Processing PDF Documents Written in Macedonian Language Using UiPath Document Understanding

Marija Krpachovska and Georgina Mirceva
Faculty of Computer Science and Engineering
Ss. Cyril and Methodius University in Skopje
Skopje, Macedonia

marija.krpachovska@students.finki.ukim.mk
georgina.mirceva@finki.ukim.mk

***Abstract*—Given the rapid advancement of technology over the past decades, the digitalization of repetitive processes is inevitable and for most parts of the world, it is already a reality. With the possibilities to make the execution of everyday tasks quicker, more precise, less expensive, and overall more efficient, it is not surprising that so many companies and institutions have already started transforming their processes to take full advantage of the benefits that current technology has to offer. Despite all this, when it comes to process automation in Macedonia there are still many areas in which there is room for further development. The purpose of this paper is to explore the possibilities, as well as the limitations that current technology has to offer in regard to processing PDF documents written in Macedonian language, more specifically focusing on Hipbath Document Understanding.**

Keywords—UiPath, RPA, PDF, automation, digitalization, document understanding, business process, workflow, package, JSON

I. Introduction

The automation of processes is the future of every industry. Delivering greater efficiency, cutting costs and offering better customer service are just some of the many perks that automating business processes has to offer. So, it is no surprise that most industries have already started taking advantage of what it has to offer. From the digitalization of approval requests, quotations and reposting, down to processes like employee onboarding and performance management, automation technologies are undeniably changing the workplace as we all know it.

Although Macedonia has already started adapting to this change, currently there are still many areas of unexplored potential. Contrary to popular belief, the automation of processes does not pose a threat to the availability of jobs, as it leads to the creation of new, often higher-paying job positions [1]. And in a time when Macedonia faces a serious deficit of workforce, automation might be the solution [2].

A commonly used technology in process automation is UiPath, a global software company that makes Robotic process automation software. As an RPA platform, UiPath can be used to create robots that emulate humans' actions while interacting with digital systems and software [3]. UiPath offers a variety of possibilities, from automating simple tasks like moving files from one folder to another, to more complex automation that includes interactions with web and desktop applications, processing excel or PDF files and working with email.

In this paper, our main area of focus is PDF processing with the help of UiPath Document Understanding, a framework that enables the processing of documents of different types and with different structures using one automated process. This solution can lead to lowering the error rate by up to 52%, reducing the costs that come with manual document processing by 35% and the time spent on document-related tasks by up to 17% [4]. It is not surprising that if implemented, this technology could transform the execution of document-related tasks in many companies and industries in Macedonia.

This raises the question: How advanced is UiPath, when it comes to processing documents written in Macedonian language, using the Cyrillic alphabet?

With the purpose of finding out the answer to that question, a workflow was constructed following UiPath Document Understanding's best practices. That workflow will be tested using 3 different types of documents: a digital PDF file written in Macedonian (Cyrillic), a scanned PDF file written in Macedonian (Cyrillic) and a handwritten PDF file written in Macedonian (Cyrillic). The results of that testing will be used to get a better understanding of the possibilities and limitations that UiPath currently has to offer when processing documents written in Macedonian language.

