



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**AUTOMATICKÝ PŘEPIS DODACÍCH LISTŮ**

AUTOMATIC DELIVERY NOTE TRANSCRIPTION

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**DÁVID NECPÁL**

**VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. MICHAL HRADIŠ, Ph.D.**

BRNO 2020

## Zadání bakalářské práce



Student: **Necpál Dávid**  
Program: Informační technologie  
Název: **Automatický přepis dodacích listů**  
**Automatic Delivery Note Transcription**  
Kategorie: Zpracování obrazu

### Zadání:

1. Prostudujte základy konvolučních sítí a rozpoznávání ručně psaného textu.
2. Vytvořte si přehled o současných metodách rozpoznávání ručně psaného textu a analýzy tabulek.
3. Navrhněte systém umožňující automatický přepis ručně vyplněných dodacích listů a jejich zadávání do systému účetnictví.
4. Obstarejte si databázi vhodnou pro experimenty.
5. Implementujte navržený systém proveďte experimenty nad datovou sadou a otestujte napojení na systém účetnictví.
6. Porovnejte dosažené výsledky a diskutujte možnosti budoucího vývoje.
7. Vytvořte stručné video prezentující vaši práci, její cíle a výsledky.

### Literatura:

- Shi et al: An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition, 2015, arXiv:1507.00571.
- Kang L. et al.: Convolv, Attend and Spell: An Attention-based Sequence-to-Sequence Model for Handwritten Word Recognition. GCPR, 2019.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 4.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Hradiš Michal, Ing., Ph.D.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2019

Datum odevzdání: 31. července 2020

Datum schválení: 6. listopadu 2019

## Abstrakt

Cieľom tejto bakalárskej práce je vytvorenie systému pre automatický prepis dodacích listov – dokumentov s pevnou štruktúrou. Riešenie práce je rozdelené do dvoch častí. Prvou časťou je detekcia čiar tabuliek a z nich následná detekcia a extrakcia buniek, ktoré obsahujú požadované dáta. Druhá časť je samotné rozpoznanie ručne písaných numerických znakov v obrázkoch vyrezaných buniek. Výsledný systém dokáže pri kvalitne naskenovaných dodacích listoch detegovať bunky s požadovanými údajmi s presnosťou 100 %, pričom úspešnosť samotného rozpoznávania numerických znakov je viac ako 95 % pre samostatné znaky a vyše 92 % pre celé sekvencie znakov. Prínosom tejto práce je systém pre automatický prepis dodacích listov, ktorý zabezpečuje rýchlejšie a jednoduchšie inak zdĺhavé „prepisovanie“ obsahu dodacích listov do informačného systému maloobchodu. Využitím tohoto systému ušetrí pracovník na každom dodacom liste viac než 50 % času.

## Abstract

This bachelor thesis aims to create a system for automatic transcription of delivery notes - documents with a fixed structure. The solution is divided into two parts. The first part is table lines detection and subsequent detection and extraction of cells, that contain required data. The second part is handwritten numeric characters recognition in the images of the cutted cells. The resulting system can detect cells with the required data with 100% accuracy with well-scanned delivery notes, while the success rate of numerical character recognition is more than 95% for individual characters and more than 92% for entire character sequences. The benefit of this work is a system for automatic transcription of delivery notes, which provides faster and easier otherwise lengthy rewriting of the contents of delivery notes to the information system in the retail. By using this system, the employee saves more than 50% of the time on each delivery note.

## Klíčové slová

dodacie listy, OCR, HWR, HTR, ručne písaný text, rozpoznanie textu, prepis textu, spracovanie obrazu, tabuľky, spracovanie tabuliek, obrázkové dokumenty, dokumenty založené na obrázkoch, NN, neurónová sieť, CRNN, konvolučná rekurentná neurónová sieť

## Keywords

delivery notes, OCR, HWR, HTR, handwritten text, text recognition, text transcription, image processing, tables, table processing, document images, image based documents, NN, neural network, CRNN, convolutional recurrent neural network

## Citácia

NECPÁL, Dávid. *Automatický prepis dodacích listů*. Brno, 2020. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Michal Hradiš, Ph.D.

# Automatický přepis dodacích listů

## Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Michala Hradiša, Ph.D. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....  
Dávid Necpál  
29. júla 2020

## Podakovanie

Rád by som poďakoval pánovi Ing. Michalovi Hradišovi, Ph.D. za odborné vedenie, cenné rady a pomoc pri vypracovávaní tejto bakalárskej práce.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Definícia úlohy</b>	<b>3</b>
2.1	Aktuálna situácia evidencie pohybu tovaru v maloobchodnej prevádzke . . .	3
2.2	Dodacie listy . . . . .	6
2.3	Automatizácia prepisov . . . . .	6
<b>3</b>	<b>Detekcia a spracovanie tabuliek v dokumentoch</b>	<b>8</b>
3.1	Popis tabuliek . . . . .	8
3.2	Spracovanie tabuliek . . . . .	9
3.3	Existujúce riešenia pre spracovanie obrázkových dokumentov . . . . .	10
<b>4</b>	<b>Rozpoznanie ručne písaného textu</b>	<b>13</b>
4.1	Rozpoznávanie ručne písaného textu . . . . .	13
4.2	Existujúce riešenie pre rozpoznávanie znakov . . . . .	16
4.3	Neurónová sieť pre rozpoznávanie slov . . . . .	16
<b>5</b>	<b>Návrh riešenia</b>	<b>19</b>
5.1	Detekcia a spracovanie tabuliek dodacích listov . . . . .	19
5.2	Rozpoznávanie a prepis ručne písaného textu . . . . .	23
<b>6</b>	<b>Implementácia</b>	<b>25</b>
6.1	Použité nástroje . . . . .	25
6.2	Implementácia systému pre automatický prepis dodacích listov . . . . .	25
6.3	Pomocné nástroje . . . . .	28
<b>7</b>	<b>Experimenty a výsledky</b>	<b>30</b>
7.1	Detekcia buniek . . . . .	30
7.2	Rozpoznávanie textu . . . . .	31
7.3	Automatizácia prepisu dodacích listov . . . . .	32
<b>8</b>	<b>Záver</b>	<b>34</b>
	<b>Literatúra</b>	<b>35</b>
<b>A</b>	<b>Obsah pamäťového média</b>	<b>39</b>
<b>B</b>	<b>Inštalácia a návod k použitiu</b>	<b>40</b>

# Kapitola 1

## Úvod

Zápis pohybu tovaru je dôležitou súčasťou evidencie tovaru najmä podnikateľov a podnikov v oblasti predaja a výroby. Pre takýto zápis sa využívajú dodacie listy – dokumenty s pevnou tabuľkovou štruktúrou, do ktorých sa zapisujú potrebné údaje o danom tovare. V dnešnej dobe je žiadúce spracovávať a uchovávať informácie v elektronickej podobe, a to hlavne kvôli možnosti rýchleho prístupu a spracovania veľkého množstva dát. Z toho dôvodu je nevyhnutné dáta z dodacích listov prepísať do elektronickej podoby, najčastejšie v špecializovanom informačnom systéme. Takéto „prepísovanie“ dodacích listov môže byť z časového hľadiska pomerne náročné a vplyvom väčšieho množstva dát a monotónnosti práce môže dochádzať k občasným chybám. To zväčša spôsobuje mrhanie času pri hľadaní a opravovaní týchto chýb, v najhorších prípadoch dochádza aj k finančným stratám.

Cielom tejto práce je aplikácia využívajúca optické rozpoznávanie znakov (*Optical Character Recognition*, OCR), ktorá z naskenovaného dodacieho listu automaticky rozpozná ručne písané numerické znaky a vytvorí textový dokument obsahujúci prepis dát v elektronickej podobe, ktorý je následne využitý pre vloženie obsahu takto spracovaného dodacieho listu do informačného systému maloobchodu.

Systém pre vstup využíva obrázok dodacieho listu z digitálneho skenera, ktorý následne systém upraví. V obrázku je následne nájdená hlavná tabuľka, ktorá obsahuje požadované – ručne písané – informácie o tovare. Aby bolo možné tieto informácie získať, je potrebné pomocou detekcie čiar nájsť jednotlivé bunky tabuľky a tie, ktoré obsahujú potrebné informácie, z tabuľky vyrezať. Keď sú získané obrázky obsahujúce tieto bunky, je možné využiť už spomenuté rozpoznávanie znakov implementované pomocou konvolučnej rekurentnej neurónovej siete [39]. Pomocou neurónovej siete sa získajú digitálne prepisy všetkých ručne písaných informácií z jednotlivých buniek, z ktorých sa vytvorí spomenutý textový dokument slúžiaci pre urýchlenie elektronickej evidencie dodacích listov.

## Kapitola 2

# Definícia úlohy

Evidencia odovzdávania a prijímania tovaru je dôležitou súčasťou každej prevádzky, no nie všetky prevádzky môžu takýto pohyb tovaru evidovať priamo v informačnom systéme. V mnohých prevádzkach príjem a výdaj tovaru prebieha prevažne v skladových priestoroch, v ktorých nie je vždy k dispozícii počítač alebo prenosné zariadenie, ktoré by takýto pohyb tovaru umožňovali evidovať elektronicky.

