nl.

Please click YES below if you are at least 18 years old and agree to take part in the questionnaire.

I hereby declare that I have been clearly informed about the nature and method of the research and I agree to participate.

I give consent for my answers to be used in presentations and reports. If they are used, they will be completely anonymous.

Finally, I declare that I am at least 18 years old

Mark only one oval.

☐ Yes

☐ No

Practice questions

Now follows a series of practice questions intended as training for the main part of the experiment, their answers won't be taken into account. Given an image, you will be given some hints about the expected answer, and we will validate your answer, making sure you have seen the range of possible images in terms of quality.

We will evaluate images where the original data fields have been erased automatically and have been substituted by new text. In order to delete text and substitute its pixels with something visually similar to its surrounding, we used different inpainting techniques. Inpainting is the process of filling parts in missing parts of an image so it becomes complete, in a concealed way. There are multiple options for this process, which we intend to evaluate with this questionnaire.

We are interested in those parts of the image where text has been removed (so we ignore the others that are kept). We are assessing the quality of the text deletion (circle E in the image below, we can't see anything weird on the background that's good, while in C and D there are some small artifacts). We won't be evaluating how good the new text is: we won't check whether it is aligned (A, the expiry date is not aligned with the issue one), we won't care about how different the new text looks from the original one (B, BI13438070 has a stronger black color and different font from C SE NOVA*COIMBRA).

We will be answering one question:

1. How good is the overall visual quality? You will pick a score from 1 to 5, we would like you to rate an image with a high quality score (close to 5) if you couldn't really tell that the image has been manipulated (text is removed,) and give a low score (close to 1) if you can tell something happened or it simply doesn't look very good. Also, a low score will be assigned if some original text was not concealed and it can still be guessed, even if it is just one character.

Aspects to take into account:

In areas where text has been erased and substituted, we will give a high score if:

- The background is consistent, the surrounding pattern is continued, like what happens in E
- There are no clear strokes of erased text: no strokes with a different color, unlike what happens in C, D or under the RTU of PORTUGUESA, where we see a darker stroke.
- We can't see any characters from the original text that weren't properly erased

 <p>Passaporte Passport Passeport</p> <p>Especial Special Spécial</p>	<p>República Portuguesa / Portuguese Republic / République Portugaise</p>	 <p>Passaporte n.^o Passport no. / Passeport n.^o</p> <p>V111659</p>
<p>Tipo/Type/Type Código do País/Code of issuing State/Code de l'État émetteur</p> <p>PS PRT</p>		
<p>[01] Apelido(s)/Surname/Nom C</p> <p>NUNES MENDES</p>		
<p>[02] Nome(s) próprio(s)/Given name(s)/Prénoms</p> <p>TIAGO FERNANDO</p>		
<p>[03] Nacionalidade/Nationality</p> <p>PORTUGUESA</p>		
<p>[05] Data de nascimento/Date of birth/Data de naissance</p> <p>25.03.1987</p>		
<p>[07] Sexo/Sex/Sexe</p> <p>M</p>		
<p>[09] Data de emissão/Date of issue/Data de délivrance</p> <p>08.06.2009</p>		
<p>[11] Válido até/Date of expiry/Data d'expiration</p> <p>08.06.2013 D</p>		
<p>[04] Altura/Height/Taille</p> <p>1.74 m</p>		
<p>[06] Número de identificação pessoal/Personal identifying number/Numéro personnel</p> <p>B113438070</p>		
<p>[08] Local de nascimento/Place of birth/Lieu de naissance</p> <p>C SE NOVA*COIMBRA</p>		
<p>[10] Autoridade emissora/Issuing authority/Autorité émissora</p> <p>SECR. GERAL DO MAI</p>		
<p>[12] Assinatura do titular/Holder's signature/Signature du titulaire</p> <p><i>Tiago Fernando Nunes Mendes</i></p>		

2. Our first example is that of an image which looks bad (in terms of text deletion) ★ 0 pc
although it is impossible to guess any of the original text
-Some parts don't have a consistent background, it looks blurred
-We can see obvious strokes of darker colors
+All original text is concealed, we can't see any remains.
Thus, we would give a low score.