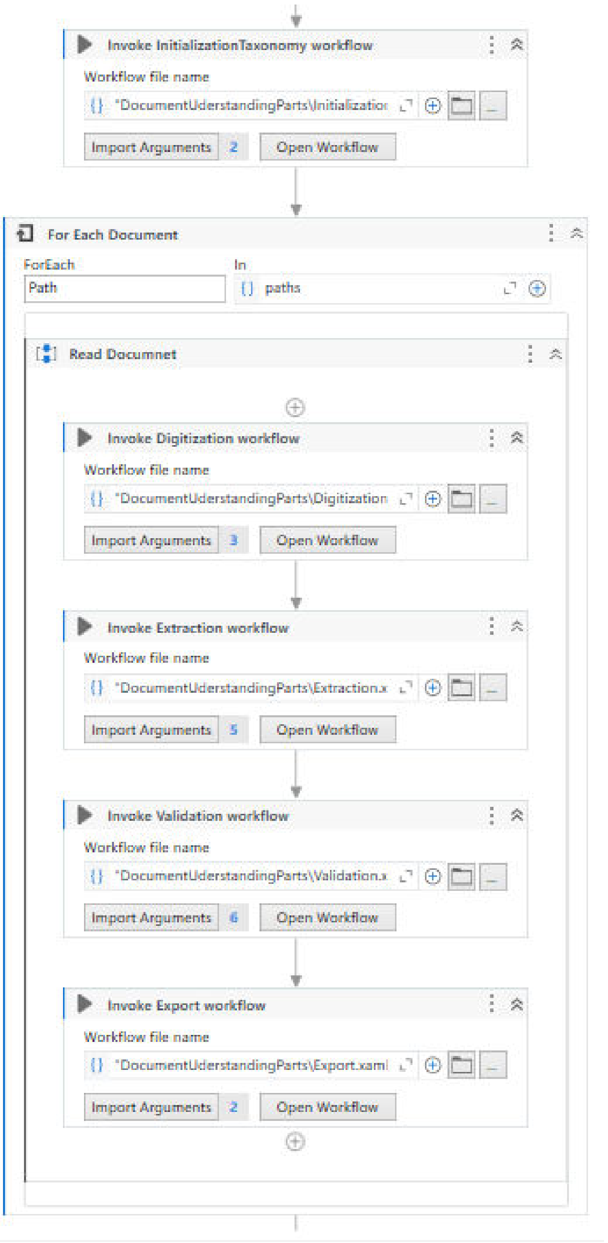
II. Document Understanding Flow

The Document Understanding flow was constructed using UiPath's official Intelligent OCR Package. In addition to the main workflow, the process consists of 5 additional workflows, each representing Document Understanding's main steps: Initialization and Taxonomy, Digitization, Extraction, Validation and Export Data.

In addition to this, a "Document Processing" subfolder has been added to the project structure, which contains the taxonomy file (explained in detail later), as well as the input files used for testing purposes and the promptly generated output files.

A. Main Workflow

As shown in Fig. 1, the Main workflow starts by invoking the "InitializationTaxonomy" workflow. From there, the taxonomy is loaded and a list of the files that need to be processed is retrieved. Once the list of all available files is retrieved, the process starts iterating through each of them and invoking the workflows corresponding to each Document Understanding phase. It starts by digitizing the file to obtain machine-readable text and understand its contents. From there the file is sent to the Extraction step to identify the specific information that we are interested in retrieving from the file. Next comes the validation step, where a human will review the extracted data and correct it where necessary. Finally, the results from the extraction and validation steps can be exported and saved in an excel spreadsheet.


Invoke InitializationTaxonomy workflow
Workflow file name: "DocumentUderstandingParts\Initializatio
Import Arguments 2 | Open Workflow

For Each Document
ForEach: Path
In: paths

Read Documnet

Invoke Digitization workflow
Workflow file name: "DocumentUderstandingParts\Digitization
Import Arguments 3 | Open Workflow

Invoke Extraction workflow
Workflow file name: "DocumentUderstandingParts\Extraction.x
Import Arguments 5 | Open Workflow

Invoke Validation workflow
Workflow file name: "DocumentUderstandingParts\Validation.x
Import Arguments 6 | Open Workflow

Invoke Export workflow
Workflow file name: "DocumentUderstandingParts\Export.xaml
Import Arguments 2 | Open Workflow