Táto kapitola sa zameriava na popis aktuálneho stavu evidencie pohybu tovaru v maloobchodnej prevádzke železiarstva Remeslo Stavmat, s.r.o., problémy spojené s manuálnym zadávaním dát z dodacích listov do informačného systému, ako aj samotný informačný systém využívaný v tejto prevádzke. Ďalej popisuje dodacie listy – dokumenty pre evidenciu pohybu tovaru. Na konci kapitoly je zhrnutý cieľ tejto bakalárskej práce – automatizácii prepisov dát z dodacích listov.

### 2.1 Aktuálna situácia evidencie pohybu tovaru v maloobchodnej prevádzke

Remeslo Stavmat, s.r.o. je maloobchodná prevádzka, ktorá pre evidenciu pohybu tovaru a potvrdení o odovzdaní tovaru využíva predtlačené dokumenty – dodacie listy, ktoré pracovník vyplňa ručne. Obrázok 2.1 zobrazuje dodacie listy používané v tejto prevádzke. Po vyplnení dodacích listov je potrebné ručne zapísané údaje z dodacích listov vložiť do informačného systému maloobchodu. To je potrebné hlavne z dôvodu vystavovania faktúr a evidencie tovaru v účtovníctve. Tento proces nie je nijako automatizovaný, preto je v súčasnosti nutné každú položku z dodacieho listu zapísať do informačného systému manuálne.

Proces vyplňania dodacieho listu začína prijatím objednávky od odberateľa. Na dodací list sa uvedú potrebné údaje o odberateľovi, dátum a ostatné formálne náležitosti. Nasleduje samotná príprava tovaru pre odberateľa a vyplňanie obsahu dodacieho listu. Pre každú položku sa do dodacieho listu zapíše názov tovaru, jeho registračné číslo a množstvo. Nakoľko každý tovar nie je označený štítkom s cenou, do dodacieho listu sa ceny jednotlivých položiek a celková cena zapisujú až na konci – keď sú do dodacieho listu zapísané všetky položky odovzdávaného tovaru. Vtedy je dodací list odnesený do kancelárie k počítaču s informačným systémom, kde pracovník dopíše ceny položiek podľa údajov zo skladových zásob, urobí súčet ceny, skontroluje dodací list s objednávkou, podpíše sa a odovzdá originál dodacieho listu na priloženie k objednávke. Až potom pracovník môže obsah dodacieho listu vložiť do informačného systému.

**Dodacia list E.C.** 65501  
 Dodávateľ: REMESLO STAVMAT, s.r.o.  
 566 81 Břeclav, ul. Hradebná 14  
 IČO: 31566711, IČ DPH: SK202057496

Osoba: 16 G 2025  
 Odberať: ČERVENO STRAU  
 PUCHTŠTEJNA 11  
 2102 NAD HÁDOVOM 963 01

Číslo: KOTOL ZH-KULTIVOVÁ

Názov a druh tovaru	Množstvo	Jednotka	Za položku	Cena bez DPH		Cena za jednotku	Spolu DPH
				EUR	cent		
... 010640	1	X/A					
... 20212	2	X/A					
... 02022	2	X/A					
... 02026	1	X/A					
... 01086	1	X/A					
... 20193	1	X/A					
... 20196	1	X/A					
... 20210	2	X/A					
... 01067	1	X/A					
... 01076	1	X/A					
... 01026	1	X/A					
... 01049	1	X/A					
... 01024	1	X/A					
... 01023	1	X/A					
... 01021	1	X/A					
... 01022	1	X/A					
... 01078	1	X/A					
... 01010	1	X/A					
... 20193	1	X/A					
... 01020	1	X/A					
Odstavenie od DPH				Spolu DPH			
Spolu				EUR	cent	Celková suma EUR	

SPZ automobi: Vykazatel: Kontrolovat: Podpis skladca:

Číslo prenosu: HICIANJ

Dátum podpisu a pečiatka:

Tlačová výstražka EkonSpol® P.O. BOX 52, ul. K. Galvárisu 36, 619 08 Zlín, tel. a fax (041)7636 035

(a)

**PRÍJEMKA - VÝDAJKA - PŘEVODKA**

Firma: REMESLO STAVMAT, s.r.o.  
 566 81 Břeclav, ul. Hradebná 14  
 IČO: 31566711, IČ DPH: SK202057496

Odstavo: Remeslo Stavmat, s.r.o.  
 (Pozor: 000000)

Číslo: Ma dát: Ma dát:

Účel: Práporok odstav (číslo, dátum): Dať: Dať:

Pr. č.	Názov (druh, rozmer)	Číslo výrobku	M. j.	Stav: príjme množstvo	Množstvo		Cena za jednotku	Spolu EUR
					Získané	Vydané		
	Súdržba A 4140	010678				1		
	... 010478					1		
	... 010010					1		
	... 010016					1		
	... 010010					1		
	... 010472					1		
	... 010124					2		
	... 020933					1		

Podpis: Vykazatel: 6.7.2010 (dátum, podpis) Vydal: (dátum, podpis) Prijal: (dátum, podpis) Schválil: (dátum, podpis)

www.ekonspol.sk

(b)

Obr. 2.1: Vyplnené dodacie listy používané v prevádzke Remeslo Stavmat, s.r.o.

### Informačný systém pre maloobchody Ventus MOPwin

Ventus ISM MOPwin<sup>1</sup> je informačný systém pre maloobchodné predajne zabezpečujúci predaj tovaru, fakturáciu a správu skladu. V informačnom systéme je možné nájsť stav skladových zásob, ako aj prehľady faktúr a dodacích listov dodávateľov a odberateľov.

Informačný systém poskytuje dva spôsoby pre vytváranie / zápis dodacích listov. Prvý a aktuálne využívaný spôsob je manuálne vytváranie, pozri obrázok 2.2. Druhý spôsob je využitie služby *Import položiek z prenosného terminálu (záznamníka)*.

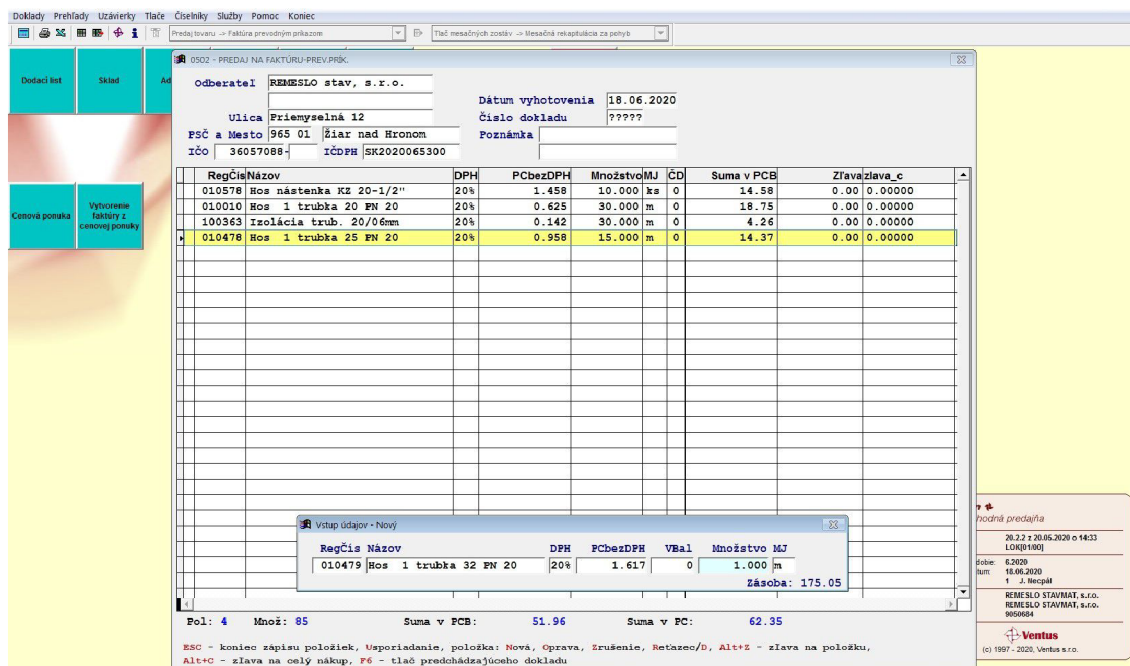
**Manuálne vytváranie dodacích listov** v informačnom systéme začína vložením údajov o odberateľovi. Následne sa zapisuje samotný obsah – položky tovaru z dodacieho listu. Tieto položky sa zapisujú postupne, pričom pre každú položku je potrebné napísať názov alebo registračné číslo výrobku, prípadne nájsť tento výrobok v zozname skladových zásob. Potom sa uvedie množstvo danej položky a po potvrdení zápisu sa pokračuje na ďalšiu položku. Po zápise všetkých položiek a vizuálnej kontrole správnosti obsahu je možné dodací list uložiť.