How would you rate the quality of the image?



Mark only one oval.

- ☐ 1 (Low) Skip to question 4
☐ 2 Skip to question 4
☐ 3 Skip to question 3
☐ 4 Skip to question 3
☐ 5 (High) Skip to question 3

Skip to question 4

Practice
questions
I (TRY
AGAIN!)

Sorry! Your answer wasn't right, please answer again. Remember, that the quality will have a high score (close to 5) if the image looks natural, no weird strokes from different colors or other bad looking elements and a low score (close to 1) if it does not look natural or it's obvious that text was erased or old one can be guessed.

3. Take into account that:

*

0 pc

- Some parts don't have a consistent background, it looks blurred
 - We can see obvious strokes of darker colors
 - +All original text is concealed, we can't see any remains.
- Thus, we would give a low score.

How would you rate the quality of the image?



Mark only one oval.

- ☐ 1 (Low) Skip to question 4
- ☐ 2 Skip to question 4
- ☐ 3 Skip to question 3
- ☐ 4 Skip to question 3
- ☐ 5 (High) Skip to question 3

Skip to question 4

Practice questions II

- 4. Our next example is a near-perfect image.
 - +The patterns of the background have been mostly preserved, there are only small blurs
 - +There are no obvious strokes
 - +All original text is concealed, we can't see any remains.Thus, we would give a high score.

0 pc

How would you rate the quality of the image?



Mark only one oval.

- ☐ 1 (Low) Skip to question 5
- ☐ 2 Skip to question 5
- ☐ 3 Skip to question 5
- ☐ 4 Skip to question 6
- ☐ 5 (High) Skip to question 6

Practice
questions
II (TRY
AGAIN!)

Sorry! Your answers weren't right, please answer again. Remember, that quality will have a high score (close to 5) if the image looks natural, no weird strokes from different colors or other bad looking elements) and a low score (close to 1) if it does not look natural or it's obvious that text was erased.

5. How would you rate the quality of the image?

0 pc



Mark only one oval.

- ☐ 1 (Low) Skip to question 5
- ☐ 2 Skip to question 5
- ☐ 3 Skip to question 5
- ☐ 4 Skip to question 6
- ☐ 5 (High) Skip to question 6

Practice
questions III

Our next example is an image with a regular- low quality and we can guess some of the original text.

-There are obvious pink areas where text was erased, they don't conserve the original background.

-Not all original text was erased, we can still see some of the original characters like the last 4 in the date of issue.

Thus, we would give a low score.

6. How would you rate the quality of the image? *

0 pc



Mark only one oval.

- ☐ 1 (Low) Skip to question 8
- ☐ 2 Skip to question 8
- ☐ 3 Skip to question 7
- ☐ 4 Skip to question 7
- ☐ 5 (High) Skip to question 7

Practice
questions
III (TRY
AGAIN!)

Sorry! Your answers weren't right, please answer again. Remember, that the quality will have a high score (close to 5) if the image looks natural, no weird strokes from different colors or other bad looking elements) and a low score (close to 1) if it does not look natural or it's obvious that text was erased.

7. How would you rate the quality of the image? *

0 pc



Mark only one oval.

- ☐ 1 (Low) Skip to question 8
- ☐ 2 Skip to question 8
- ☐ 3 Skip to question 7
- ☐ 4 Skip to question 7
- ☐ 5 (High) Skip to question 7

Skip to question 8

Image
evaluation

Now you will be evaluating real images and your answers will be taken into account, no more training!

8. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

9. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

10. How would you rate the quality of the image? *



Mark only one oval.