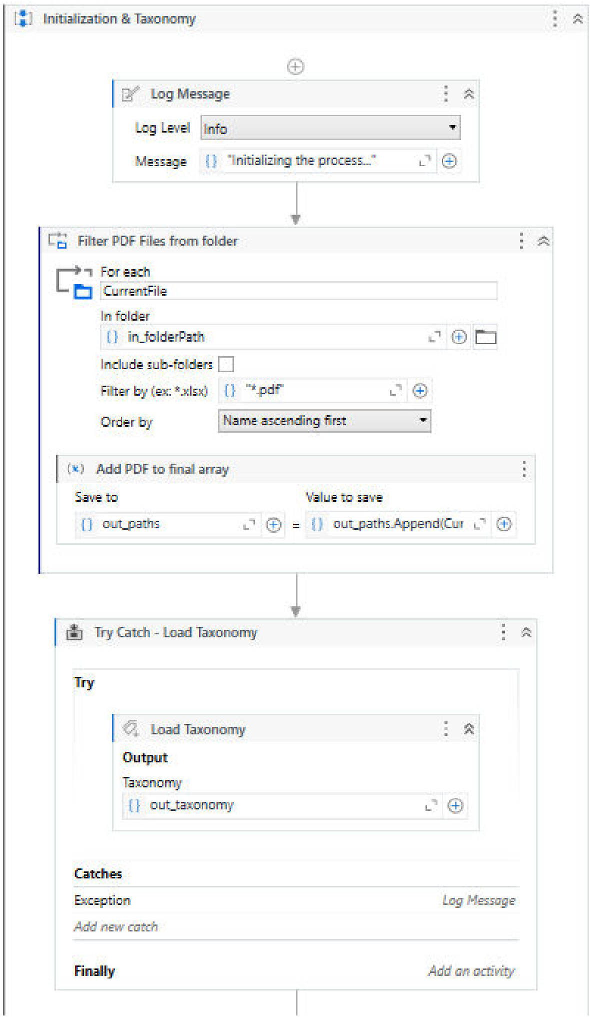
Fig. 1. Main workflow

B. *Initialization and Taxonomy*

The Initialization and taxonomy workflow is a simple workflow that has the purpose of retrieving PDF files from a specified location as well as reading the taxonomy file defined for the process. It accepts one input argument that is the folder where all the files the process uses can be found, and it has two output arguments, i.e. a list of files to be processed and the taxonomy data.

As shown in Fig. 2, the first activity completed in the Initialization and Taxonomy workflow is filtering files in the specified input folder by a '.pdf' extension and adding the results to the output array.

The second activity of the process is loading the Taxonomy.


Initialization & Taxonomy

Log Message
Log Level: Info
Message: "Initializing the process..."

Filter PDF Files from folder
For each: CurrentFile
In folder: in_folderPath
Include sub-folders ☐
Filter by (ex: *.xlsx): "*.pdf"
Order by: Name ascending first

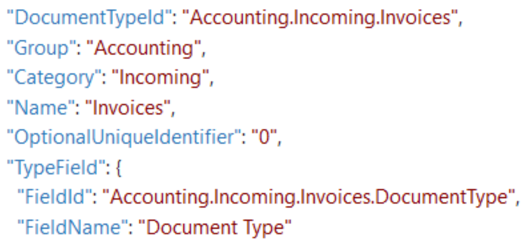
Add PDF to final array
Save to: out_paths = Value to save: out_paths.Append(Cur

Try Catch - Load Taxonomy
Try
Load Taxonomy
Output
Taxonomy: out_taxonomy
Catches
Exception — Log Message
Add new catch
Finally — Add an activity

Fig. 2. Initialization and Taxonomy workflow

Taxonomy refers to a JSON file that serves as metadata that the Document Understanding framework considers in each of its steps. This file contains the different document types expected in the process, as well as the individual fields that need to be extracted from each file type.

For the purposes of this project, a single file type was defined under the 'Accounting' section named 'Invoices', as visible in Fig. 3. The fields that will be extracted include information about the clients, the vendors, invoice number, date and due date, as well as the table of items and the subtotal and total calculations. Finally, the invoice documents might contain a comment or note, that will also be extracted.



```
"DocumentTypeId": "Accounting.Incoming.Invoices",
"Group": "Accounting",
"Category": "Incoming",
"Name": "Invoices",
"OptionalUniqueIdentifier": "0",
"TypeField": {
  "FieldId": "Accounting.Incoming.Invoices.DocumentType",
  "FieldName": "Document Type"
```

Fig. 3. Taxonomy file type definition

C. *Digitization*

The main purpose of this workflow, visible in Fig. 4, is digitizing the document, in order to obtain the document's text in a machine-readable form and the Document Object Model, which are returned as an output. The only needed input is the file's full path.