**Služba Import položiek z prenosného terminálu (záznamníka)** slúži na vloženie obsahu z pripraveného dokumentu do informačného systému. Táto služba<sup>2</sup> vyžaduje po-

<sup>1</sup>Informačný systém pre maloobchodné predajne od spoločnosti Ventus, s.r.o., viac informácií o produkte je možné nájsť na oficiálnej stránke produktu <https://www.ventus.sk/mopwin/>.

<sup>2</sup>Informácie o tejto službe mi zdelil obchodný zástupca firmy Ventus, s.r.o. cez e-mailovú konzultáciu.





Obr. 2.2: Manuálne vytváranie / vyplňanie dodacieho listu v informačnom systéme.

čiatocné nastavenie, ktoré je potrebné urobiť iba pri prvom využití tejto služby. V službe *Definovanie parametrov záznamníka* sa nastaví parameter *Cesta na dátový súbor*, kde bude informačný systém súbor hľadať. Súbor musí mať názov *vydaj.txt*, obsahuje polia oddelené čiarkou a množstvo používa ako desatinný oddeľovač bodku. Štruktúra textového súboru je čiarový kód alebo registračné číslo, množstvo, napríklad 660075,15.6.

Samotné použitie služby začína rovnako, ako v prípade manuálneho vytvárania dodacích listov, a to vyplnením údajov o odberateľovi. Po potvrdení sa namiesto evidovania položiek vyvolá služba *Import zo záznamníka*, respektíve zo súboru, ktorý pripravil záznamník. V prípade, že je textový súbor nájdený, do informačného systému sa vloží celý obsah textového súboru. Dodací list je možné následne uložiť alebo vložiť obsah ďalšieho textového súboru.

## Problémy manuálneho prepisovania dodacích listov

Väčší objem dát, ich podobnosť a monotónnosť činnosti môže pri takomto „prepisovaní“ dodacích listov viesť k problémom. Vznik problémov sa odzrkadľuje na množstve času, ktoré je potrebné venovať tejto činnosti, a to pri kontrole správnosti prepísaných dát, ako aj pri hľadaní a odstraňovaní vzniknutých chýb.

Chyby, ktoré môžu vzniknúť pri manuálnom prepisovaní dát z dodacích listov do informačného systému maloobchodu:

- chyba vzniknutá vynechaním riadku na dodacom liste,
- chyba vzniknutá horšou čitateľnosťou znakov dát,
- chyba vzniknutá nepozornosťou a prekliknutím sa na klávesnici.

V prípade, že sa tieto chyby neodhalia, môže dôjsť aj k finančným stratám, buď to na strane dodávateľa, alebo na strane odberateľa.

Chyby ale môžu vzniknúť už pri vyplňaní dodacích listov, no prejavajú sa až pri prepisovaní dát do informačného systému. Zväčša sa jedná o chybný názov alebo registračné číslo tovaru, prípadne zlý údaj o množstve tovaru. Takéto chyby sa ťažko odhaľujú a ešte ťažšie opravujú, nakoľko prepis dodacích listov do informačného systému neprebieha hneď a preto je zložité zistiť skutočný stav odovzdaného alebo prijatého tovaru.

## 2.2 Dodacie listy

Dodacie listy sa považujú za účtovné doklady, využívajú sa v oblasti predaja a výroby. Majú formu tabuľky a môžu byť vyplnené ručne alebo elektronicky. Obrázok 2.3 zobrazuje dodacie listy používané v prevádzke Remeslo Stavmat, s.r.o.

Dodacie listy slúžia ako potvrdenie odovzdaného alebo prijatého tovaru v stanovenom množstve, druhu a kvalite. Fyzicky sa prikladajú k dodávke. Vystavujú sa v dvoch vyhotoveniach, pričom originálne vyhotovenie patrí strane, ktorá za tovar platí. Dodacie listy nie sú vyžadované zákonom, ale na ich základe sa následne vystavujú faktúry.

Dodacie listy musia obsahovať určité náležitosti popísané v zákone 431/2002 Z.z. [40], ktorý upravuje obsah účtovných dokladov:

- slovné a číselné označenie účtovného dokladu,
- obsah účtovného prípadu a označenie jeho účastníkov,
- peňažnú sumu alebo údaj o cene za mernú jednotku a vyjadrenie množstva,
- dátum vyhotovenia účtovného dokladu,
- dátum uskutočnenia účtovného prípadu, ak nie je zhodný s dátumom vyhotovenia,
- podpisový záznam osoby zodpovednej za účtovný prípad v účtovnej jednotke a podpisový záznam osoby zodpovednej za jeho zaúčtovanie,
- označenie účtov, na ktorých sa účtovný prípad zaúčtuje v účtovných jednotkách účtujúcich v sústave podvojného účtovníctva, ak to nevyplýva z programového vybavenia.

O samotnom tovare sa evidujú hlavné informácie ako názov a/alebo registračné číslo tovaru, jeho množstvo a merná jednotka, prípadne aj ďalšie informácie ako kvalita, cena a podobne.

## 2.3 Automatizácia prepisov

Ako už bolo spomenuté, manuálny zápis dát dodacích listov do informačného systému pri násť určité problémy a jedná sa o časovo náročnú činnosť. V navrhnutom riešení proces automatizácie prepisov dodacích listov spočíva v tom, že na začiatku sa ručne vyplnený dodací list naskenuje a následne sa obrázok vloží do cieľovej aplikácie tejto práce, ktorá zabezpečí spracovanie obrázku a prepis textu do digitálnej podoby. Tento postup automatizácie je zvolený hlavne preto, že v prevádzke nie je v súčasnej dobe možné využiť prenosné elektronické zariadenia, ktoré by dodacie listy vytvárali elektronicky priamo do informačného systému. Obrázok sa v systéme spracuje, požadovaný obsah dodacieho listu sa pomocou metódy OCR prepíše do digitálnej podoby a vloží sa do textového súboru. Textový súbor sa následne použije na vloženie obsahu dodacieho listu do informačného systému.

MDL-111 DODACÍ LIST č. 66100 zo dňa 14.07.2020

Predajňa: REMESLO STAVMAT, s.r.o. REMESLO STAVMAT, s.r.o. Železná 74 966 01 Hlínik nad Hronom		Odberateľ: Binder Ján Svätý Anton 969 72 Svätý Anton	
---	--	---	--

RegCis	Názov	DPH	PredCena	Množstvo	MJ	Suma
360050	Dewalt klince 3,1x90/2400k	20%	92.00	1.000	ks	92.00
070587	Skrutka A 8x280 tesárska	20%	0.96	200.000	ks	192.00
070390	Skrutka A 5/ 50 Torx Zz	20%	2.50	5.000	cks	12.50
110780	Uholník staveb.105x105 KL5	20%	1.00	26.000	ks	26.00
110889	Uholník staveb.105x105 s r	20%	1.10	20.000	ks	22.00
110750	Uholník staveb. 90x90x65 s	20%	0.68	50.000	ks	34.00
110747	Uholník staveb. 90x90x65	20%	0.62	20.000	ks	12.40

Spolu k úhrade: 390.90 €

Dátum: 14.07.2020  
Tovar prevzal: .....  
číslo OP: .....  
Ďakujeme a tešíme sa na Vašu ďalšiu návštevu. Pečiatka predajne

Spracované programovým vybavením spoločnosti Ventus

(a)

**Dodací list č.:** 15.7.2020

REMESLO STAVMAT, s.r.o.  
Železná 74  
966 01 Hlínik nad Hronom  
IČO: 31566171 IČ DPH: SK2020479186

Dátum: 15.7.2020  
Objednávka čs./zo dňa: 20201884  
Tovar vyskladnený dňa:  
Tovar odoslaný dňa:  
Spôsob dopravy:  
Miesto určenia:

Dodávateľ: REMESLO STAVMAT, s.r.o.  
Odberateľ: KENTIN KONTINUIT  
SERVIS STAVMAT, s.r.o.  
BYSTRICKÝ 1608  
966 01 ŽELKOVICE

IČ DPH: .....  
dodávateľa: .....  
odberateľa: .....

Por. číslo	Názov a druh tovaru	Množstvo	Jedn. m.	Cena bez DPH		% DPH
				Za jednotku	Spolu	
10	OCEL PÚZ 28	60	27	2,40	720,00	20%
10	T 28	5	ks	6,79	700,45	20%
10	OBLEK 28 e.2	12	ks	3,65	700,02	20%
	UHTRUBOK 28	10	ks	2,45	700,00	20%
	PRECHOD 28x110Z	5	ks	6,27	700,02	20%
	VEŠP 1" FF	2	ks	8,15	0209,83	20%
	VEŠP 1 1/2" FF	5	ks	3,69	0202,50	20%
	T 1" - 1 1/2"	5	ks	1,14	0200,98	20%
	VEŠKOV 1"	10	ks	0,88	0201,73	20%

Oslbodené od DPH: .....  
Spolu DPH: .....  
spolu: EUR - cent Celková suma EUR

ŠPZ automobilu: ..... Zásielku prevzal: .....  
Vyskladni: .....  
Kontroloval: .....  
Podpis skladníka: .....  
Dátum, podpis a pečiatka: 15.7.20

Tlač Ekonspo®, ul. K cantoniu 38, 010 08 Žilina, www.ekonspo.sk tel/fax 041/7635035,7645375

(b)

Obr. 2.3: Predtlačené dokumenty pre evidenciu pohybu tovaru – dodacie listy. (a) Elektronicky vyplnený dodací list formátu A4. (b) Ručne vyplnený dodací list formátu A5.