1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

11. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

12. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

13. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

Pass Passport / Passeport		EESTI / ESTONIA / ESTONIE	
	Läht / Type Type	Riigi kood / Country code Code du pays	Dokumendi number / Document number Numéro de document
	P	EST	KS0000182
	1. Perekonnanimi / Surname / Nom	JOEORG	
	2. Eesnimi / Given name / Prénom	JAAK/KRISTJAN	
	3. Isikukood / Personal code / Identifiant personnel	38001085718	
	5. Sünniaeg / Date of birth / Date de naissance	08. 01. 1980	
	6. Sugu / Sex / Sexe	M/M	
	7. Sünnikoht / Place of birth / Lieu de naissance	EST	
	8. Väljaandud / Date of issue / Date de délivrance	21. 11. 2020	
	9. Kehtiv kuni / Date of expiry / Date d'expiration	21. 11. 2030	
	11. Valijaandja / Authority / Autorité	PPA/PBGB	
			4. Kodakondsus Citizenship / Nationalité
			EST
			
			10. Kasutaja allkiri / Holder's signature Signature du titulaire
			

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

Pass Passport / Passeport		EESTI / ESTONIA / ESTONIE	
	Liik / Type Type P	Riigi kood / Country code Code du pays EST	Dokumendi number / Document number Numéro de document KS0000182
	1. Perekonnanimi / Surname / Nom JOEORG	2. Eesnimi / Given name / Prénom JAAK/KRISTJAN	
	3. Isikukood / Personal code / Identifiant personnel 38001085718	4. Kodakondsus Citizenship / Nationalité EST	
	5. Sünniaeg / Date of birth / Date de naissance 08. 01. 1980	6. Sugu / Sex / Sexe M/M	7. Sünnikoht / Place of birth / Lieu de naissance EST
	8. Valla antud / Date of issue / Date de délivrance 21. 11. 2020	9. Kehtiv kuni / Date of expiry / Date d'expiration 21. 11. 2030	10. Kasutaja allkiri / Holder's signature Signature du titulaire 
	11. Valijaandja / Authority / Autorité PPA/PBGB		

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

Pass Passport / Passeport	EESTI / ESTONIA / ESTONIE	Dokumendi number / Document number Numéro de document
Lik / Type Type	Riigi kood / Country code Code du pays	
P	EST	KS0000182
1. Perekonnanimi / Surname / Nom	2. Eesnimi / Given name / Prénom	3. Isikukood / Personal code / Identifiant personnel
JOEORG	JAAK/KRISTJAN	38001085718
4. Sünniaeg / Date of birth / Date de naissance	5. Kodakondsus Citizenship / Nationalité	
08. 01. 1980	EST	
6. Soost / Sex / Sexe	7. Sünnikoht / Place of birth / Lieu de naissance	
M/M	EST	
8. Valla antud / Date of issue / Date de délivrance	9. Isikliku kehtivuse / Date of expiry / Date d'expiration	10. Kasutaja allkiri / Holder's signature Signature du titulaire
21. 11. 2020	21. 11. 2030	
11. Valitsus / Authority / Autorité		
PPA/PBGB		

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

Pass Passport / Passeport	EESTI / ESTONIA / ESTONIE	Dokumendi number / Document number Numéro de document
Lik / Type Type	Riigi kood / Country code Code du pays	
P	EST	KS0000182
1. Perekonnanimi / Surname / Nom	2. Eesnimi / Given name / Prénom	4. Kodakondsus Citizenship / Nationalité
JOEORG	JAAK/KRISTJAN	EST
3. Isikukood / Personal code / Identifiant personnel	5. Sünniaeg / Date of birth / Date de naissance	10. Kasutaja allkiri / Holder's signature Signature du titulaire
38001085718	08. 01. 1980	
6. Suguv / Sex / Sexe	7. Sünnikoht / Place of birth / Lieu de naissance	
M/M	EST	
8. Vabastamise kuupäev / Date of issue / Date de délivrance	9. Kehtivuse tähtaeg / Date of expiry / Date d'expiration	
21. 11. 2020	21. 11. 2030	
11. Väljaandja / Authority / Autorité		
PPA/PBGB		

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

18. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

Pass Passport / Passeport	EESTI / ESTONIA / ESTONIE	Dokumendi number / Document number Numéro de document
Lik / Type Type	Riigi kood / Country code Code du pays	
P	EST	KS0000182
1. Perekonnanimi / Surname / Nom	2. Eesnimi / Given name / Prénom	3. Isikukood / Personal code / Identifiant personnel
JOEORG	JAAK/KRISTJAN	38001085718
4. Sünniaeg / Date of birth / Date de naissance	5. Kodakondsus Citizenship / Nationalité	
08. 01. 1980	EST	
6. Suguv / Sex / Sexe	7. Sünnikoht / Place of birth / Lieu de naissance	
M/M	EST	
8. Vabastamise kuupäev / Date of issue / Date de délivrance	9. Kehtivuse kehtivusaeg / Date of expiry / Date d'expiration	10. Kasutaja allkiri / Holder's signature Signature du titulaire
21. 11. 2020	21. 11. 2030	
11. Väljaandja / Authority / Autorité		
PPA/PBGB		

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

20. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

21. How would you rate the quality of the image? *



Mark only one oval.