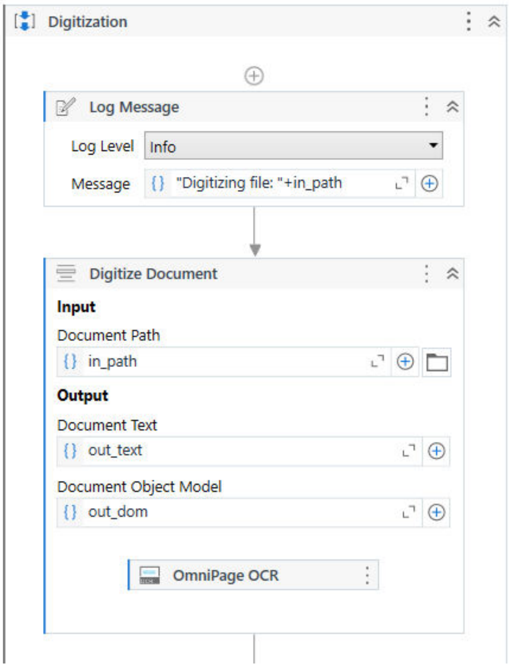
The digitization is achieved using UiPath's "Digitize Document" activity, which requires a certain OCR engine. Different engines will be tested in this step, to determine the best fit for different criteria.

D. *Extraction*

The Extraction workflow mainly consists of configuring the Data Extraction Scope activity. The received input consists of the document's path, the text retrieved during the digitization step, the DOM of the document, and the taxonomy. As a result, this workflow returns the extraction results. The structure of the Extraction workflow is shown in detail in Fig. 5.

In the Data Extraction Scope, in addition to the inputs specified above, it is also mandatory to specify either the Classification Scope or the Document Type Id. For the purposes of this paper, all test data will fall under the same Document Type, which is invoices.

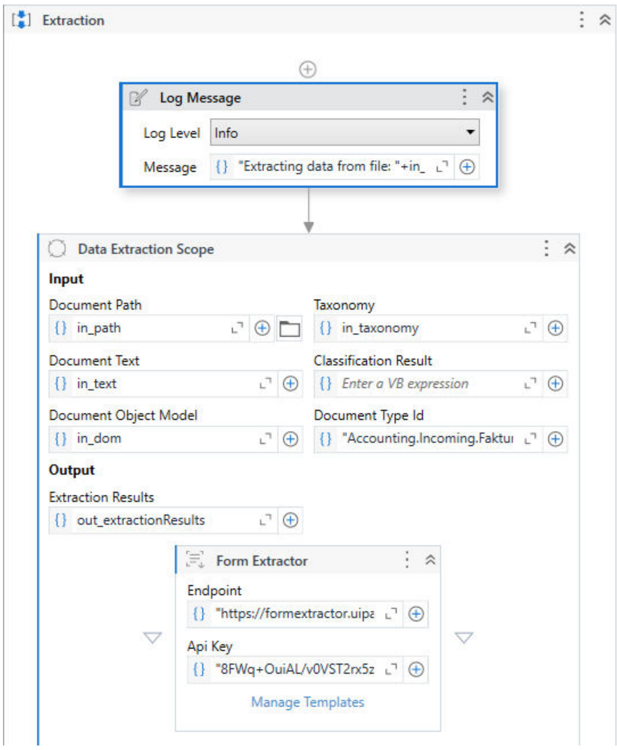
This activity also requires an extractor to be specified. For this project, the 'Form Extractor' was chosen as the best fit, because this extractor is suitable for documents where there are little to no variations in the layout. Since the documents that will be used for testing all have a similar structure, this extractor was the best fit.