Takýto proces automatizácie eliminuje časovú náročnosť manuálneho prepisovania dodacích listov do informačného systému, ako aj spomínané problémy a chyby. Oproti manuálnemu prepisovaniu je navyše vyžadované zariadenie pre skenovanie dodacích listov, ale zariadením s možnosťou skenovania dnes disponuje väčšina maloobchodných prevádzok.

Avšak aj pri navrhnutom riešení automatického prepisovania môže v malom percente prípadov dôjsť k určitým chybám, a to hlavne z dôvodu poškodenia dodacích listov alebo značným zhoršením čitateľnosti znakov dát, ktoré systém nedokáže odstrániť alebo s nimi nevie pracovať. Z toho dôvodu bude ďalšie rozšírenie systému zahŕňať pridanie jednoduchého užívateľského rozhrania, v ktorom bude možné prepis obsahu dodacieho listu skontrolovať a v prípade potreby aj jednoducho upraviť.

## Kapitola 3

# Detekcia a spracovanie tabuliek v dokumentoch

Každý deň sa vo svete vytvorí nespočetne veľa dokumentov a vo väčšine z nich sa pre zápis, zobrazenie a prácu s údajmi využívajú tabuľky, a to hlavne pre ich vysokú informatívnu hodnotu a prehľadnosť.

V oblastiach, kde sa informácie z tabuliek využívajú pre ďalšie digitálne spracovanie alebo archiváciu, je potrebné obsah tabuliek vložiť do počítača. Pri množstve dokumentov, ktoré je denno-denne potrebné spracovať, by ručné prepisovanie tabuliek do počítača nebolo možné, preto vznikajú nástroje a systémy, ktoré tento proces automatizujú. Niektoré nástroje sú jednoduché, iné zase komplexné s radou pokročilých funkcií. Proces detekcie a spracovania tabuliek v dokumentoch však často majú podobný postup a práve na to sa zameriava kapitola 3. Na začiatku sú stručne definované tabuľky. Následne sa kapitola venuje najpodstatnejšej časti, a to samotnej detekcii a spracovaniu tabuliek v dokumentoch.

### 3.1 Popis tabuliek

Keď sa povie tabuľka, väčšina ľudí si vie nejakú tabuľku predstaviť a popísať, čo to tá tabuľka vlastne je. Formálna definícia je však komplikovanejšia a naprieč literatúrami je formálny popis tabuľky nejednoznačný, pretože rôzni autori túto štruktúru popisujú rozdielne. Aj keď nie je formálny popis úplne jasný, súčasné formálne definície vychádzajú z rovnakého základu, a to definície tabuľky Jamesa Camerona z roku 1989 [3]. Vo svojej práci uviedol, že tabuľku je možné popísať ako objekt, v ktorom sa pomocou lineárnych vizuálnych prvkov popisujú vzťahy a logické spojenia medzi jednotlivými položkami. Základnou jednotkou tabuľky je položka obsahu, pričom to môže byť akýkoľvek vizuálny symbol. To znamená, že tabuľky môžu obsahovať slová, čísla, vzorce alebo rôzne grafické prvky [8].

Samotné tabuľky sa však klasifikujú aj podľa zložitosti, a to na jednoduché a zložené (komplexné). Jednoduché dvojrozmerné tabuľky majú štruktúru matice s jednoduchými riadkami a stĺpcami limitovanými na jednu úroveň hierarchie. Zložené dvojrozmerné tabuľky môžu obsahovať rekurzívne stĺpce a riadky [4]. Samotný obsah buniek v tom prípade nie je limitovaný na slová, čísla, vzorce alebo grafické prvky, ale tabuľky vo svojich bunkách obsahujú vnorené tabuľky, ktoré opäť vo svojich bunkách môžu obsahovať vnorené tabuľky.

Tabuľky môžeme rozdeliť aj na základe ich štruktúry, a to na tri typy [5]. Prvým typom sú tabuľky, ktoré používajú na oddelovanie buniek úsečky, horizontálne aj vertikálne. Druhý typ sú tabuľky obsahujúce horizontálnu čiaru na oddelenie riadkov a medzery na oddelenie

Položka 1	060 261	17,43	Položka 1	060 261	17,43	Položka 1	060 261	17,43
Položka 2	730 540	31,15	Položka 2	730 540	31,15	Položka 2	730 540	31,15
Položka 3	111 235	142,00	Položka 3	111 235	142,00	Položka 3	111 235	142,00
Položka 4	927 183	99,20	Položka 4	927 183	99,20	Položka 4	927 183	99,20

(a) (b) (c)

Obr. 3.1: (a) Tabuľka typu 1 s čiarami pre oddelenie riadkov a stĺpcov. (b) Tabuľka typu 2 s čiarami pre oddelenie riadkov a medzerami pre oddelenie stĺpcov. (c) Tabuľka typu 3 s medzerami pre oddelenie riadkov a stĺpcov. Obrázky sú prevzaté a upravené z [5].

stĺpcov. Tretí typ tabuliek používa pre oddelenie riadkov a stĺpcov len medzery. Obrázok 3.1 zobrazuje tri typy štruktúr tabuliek. Vzhľadom na to, že dodacie listy (ako aj veľké množstvo tabuliek v dokumentoch) majú prvý typ štruktúry tabuľky a pre tento typ sa používajú iné prístupy detekcie a rozpoznania tabuľky ako pre ostatné dva typy, práca sa zameriava prevažne na tabuľky, ktorých bunky sú oddelené horizontálnymi a vertikálnymi čiarami.

## 3.2 Spracovanie tabuliek

Aj keď je v dnešnej dobe veľa procesov už digitalizovaných a mnohé dokumenty sa vytvárajú, vypisujú, posielajú a ukladajú elektronicky, stále sa množstvo dokumentov vytvára, vypisuje, posielajú alebo ukladajú ručne, v papierovej podobe.

Na základe toho je dokumenty možné rozdeliť na dve hlavné skupiny, a to digitálne dokumenty a dokumenty založené na obrázkoch (obrázkové dokumenty). Pre obidva typy existujú rozdielne prístupy pre detekciu, rozpoznanie a následne aj spracovanie tabuliek.

Na to, aby tabuľky v dokumentoch mohli byť spracované, najprv musia byť nájdené. Obzvlášť dôležité to je v prípade obrázkových dokumentov, kde hlavnou úlohou pri detekcii tabuľky je zistiť, či dokument nejakú tabuľku obsahuje a kde sa v dokumente nachádza [4]. V prípade digitálnych dokumentov je nájdenie tabuľky taktiež náročnou úlohou. V textových dokumentoch je časté, že tabuľka neobsahuje žiadne fyzické stopy, ale pre oddelenie riadkov a stĺpcov sú využité len medzery. Proces spracovania tabuliek taktiež zahŕňa rozpoznanie a rozdelenie tabuľky na jej štruktúru a obsah. Rozpoznanie štruktúry je dôležité práve na to, aby bolo možné dáta extrahovať. Po tom, ako je štruktúra detekovaná, je možné rozpoznať samotné dáta. Na to sa najčastejšie využívajú systémy OCR.

### Obrázkové dokumenty

Kategória obrázkových dokumentov zahŕňa väčšie množstvo kombinácií, akým spôsobom môžu byť tieto dokumenty vytvorené. Najčastejšie sa však dá stretnúť s tromi druhmi dokumentov, a to dokumenty písané rukou, dokumenty vytvárané na počítači – v elektronickej podobe a následne sú vytlačené, často krát pre odoslanie poštou alebo priložené k zásielke, alebo ich kombinácia, čo znamená, že časť dokumentu je vytvorená elektronicky, hlavne štruktúra tabuliek, nadpisy, hlavičky a podobne, a dáta v dokumente sú vpisované ručne. Takéto dokumenty však neobsahujú žiadne elektronické metadáta, ktoré by umožnili priame rozpoznanie týchto dokumentov [4]. Prvým krokom pri digitalizácii takýchto dokumentov je naskenovanie alebo odfotenie dokumentu digitálnym skenovacím zariadením.

Najjednoduchším spôsobom pre detekciu a lokalizáciu tabuliek v obrázkových dokumentoch je analyzovať štrukturálne prvky – vertikálne a horizontálne čiary tabuľky. Jednou z možností je, že sa najprv nájdu segmenty čiar (ako napríklad v [24]) a následne sa extrahované úsečky využijú pre získanie tabuľkovej štruktúry.