1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

22. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

23. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

24. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

25. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

26. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

27. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

	Passaporte	República Portuguesa / Portuguese Republic / République Portugaise		
	Passeport			
Especial		Tipo/Type/Type	Código do País/Code of issuing State/Code de l'État émetteur	Passaporte n. ^o Passport no. / Passeport n. ^o
Special		PS	PRT	V111659
		[01] Apelido(s)/Surname/Nom	NUNES MENDES	
		[02] Nome(s) próprio(s)/Given name(s)/Prénom(s)	TIAGO FERNANDO	
		[03] Nacionalidade/Nationality/Nationalité	PORTUGUESA	
		[05] Data de nascimento/Date of birth/Data de naissance	25.03.1987	
		[07] Sexo/Sex/Sexe	M	
		[09] Data de emissão/Date of issue/Data de délivrance	08.06.2009	
		[11] Válido até/Date of expiry/Data d'expiration	08.06.2013	
		[04] Altura/Height/Taille	1.74 m	
		[06] Número de identificação pessoal/Personal identifying number Identification personnel	B113438070	
		[08] Local de nascimento/Place of birth/Lieu de naissance	C SE NOVA*COIMBRA	
		[10] Autoridade/Authority/Autorité	SECR. GERAL DO MAI	
		[12] Assinatura do titular/Holder's signature/Signature du titulaire	Tiago Fernando Nunes Mendes	

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

 <p>Passaporte Passport Passeport Especial Special Spécial</p>	<p>República Portuguesa / Portuguese Republic / République Portugaise</p>	 Passaporte n. ^o Passport no. / Passeport n. ^o
	Tipo/Type/Type Código do País/Code of issuing State/Code de l'État émetteur PS PRT	V111659
	[01] Apêlido(s)/Surname/Nom NUNES MENDES	
	[02] Nome(s) próprio(s)/Given name(s)/Prénom(s) TIAGO FERNANDO	
	[03] Nacionalidade/Nationality/Nationalité PORTUGUESA	
	[05] Data de nascimento/Date of birth/Data de naissance 25.03.1987	[04] Altura/Height/Taille 1.74 m
	[07] Sexo/Sex/Sexe M	[06] Número de identificação pessoal/Personal identifying number Identifiant personnel BI13438070
[09] Data de emissão/Date of issue/Data de délivrance 08.06.2009	[08] Local de nascimento/Place of birth/Lieu de naissance C SE NOVA*COIMBRA	
[11] Válido até/Date of expiry/Data d'expiration 08.06.2013	[10] Autoridade/Authority/Autorité SECR. GERAL DO MAI	
	[12] Assinatura do titular/Holder's signature/Signature du titulaire <i>Tiago Fernando Nunes Mendes</i>	

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

30. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

31. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

32. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

33. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

34. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

35. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

36. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

37. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

38. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

39. How would you rate the quality of the image? *



Mark only one oval.

1 2 3 4 5

(Low) ☐ ☐ ☐ ☐ ☐ (High)

40. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

41. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

42. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

43. How would you rate the quality of the image? *



Mark only one oval.

	1	2	3	4	5	
(Low)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	(High)

The
end!

That's all. Thank you very much for your help. I hope you enjoyed filling out this questionnaire and don't hesitate to leave any comment or remarks in the box below. If you have any further questions or you would like to hear about the final results of this experiment, please send an email to e.santiagogarcia@student.utwente.nl. Have a nice day!

44. Please, write any comments or remarks.

This content is neither created nor endorsed by Google.

Google Forms