Log Level: Info
Message: "Digitizing file: "+in_path

Input
Document Path: in_path
Output
Document Text: out_text
Document Object Model: out_dom

Fig. 4. Digitization workflow



Log Level: Info
Message: "Extracting data from file: "+in_

Input
Document Path: in_path
Taxonomy: in_taxonomy
Document Text: in_text
Classification Result: Enter a VB expression
Document Object Model: in_dom
Document Type Id: "Accounting.Incoming.Faktur
Output
Extraction Results: out_extractionResults

Endpoint: "https://formextractor.uipa
Api Key: "8FWq+OuiAL/v0VST2rx5z

Fig. 5. Extraction workflow

E. *Validation*

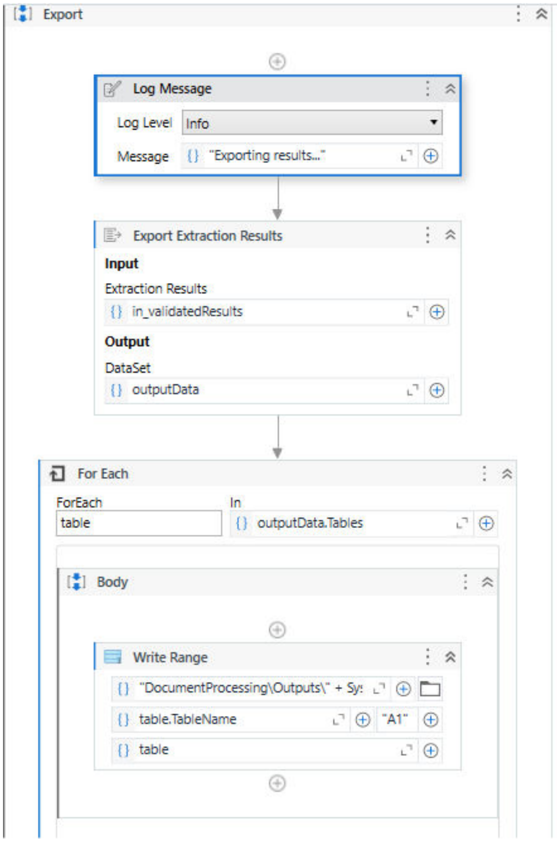
The purpose of the non-mandatory step of validation in document understanding in real life is for knowledge workers to review the automatically extracted results and correct them when necessary. In this project, the validation workflow will be used to maximize the possibility of a successful outcome.

It is built using UiPath's Validation Station and takes the document path, digitized text, and DOM, as well as the automatically extracted results. After the validation is performed, this step will return the new validated results.

F. Export

The final workflow that is a part of this project is the Export workflow. This flow, shown in Fig. 6, takes advantage of UiPath's Export Extraction Results activity to extract the results generated in the previous steps and save them in a data table.

The final results will then be displayed in a table in an excel spreadsheet. The Export Extraction Results activity returns a data set, that in this case consists of four data tables, a formatted and an unformatted version of the simple fields, and a formatted and unformatted version of the invoice items table. This activity is also configured to include the extraction confidence and OCR confidence of the results, which will be useful during testing.



Export

Log Message
Log Level: Info
Message: "Exporting results..."

Export Extraction Results
Input
Extraction Results: in_validatedResults
Output
DataSet: outputData