V obrázkoch dokumentov často dochádza k nežiadúcemu šumu a prítomnosť rukopisu v ručne vyplnených tabuľkách zhoršuje detekciu vertikálnych a horizontálnych úsečiek tabuliek, kvôli čomu môže pri hľadaní tabuliek v dokumentoch dochádzať k chybám. Preto väčšina techník detekcie a extrakcie úsečiek tabuliek vyžaduje fázu predbežného spracovania, ktorá zahŕňa rôzne techniky úpravy obrazu, ako napríklad korekcia zošikmenia, odstránenie šumu, využitie morfológických operácií a transformácií a podobne. Viac detailov o technikách spracovania obrazu dokumentov je možné nájsť v knihe [12].

### 3.3 Existujúce riešenia pre spracovanie obrázkových dokumentov

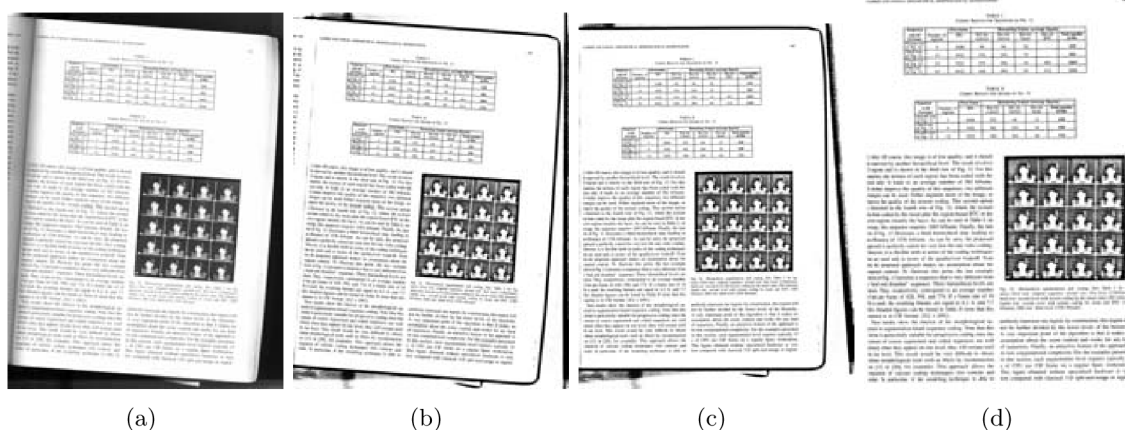
V období 21. storočia je digitalizácia dokumentov veľmi populárna, preto vznikajú mnohé systémy, ktoré sú pre detekciu a spracovanie tabuliek v dokumentoch priamo určené. Jedným z aktuálne dostupných riešení pre detekciu tabuliek v obrázkových dokumentoch je prístup od B. Gatosa a spol. [10]. Nižšie bude tento prístup podrobnejšie popísaný, a to hlavne z dôvodu, že čiastočne inšpiroval aj moje riešenie systému pre automatický prepis dodacích listov. Detekcia tabuliek je v spomenutom prístupe rozdelená na tri kroky, a to predspracovanie obrázkov, detekcia horizontálnych a vertikálnych čiar a detekcia tabuľky.

#### Predspracovanie obrázkov

Ako už bolo spomenuté, prvým dôležitým a nevyhnutným krokom pri detekcii čiar a tabuliek je predspracovanie obrazu. To zahŕňa napríklad binarizáciu, otočenie obrazu a korekciu zošikmenia, odstránenie šumu, dilatáciu obrazu a iné. Vyššie spomenutý prístup začína binarizáciou, kedy je šedotónový obrázok za použitia určitej prahovej hodnoty prevedený do binárnej reprezentácie. Body s vyššou intenzitou sa zmenia na čierne, ostatné sa zmenia na biele. Binarizácia sa vykonáva hlavne z dôvodu redukcie tieňov, nejednotného osvetlenia, nízkeho kontrastu, rôznych šmúh a podobne [11]. Po binarizácii nasleduje krok korekcie zošikmenia. V popisovanom prístupe je pre to využité určenie orientácie textu [44] a následne rýchla Houghova transformácia [31]. Jedným z ďalších krokov je odstránenie okrajov nesúvisiacich s obsahom dokumentov [1]. Takéto okraje vznikajú vždy, keď je pri skenovaní alebo foteaní dokumentu zachytené aj okolie mimo daného dokumentu. Tieto okraje veľmi výrazne zhoršujú spracovanie obrázkov dokumentov a preto je potrebné ich odstrániť. Na obrázku 3.2 je vidieť, ako prebieha proces predspracovania a jeho výsledok.

#### Detekcia čiar

Aby bolo možné detegovať tabuľku a samotný obsah v nej, je najprv potrebné nájsť čiary, ktoré túto tabuľku tvoria. Technika detekcie čiar je založená hlavne na spracovaní tmavých horizontálnych a vertikálnych čiar a na odhade obrazových a textových oblastí, aby bolo možné rozpoznať, ktoré úsečky patria do obsahu buniek a ktoré patria tabuľke. Pre dosiahnutie vyššej úspešnosti spracovania horizontálnych a vertikálnych čiar popisovaný prístup využíva pri ich detekcii morfológické operácie, a to hlavne dilatáciu a eróziu. Dilatácia slúži na spojenie blízkyh objektov alebo zaplnenie malých dier, čím sa objekt v obraze zväčší



Obr. 3.2: Ukážka prepracovania obrázkového dokumentu podľa [10]. (a) Vstupný šedotónový obrázok. (b) Výsledok po binarizácii a vylepšení obrazu. (c) Výsledok po korekcii zošikmenia. (d) Výsledok po odstránení okrajov nesúvisiacich s obsahom dokumentu. Obrázky sú prevzaté z [10].

a čiary, ktoré majú medzi sebou malé medzery, sa spoja a môžu vytvoriť jednotné čiary tabuľky. Erózia naopak objekt v obraze znižuje, odstraňuje okraje a rozširuje úzke medzery [32]. Po aplikácii morfológických operácií je finálny krok detekcie čiar rozdelený na dve časti, a to odhad horizontálnych a vertikálnych čiar a vylepšenie odhadu čiar pomocou odstránenia obrazových a textových oblastí.

## Detekcia tabuľky

Posledným krokom je samotná detekcia tabuliek, ktorá je v popisovanom prístupe rozdelená na dve fázy. Prvou fázou je detekcia priesečníkov úsečiek tabuľky, následne druhou fázou je detekcia tabuľky – rekonštrukcia.

- Detekcia priesečníkov úsečiek tabuľky (Detection of line intersections)**  
 Algoritmus detekcie priesečníkov [10] funguje v niekoľkých krokoch (pozri obrázok 3.3). V prvom kroku sa nájde koncový bod vertikálnej priamky a koncový bod horizontálnej priamky. Ak tieto body majú medzi sebou menšiu vzdialenosť ako ostatné body v okolí, sú označené ako priesečník tvoriaci rohový bod v tabuľke. Ďalším krokom je porovnávanie koncového bodu horizontálnej priamky alebo vertikálnej priamky s ďalším bodom priamky, ktorý nie je koncovým bodom a leží na horizontálnej alebo vertikálnej priamke. Tieto body musia mať najmenšiu vzdialenosť oproti ostatným bodom v okolí. Posledným krokom je nájdenie priesečníkov vertikálnych a horizontálnych priamok.
- Detekcia tabuľky a rekonštrukcia (Table detection and reconstruction)**  
 Proces pre detekciu tabuľky a rekonštrukciu začína tým, že sa odstránia všetky body patriace nájdeným čiarom. Následne sa všetky nájdené priesečníky zoskupia najprv horizontálne a potom vertikálne. Každá skupina je potom zarovnaná podľa strednej hodnoty vertikálnych alebo horizontálnych polôh pre horizontálne, respektíve vertikálne zoskupenie. Nakoniec je dosiahnutá rekonštrukcia tabuľky zakreslením vertikálnych a horizontálnych čiar, ktoré spájajú všetky páry priesečníkov čiar.

ID	1	2	3	4	5	6	7	8	9
Priesečníky čiar									

Obr. 3.3: Priesečníky čiar. Obrázok je prevzatý a upravený z [10].

Medzi ďalšie možné prístupy pre detekciu tabuliek v obrázkových dokumentoch patrí použitie modelu **CascadeTabNet** [34]. CascadeTabNet slúži ako na detekciu tabulky, tak aj na rozpoznanie štruktúry tabulky. Pre obidve úlohy využíva jednu konvolučnú neurónovú sieť (Convolution Neural Network – CNN). Model CascadeTabNet dokáže detekovať a rozpoznať tabulky využívajúce čiary pre ohraničenie riadkov a stĺpcov a aj tabulky bez ohraničenia – tabulky využívajúce medzery pre oddelenie riadkov a stĺpcov, avšak využitie konvolučnej neurónovej siete je iba pri tabuľkách bez ohraničenia. V prípade tabuliek bez ohraničenia sa odhadované bunky v tabuľke usporiadajú na základe ich polohy. Na základe odhadu buniek sa odhadnú riadky a stĺpce, vďaka čomu je možné nájsť aj bunky, ktoré neboli v prvotnom odhade nájdené. Pre tabulky s ohraničením je pre detekciu čiar využitý tradičný algoritmus pre detekciu čiar tabulky. Bunky sú následne identifikované pomocou priesečníkov čiar. Následne je využitý ďalší algoritmus pre detekciu textovej oblasti v samotných bunkách.



## Kapitola 4

# Rozpoznanie ručne písaného textu

Po detekcii a rozpoznaní tabuľky prichádza na rad rozpoznanie obsahu tabuľky. Pokiaľ je obsah tabuľky tvorený textom, pre jeho rozpoznanie sa využívajú nástroje pre rozpoznávanie textu (Optical Character Recognition – OCR). OCR je prevod obrazu obsahujúceho textové znaky do digitálnej editovateľnej textovej podoby pre ďalšie spracovanie. Systém OCR pre strojové automatické rozpoznanie textu je ako kombinácia očí a mozgu človeka, kedy oči vidia text na obrázku a mozog tento text spracováva [30].

Rozpoznávanie textu je v dnešnej dobe veľmi rozšírené, avšak patrí medzi najnáročnejšie oblasti pri spracovaní obrazu. OCR neslúži len na rozpoznanie textového obsahu v tabuľkách, ale využíva sa v rôznych oblastiach verifikácie dokumentov, tvorbe digitálnych knižníc, spracovaní poštových adries, šekov, úradných dokumentov, daňových formulárov, digitalizácii historických dokumentov a podobne.

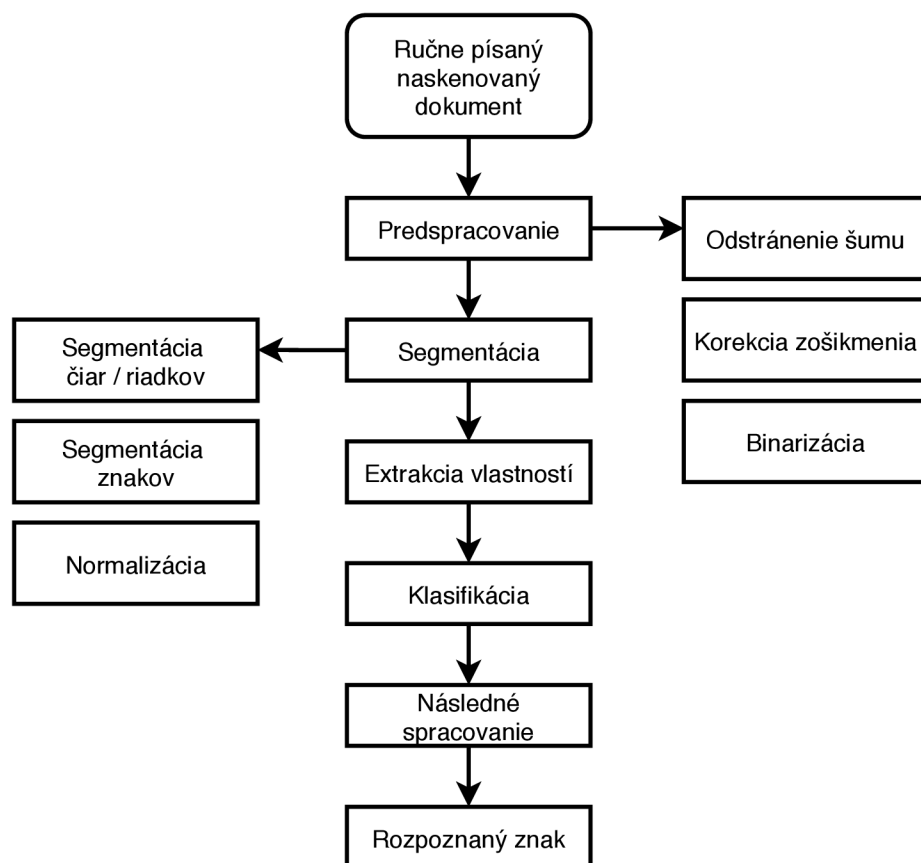
OCR je komplexná oblasť zahŕňajúca viac druhov rozpoznávania textových znakov. Hlavné delenie rozpoznávania znakov je podľa typu textu na strojové a písané rukou [27], v prípade textu písaného rukou ďalej podľa spôsobu získania dát na on-line a off-line. Vo všeobecnosti je off-line rozpoznávanie znakov oproti on-line ťažšie [33], nakoľko vlastnosti znakov počas on-line rozpoznávania môžu byť získavané ako aj z trajektórie pera a rýchlosti písania, tak aj z výsledného obrázku znaku. Pre off-line rozpoznávanie je k dispozícii len výsledný obrázok znaku. Obecne sa on-line rozpoznávanie znakov využíva pre ručne písaný text, zvyčajne písaný na digitalizačný tablet. Naopak off-line rozpoznávanie znakov je využité ako pre ručne písaný text, tak aj pre strojový (tlačný) text [36].

Pre delenie strojového a ručne písaného textu je rozpoznávanie ručne písaného textu zložitejšie, nakoľko v ručne písanom texte nie sú žiadne pravidlá na štýl písania, každý človek má iný tvar, sklon, štýl aj veľkosť písma [22].

Práca sa ďalej bude zameriavať na off-line rozpoznávanie ručne písaného textu, nakoľko cieľom bakalárskej práce je vytvorenie systému, ktorého súčasťou je prepis ručne písaného textu z obrázkov.

### 4.1 Rozpoznávanie ručne písaného textu

Proces rozpoznávania textu prebieha vo viacerých fázach, a to pedspracovanie, segmentácia, extrakcia vlastností, klasifikácia a následné spracovanie / rozpoznanie [22]. Nasledujúci popis procesu rozpoznávania textu bude vychádzať z článkov [22] a [27]. Obrázok 4.1 ukazuje celý proces rozpoznávania ručne písaného textu v dokumente. Neurónová sieť pre rozpozná-



Obr. 4.1: Hlavné fázy procesu rozpoznávania ručne písaného textu. Obrázok je prevzatý z [27].

vane ručne písaného textu v mojom systéme nevyužíva všetky nižšie popísané kroky, avšak pre úplnosť informácií a ilustráciu možného riešenia sú v práci stručne uvedené.

Nultým krokom pri rozpoznávaní ručne písaného textu je získanie obrázku obsahujúceho znaky alebo text, ktorý má byť rozpoznávaný. Najčastejšie sa na to využívajú digitálne skenovanie zariadenia, ktorých výstup je priamo vložený do systému pre rozpoznávanie ručne písaného textu. Takto získaný obrázok je potrebné upraviť, aby bolo možné znaky bez chýb rozpoznať. To sa vykonáva vo fáze predspracovania.

## Predspracovanie

Fáza predspracovania zahŕňa sériu operácií slúžiacich na zlepšenie obrázka, čo zaručuje lepšiu segmentáciu a rozpoznávanie znakov a textu. Medzi hlavné kroky predspracovania patria:

- **Odstránenie šumu**

Šum najčastejšie vzniká pri zachytávaní obrazu skenovacími zariadeniami alebo následnej kompresie obrazu pri prenose obrazu do počítača. Zvyčajne sa jedná o diery a medzery v riadkoch a čiarach, spojené znaky, periodický šum a podobne. Na odstránenie sa využívajú filtre [43] a morfologické operácie [37]. Najpoužívanejšími morfologickými operáciami sú dilatácia a erózia. Erózia objekt zmenší a medzery medzi objektami zväčší. Všetky body objektu, ktoré sa dotýkajú pozadia, sa zmenia na po-

zadie. Naopak, dilatácia objekt zväčšuje, pridáva body okolo hrán objektu. Každý bod, ktorý sa dotýka bodov objektu, sa zmení na bod objektu. Ďalšími využívanými morfológickými operáciami sú otvorenie a zatvorenie.

- **Normalizácia**

Normalizácia je proces odstránenia informácií, ktoré priamo nesúvisia s textom. Do tohto procesu zvyčajne patrí čistenie obrazu, korekcia zošikmenia a prípadne detekcia čiar a normalizácia veľkosti znakov [9], aby mali rozpoznávané znaky textu rovnakú veľkosť. Korekcia zošikmenia slúži na zarovnanie papiera dokumentu so súradnicovým systémom skeneru, v skratke zarovnať dokument do roviny. Na to sa využívajú rôzne techniky, jednou z nich je napríklad Houghova transformácia [22].

- **Kompresia**

Kompresia zahŕňa dve dôležité techniky, a to prahovanie (*thresholding*) alebo binarizácia [42] a riedenie (*thinning*) [23]. Binarizácia sa využíva na prevod farebného alebo šedotónového obrázku na binárny obrázok, vďaka čomu je spracovanie obrázku zrýchlené. Riedenie extrahuje informácie o tvaroch znakov.