For Each
ForEach: table
In: outputData.Tables

Body

Write Range
"DocumentProcessing\Outputs\" + Sy:
table.TableName
"A1"
table

Fig. 6. Export workflow

III. Comparison of Available OCR Technologies

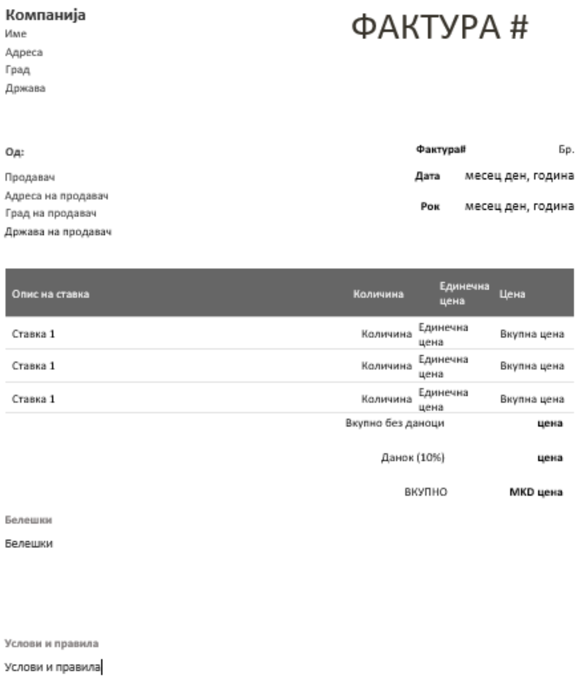
UiPath offers a selection of various OCR technologies [5] that report confidence when used for document understanding purposes, such as UiPath Document OCR, OmniPage OCR, Microsoft Azure Cloud Vision, Tesseract OCR, etc. Technologies like Microsoft OCR and UiPath Screen OCR are not meant for such usage and have been proven unsuccessful in Document Understanding scenarios and were therefore omitted. Additionally, the 'OCR – Japanese, Chinese, Korean' technology uses an OCR engine limited to understanding Japanese, Chinese and Korean and as such it was unfit for this research. Finally, for the purposes of this paper, only free OCR technologies were tested, so the Google Cloud Vision OCR and Abbyy Document OCR were not included.

A. Test Data and Criteria Definition

All mentioned technologies were tested using the exact same 3 documents with the purpose of finding the OCR technology that is best fit for processing documents in Macedonian language and were judged based on the results shown in the process. The test data includes:

- A digital Invoice in PDF form written in Macedonian language (Cyrillic).
- A scanned Invoice in PDF form written in Macedonian language (Cyrillic).
- A handwritten Invoice in PDF form written in Macedonian language (Cyrillic).

The layout of the invoices is shown in Fig. 7.



Компанија
Име
Адреса
Град
Држава

ФАКТУРА #

Од:
Продавач
Адреса на продавач
Град на продавач
Држава на продавач

Фактура# Бр.
Дата месец ден, година
Рок месец ден, година

Опис на ставка	Количина	Единечна цена	Цена
Ставка 1	Количина	Единечна цена	Вкупна цена
Ставка 1	Количина	Единечна цена	Вкупна цена
Ставка 1	Количина	Единечна цена	Вкупна цена
	Вкупно без даноци		**цена**
	Данок (10%)		**цена**
	ВКУПНО		**MKD цена**

Белешки
Белешки

Услови и правила
Услови и правила

Fig 7. Invoice layout

The fields that the process will attempt to extract from each of the documents includes the company, name, address, city and country, vendor name, address, city, invoice number, date of issuing the invoice and due date. Additionally, the table of items, including their description, quantity, rate and price, as well as the subtotal, total and taxes values will also be extracted. Finally, the process will attempt to retrieve the notes, rules and conditions from the page.

The retrieved data for each file will be saved in a final excel spreadsheet, that will contain a "raw" and "formatted" version of the extracted data.

The different OCR techniques will be evaluated, based on how much of the specified data they will be able to retrieve, how accurate the extracted data will be, and the confidence that they show.

B. *Evaluation of OCR Engines Based on Results*

- UiPath Document OCR

The UiPath Document OCR is UiPath's very own OCR engine, designed for extracting text from an image, as well as word positions, and it is optimized for processing document images. Currently, this engine is limited to processing invoices and receipts.

For the purposes of this paper, the UiPath Document OCR was configured with the Document Understanding package, since unlike the Invoice package, this one includes support for languages using the Cyrillic alphabet [6]. The required API Key was generated using a community version of UiPath cloud.

Unlike the OmniPage and Tesseract OCR engines, the UiPath Document OCR doesn't have explicit support for the Macedonian language, so it is not surprising that the success shown was noticeably lower.

Although it managed to extract all 16 fields when processing the digital PDF document, there were some noticeable mistakes in the extracted data, including missing letters and mixing up of similar letters in the Cyrillic alphabet, as well as mixing up letters and numbers. For instance, 'б' was recognized as '6', and ‚В' as '8' on multiple occasions. The confidence was higher than 80% for most fields.