## Segmentácia

Ďalším dôležitým krokom v každom systéme pre rozpoznávanie textu, ktorý ovplyvňuje úspešnosť samotného rozpoznávania, je segmentácia. Slúži na oddelenie textu od zvyšnej časti dokumentu, ako sú tabuľky, grafy, obrázky, riadky a podobne. Segmentácia je dôležitá hlavne vtedy, keď dochádza k prekrytiu alebo dotyku textu s inými objektami, čo je v prípade rozpoznávania ručne písaného textu bežné. Výstupom úspešnej segmentácie je obrázok obsahujúci iba požadovaný text. Segmentácia môže byť externá a interná [26, 29]. Externá segmentácia sa zameriava na oddelenie jednotlivých odsekov, viet alebo slov. Interná segmentácia predstavuje rozloženie znakov na podobrazy – časti znakov. Niektoré systémy využívajú segmentáciu znakov, a to pre získanie jednotlivých znakov zo slov. Pre dosiahnutie vyššej presnosti pri rozpoznávaní ručne písaných znakov sa všetky znaky normalizujú na rovnakú veľkosť. Pre to je možné využiť algoritmy na nájdenie, zoskupenie a rozdelenie oblastí [7], alebo sa využije algoritmus heuristickej segmentácie [15]. Ďalšie prístupy k segmentácii znakov je možné nájsť v [35].

## Extrakcia vlastností

Extrakcia vlastností slúži na nájdenie parametrov, ktoré presne a jednoznačne definujú tvar znaku. Metódy extrakcie vlastností [6] sa rozdeľujú na tri skupiny, a to štatistické vlastnosti, globálna transformácia a rozširovanie skupín (*global transformation and series expansion*) a geometrické a topologické vlastnosti. Prvá metóda ilustruje obraz ako štatistické rozdelenie bodov. Druhá metóda využíva rôzne techniky, ako napríklad Fourierova transformácia alebo Gaborova transformácia. Posledná metóda využíva konštrukčné prvky, napríklad slučky a krivky a využívajú sa spolu s fuzzy logikou na rozpoznávanie znakov [25]. Ďalší prehľad metód na extrakciu vlastností je možné nájsť v [6, 19].

## Klasifikácia

Z historického hľadiska, proces klasifikácie slúži na priradenie každého znaku do správnej triedy znakov, ide o takzvanú fázu rozhodovania. Pri vyššej kvalite extrahovaných vlastností z predošlého kroku je klasifikácia úspešnejšia. Pre ručne písaný text existuje viac prístupov

[21], ktoré sa delia na klasické techniky a techniky *Soft Computing*. Klasické techniky zahŕňajú *template matching*, štatistické techniky a štrukturálne techniky. Pod techniky *Soft Computing* spadá využitie neurónových sietí, techniky fuzzy logiky a techniky výpočtov pomocou evolučných algoritmov.

Jednou z techník *Soft Computing* je využitie systému od autorov H. C. Fu a spol. [18] pre rozpoznávanie rukopisu založeného na neurónovej sieti. Tento systém poskytuje predbežnú klasifikáciu, rozpoznávanie znakov a prispôsobenie systému podľa osobných potrieb. Hlavné rozpoznávacie moduly systému sú realizované pomocou samorastúcich neurónových sietí založených na pravdepodobnostných rozhodnutiach (*self-growing probabilistic decision based neural networks* – SPDNNs). Neurónová sieť používa jednu podsieť pre jeden objekt (znak), vďaka čomu je schopná lokálne a presne priblížiť rozhodovaciu oblasť každej triedy. Táto vlastnosť je žiadaná hlavne v oblasti rozpoznávania osobného rukopisu alebo identifikácie podpisu. SPDNN má taktiež vlastnosti ako nízka miera nesprávneho prijatia a nesprávneho odmietnutia.

Ďalšou možnosťou je využitie prístupu s názvom metóda boxu pre extrakciu funkcií pre rozpoznávanie ručne písaných znakov [16]. V tejto metóde je binárny obraz každého znaku rozdelený na fixný počet podobrazov nazývaných boxy. Pre rozpoznanie znakov je následne využitá neurónová sieť alebo fuzzy logika.

Alex Graves a spol. [13] navrhli prístup s využitím rekurentnej neurónovej siete špeciálne navrhutej pre úlohy sekvenčného značenia, v ktorých je problém so segmentáciou dát.

## 4.2 Existujúce riešenie pre rozpoznávanie znakov

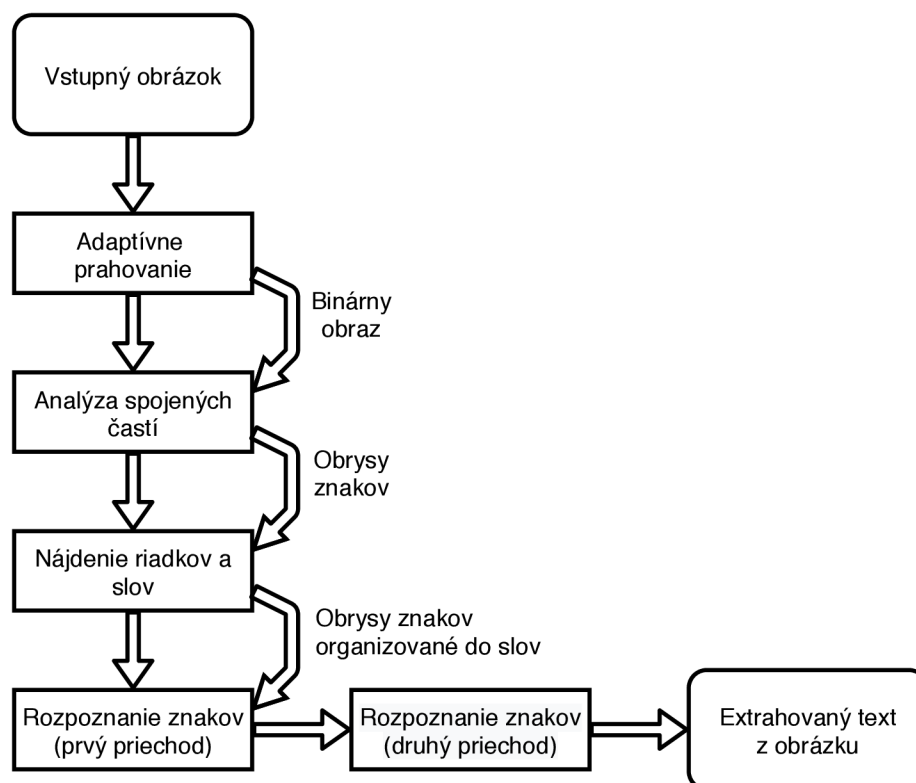
V dnešnej dobe existuje veľké množstvo OCR softvérov, no len málo z nich je zadarmo a s verejnými zdrojovými kódmi.

**Tesseract**[41] je jedným z open-source softvérov pre rozpoznávanie znakov. Bol vyvinutý spoločnosťou HP a od roku 2005 je k dispozícii ako open source<sup>1</sup>. Aktuálne je Tesseract vyvíjaný a udržiavaný spoločnosťou Google, pričom poskytuje podporu pre rôzne jazyky. Proces rozpoznávania textu systémom Tesseract prebieha v niekoľkých krokoch podľa architektúry zobrazenej na obrázku 4.2. Pre získanie binárneho obrázku zo vstupného čiernobielo alebo farebného obrázku obsahujúceho text je využité adaptívne prahovanie [38]. Aktuálna verzia Tesseractu využíva neurónovú sieť LSTM a metódu adaptívnej klasifikácie [41]. V tomto prípade je rozpoznávanie textu vykonávané v dvoch fázach. V prvom priechode je snaha o rozpoznanie každého slova z textu. Každé slovo, pri ktorom je kvalita rozpoznania dostatočujúca, je predané adaptívnemu klasifikátoru ako tréningové dáta. Ten v druhej fáze rozpozná text s vyššou presnosťou, nakoľko v prvej fáze obdržal niektoré slová ako tréningové dáta.

## 4.3 Neurónová sieť pre rozpoznávanie slov

V oblasti rozpoznávania ručne písaného textu je častou a dôležitou časťou rozpoznávanie ručne písaného textu vo vetách. Je veľa systémov pre rozpoznávanie ručne písaných znakov alebo slov s rovnakou dĺžkou, avšak tieto prístupy nie je možné s vysokou úspešnosťou využiť pre slová s rôznou dĺžkou, prípadne celé vety. Hlavnými dôvodmi je to, že slovo ako objekt môže byť zložené z variabilného počtu znakov – písmen a slovo nemôže byť označené

<sup>1</sup>Open source OCR softvér Tesseract dostupný na <https://github.com/tesseract-ocr/>



Obr. 4.2: Hlavné fázy procesu rozpoznávania ručne písaného textu v systéme Tesseract. Obrázok je prevzatý z [27].

štítkom a zaradené do konkrétnej triedy, nakoľko len pri anglickom jazyku sa odhaduje, že obsahuje okolo 450 000 slov.

Na základe toho vznikol model využívajúci konvolučnú rekurentnú neurónovú sieť (*Convolutional Recurrent Neural Network* – CRNN) [39], čo je kombinácia hlbokých konvolučných neurónových sietí (*Deep Convolutional Neural Networks* – DCNN) a rekurentných neurónových sietí (*Recurrent Neural Networks* – RNN). Výhodami je to, že môže byť trénovaný priamo celými slovami bez detailnejšej anotácie znakov. Taktiež má rovnaké vlastnosti učenia ako DCNN bez potreby ručne vkladaných vlastností alebo kroku predspracovania priamo z obrázkových dát. Má rovnakú vlastnosť ako RNN, a to schopnosť vytvorenia sekvencie štítkov. Nie je obmedzená dĺžka sekvenčných objektov – slov, vyžaduje sa iba rovnaká výška vstupných obrázkov vo fáze tréovania a testovania. Dosahuje porovnateľné alebo dokonca lepšie výsledky v rozpoznaní slov ako aktuálne využívané algoritmy a systémy. CRNN nie je primárne určená pre rozpoznávanie ručne písaného textu, avšak jej architektúra je natoľko robustná, že ju na túto účel je možné natrénovať a použiť s veľmi dobrými výsledkami.