The same was visible in the extraction results for the scanned document, where in addition to the occasional mistakes, only 12 of the 16 fields were extracted. The confidence was noticeably lower than for the digital PDF, and the mistakes in the extracted data, including missing letters, mixing up of similar letters and numbers, and missing words, were noticeably more frequent in comparison to the digital document.

According to the official documentation, the UiPath Document OCR can process documents that include handwriting [7], so it is not surprising that this OCR managed to extract 11 out of the 16 fields in the handwritten document. Once again, mistakes were visible in the extracted results, especially in the table of items and the confidence was significantly lower in comparison to the digital and scanned results.

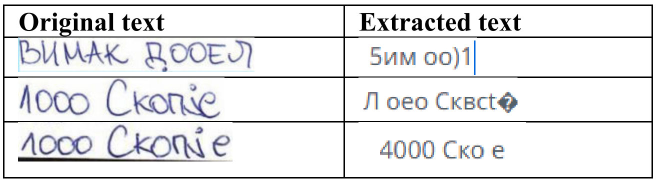
- OmniPage OCR

Some of the best results when processing the digital and scanned invoices, were by far achieved using the OmniPage OCR Engine. Designed for reading text from UI elements and images, the OmniPage OCR is often used in UiPath Document Understanding. This OCR technology supports over 240 different languages in their basic pack and additional 8 ones in their extended pack [8]. The OCR was configured with the 'Basic' engine pack, and the specified language value was Macedonian.

Since Macedonian is one of their supported languages, it is not surprising that it showed great results when tested with the digital PDF file. All values were extracted with no issues, with complete accuracy and a high confidence value. It is also important to point out that this engine is completely free and extremely easy to configure.

Similar results were achieved during the processing of the scanned invoice document, extracting all desired fields with high confidence. The only minor issues that were detected were the mixing up of similar letters in the Cyrillic alphabet, such as 'c' and 'e' on rare occasions.

When it comes to processing the handwritten document, the OmniPage OCR works only for hand-printed text, where the characters don't have a connection, best results being achieved for text of size 25 – 45 pixels. However, despite adjusting the handwritten document to fit those requirements, this engine was not able to show a lot of success. Not only did it manage to recognize just 6 of the total 16 fields in the invoice, but the detected text had a lot of character errors and was essentially unreadable. For instance, in Fig. 8 you can see a side-by-side comparison of some of the invoice values and their respective extracted results.



Original text	Extracted text
БИМАК ДООЕЛ	5им оо)1
1000 Скопје	Л оео Сквст�
1000 Скопје	4000 Ско е

Fig. 8. Original text vs extracted results

It is noteworthy, that often it was not only the Cyrillic handwritten letters that were not recognized but rather also the numeric values.

- Microsoft Azure Cloud Vision OCR

Microsoft Azure Cloud Vision OCR is Microsoft's OCR engine for extracting text from UI elements and images [9].

In order to be able to use this engine for the purposes of this paper, a free Azure account was created, Computer Vision was added to the Azure portal and the activity was configured with the generated API Key and Endpoint. Different scale factors were tested out, to find the best fit.

This engine offers an OCR API, which recognizes printed text and supports a large variety of languages and was used when testing the digital and scanned document. This API doesn't report confidence, so that criteria was omitted when testing this technology. This OCR also has an Azure Computer Vision Read API for handwritten and printed text, which is currently available only in English. This API was used when processing the handwritten document.

Despite being one of the hardest engines to configure, Microsoft Azure Cloud Vision OCR showed incredible results, when testing the digital and scanned file types.

The digital PDF file was processed with complete success, all values were extracted with no issues, with complete accuracy.

The same results were achieved with the scanned invoice, showing only minor discrepancies in some of the fields.

When processing the handwritten document, the Microsoft Azure Cloud Vision OCR had easily the worst results of all of the OCR technologies, which is not surprising, considering the Read API that it offers for processing handwritten documents only supports the English language. Most of the fields were extracted, however, all of the letters in the Cyrillic alphabet were swapped for a similar letter in the Latin alphabet or a digit.

- Tesseract OCR

The Tesseract OCR is a part of UiPath's Google OCR package and in UiPath it is primarily used for reading text from UI elements and images [10].

To maximize the possibility of success during this research, the highest available version of the engine was used, and it was configured by downloading their language package and setting the language to Macedonian. Different scale values were tested out, to determine the best fit for the used documents. It is noteworthy that this OCR offers a 'Invert' option, which is useful when the background of the documents is darker than the text color, but since that is not the case for the selected test documents this property was set to false.

When processing the digital and scanned PDFs, the Tesseract OCR showed moderate success, extracting most of the fields, with high confidence and occasional issues. However, this technology definitely showed less consistent results and was more prone to error, than some of the other free options, like the OmniPage OCR.

On the other hand, it showed better results in comparison to the OmniPage OCR, when processing the handwritten document. It managed to extract 14 of the 16 invoice fields, with a lot of success in most of the fields. However, some values were incomplete, some of the numbers were confused for letters, and vice-versa and there were some missing letters in the extracted words.

IV. Conclusion

In this paper we analyzed the possibilities and limitations for processing PDF documents written in Macedonian language using the some of the current OCR technologies for UiPath Document Understanding. In this study we considered 3 different types of documents: a digital PDF file, a scanned PDF file and a handwritten PDF file.

It is safe to say, that when it comes to Document Understanding for documents written in Macedonian, there is still room for improvement. However, the current technology that is made available by UiPath is undoubtedly competent and can be used for use cases that require the processing of documents written in Macedonian language. Both the OmniPage OCR and the Microsoft Azure Cloud Vision OCR were able to extract all required data from a digital PDF document without any human intervention. They were both also extremely successful when processing the scanned PDF, only facing minor issues. When it comes to processing handwritten PDF documents in Macedonian, the findings are a little different, as none of the tested OCR technologies showed complete success when processing handwriting in Macedonian.

However, even though the available OCR technologies might not be able to fully replace humans yet, with the help of human intervention, during the validation step of document understanding, success can undoubtedly be achieved. This can still help expedite the processing of documents in a lot of industries in Macedonia, as humans will only be required to intervene for values that failed to be extracted using document understanding. For activities that require 100% confidence when processing documents, it would be helpful to always include the step of human validation, to avoid minor discrepancies.

It is noteworthy that there are OCR technologies that were not included in this research and have the potential of achieving even better results than the ones in this paper.

References

[1] Brookings Institution, "Understanding the impact of automation on workers, jobs, and wages," [Online]. Available: https://www.brookings.edu/blog/up-front/2022/01/19/understanding-the-impact-of-automation-on-workers-jobs-and-wages/#:~:text=Indeed%2C%20digital%20automation%20since%20the,for%20highly%20educated%20analytical%20workers.

[2] VRabotuvanje, "Namaluvanje na administracija, samoposlužuvanje i avtomatizacija: Realni rešenija za nedostatokot na rabotnici," [Online]. Available: https://www.vrabotuvanje.com.mk/Vest/18408/Namaluvanje-na-administracijata-samoposluzhuvanje-i-avtomatizacija-Realni-reshenija-za-nedostatokot-na-rabotnici/2/

[3] Wikipedia, "UiPath," [Online]. Available: https://en.wikipedia.org/wiki/UiPath

[4] UiPath, "Document Understanding," [Online]. Available: https://www.uipath.com/product/document-understanding

[5] UiPath, "OCR Engines," [Online]. Available: https://docs.uipath.com/document-understanding/docs/ocr-engines

[6] UiPath, "ML Packages Supported Languages," [Online]. Available: https://docs.uipath.com/document-understanding/docs/ml-packages-supported-languages

[7] UiPath, "UiPath Document OCR," [Online]. Available: https://docs.uipath.com/activities/docs/ui-path-document-ocr

[8] UiPath, "Omnipage OCR," [Online]. Available: https://docs.uipath.com/activities/docs/omnipage-ocr

[9] UiPath, "Microsoft Azure Computer Vision OCR," [Online]. Available: https://docs.uipath.com/activities/docs/microsoft-azure-computer-vision-ocr

[10] UiPath, "Google OCR," [Online]. Available: https://docs.uipath.com/activities/docs/google-ocr