## Architektúra siete CRNN

Architektúra siete [39] pozostáva z troch častí, a to konvolučných vrstiev, rekurentných vrstiev a transkripčnej vrstvy (*transcription layer*). CRNN na začiatku za použitia konvolučných vrstiev extrahuje vlastnosti zo vstupného obrázku. Po konvolučných vrstvách sú rekurentné vrstvy, ktoré vytvárajú predikcie pre každú skupinu vlastností. Posledná

vrstva v CRNN – transkripčná vrstva – prevádza predikcie jednotlivých rámcov na finálnu sekvenciu znakov – slov. Sieť je trénovaná s využitím *stochastic gradient descent* (SGD), kde gradienty sú počítané za pomoci algoritmu spätnej projekcie.

## Extrakcia vlastností

V CRNN sú konvolučné vrstvy vytvorené spojením konvolučných a *max-pooling* vrstiev z CNN modelov. Tieto vrstvy sa využívajú na extrakciu vlastností zo vstupného obrázku. Je potrebné, aby vstupné obrázky boli pred vstupom do konvolučných vrstiev normalizované na rovnakú výšku. Konvolučné vrstvy následne extrahujú vlastnosti – popisné vektory zo vstupného obrázka, ktoré ďalej slúžia ako vstup pre rekurentné vrstvy. Výsledkom extrakcie vlastností je sekvencia vlastností popisujúcich vstupný obrázok.

## Predikcia vlastností

Pre vytvorenie rekurentných vrstiev sa využíva hlboká obojsmerná rekurentná neurónová sieť. Rekurentné vrstvy zabezpečujú predikciu pre každý stĺpec z predošlých vrstiev. Výhodou je, že RNN má schopnosť zachytávať informácie v rámci sekvencie znakov, čo je v rámci rozpoznávania z obrázkov presnejšie oproti získavaniu informácií z každého znaku samostatne. Ďalšou výhodou je to, že RNN je schopná pracovať so sekvenciami znakov ľubovoľnej dĺžky.

Bežná RNN však trpí na problém miznúceho gradientu (*Vanishing gradient problem*) [2], čo znižuje rozsah kontextu, ktorý si RNN pamätá a to zafažuje proces trénovania. Preto je v konkrétnej CRNN využitá RNN typu *Long Short-Term Memory* [17] (LSTM), ktorá postupným spracovaním sekvencie znakov modifikuje internú pamäť.

LSTM sa skladá z pamätovej bunky a troch „brán“, konkrétne vstupná brána, výstupná brána a brána „zabudnutia“, kde pamäťová bunka ukladá predošlé kontexty rozpoznávania a vstupná a výstupná brána umožňujú, aby tento kontext pamäťová bunka uložila na dlhšiu dobu. Pre vymazanie pamäte v pamätovej bunke slúži brána „zabudnutia“.

Nakoľko je LSTM jednosmerná a používa iba predošlé kontexty, sú použité dve, navzájom opačne otočené LSTM pre vytvorenie obojsmernej LSTM. Zároveň je možné spojiť viac obojsmerných LSTM a tak vytvoriť hlboké obojsmerné LSTM [14] pre dosiahnutie vyššej úrovne abstrakcie a značného zlepšenia pri rozpoznávaní reči. Popisovaná CRNN využíva dve obojsmerné LSTM, pričom každá z nich obsahuje 256 pamäťových buniek.

## Transkripčia znakov

Transkripčia je proces zmeny predikcie jednotlivých rámcov z RNN na sekvenciu označení. Transkripčia sa delí na dve skupiny, a to na transkripciu bez slovníku, kde sú predikcie vytvárané bez slovníku a transkripciu so slovníkom, kde sú predikcie tvorené výberom sekvencie označení, ktorá má najvyššiu pravdepodobnosť. Slovník je skupina označení, na ktoré je obmedzená predikcia, niečo ako slovník kontroly pravopisu. V bezslovníkovom režime je najvyššia pravdepodobnosť určená predikciou siete. Popisovaná CRNN využíva kombináciu oboch prístupov – slovníkový aj bezslovníkový.

## Kapitola 5

# Návrh riešenia

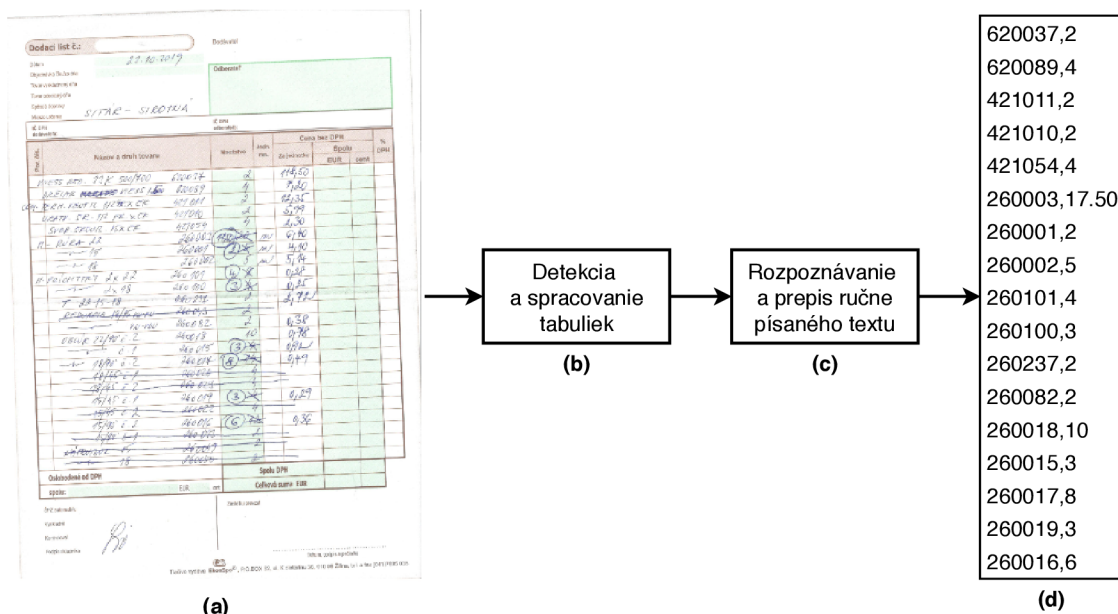
Táto kapitola popisuje návrh riešenia systému pre automatický prepis dodacích listov. Venuje sa postupu riešenia, úvahám o riešení a dôvody k vytvoreniu finálneho riešenia. Celá architektúra systému je zobrazená na obrázku 5.1.

Navrhnutý systém pre automatický prepis dodacích listov ako vstup očakáva obrázok dodacieho listu vo formáte JPG alebo PDF získaný z digitálneho skeneru. Ďalej je systém rozdelený na dve hlavné fázy. Prvou hlavnou fázou je detekcia a spracovanie tabuliek v dodacích listoch. V tejto fáze prejde obrázok krokom predspracovania, v ktorom je vykonaná korekcia zošikmenia, orezanie okrajov a z obrázku sa odstráni pozadie. Následne sa v obrázku dodacieho listu detegujú čiary tabuľky a pomocou nich sa nájdu požadované boxy označujúce bunky tabuľky. Nasleduje vyrezanie buniek z pôvodného obrázku. Výrezy sa uložia a v prípade potreby sa ďalej upravujú, nakoľko niektoré výrezy buniek obsahujú textový názov tovaru a jeho registračné číslo. V tom prípade sa textový názov tovaru z výrezu bunky odreže. Výstupom tejto fázy sú obrázky výrezov buniek len s registračným číslom tovaru alebo jeho množstvom. To znamená, že z každého vyplneného riadku tabuľky sú v ideálnom prípade vyrezané a uložené dva obrázky obsahujúce číselné hodnoty.

Tieto obrázky buniek ďalej slúžia ako vstup pre druhú hlavnú fázu systému – rozpoznávanie a prepis textu. V tejto fáze sú obrázky vložené do CRNN [39], ktorá zabezpečuje prepis numerických a matematických znakov do textovej podoby – v ASCII kódovaní. Všetky výrezy sú spracované v jednom behu programu a následne sú zapísané do textového dokumentu v poradí, v ktorom sa nachádzali na pôvodnom dodacom liste. Textový dokument sa následne využije pre vloženie obsahu dodacieho listu do informačného systému prevádzky Remeslo Stavmat, s.r.o.

### 5.1 Detekcia a spracovanie tabuliek dodacích listov

Prvým krokom celého systému je získanie obrázku dodacieho listu vo formáte JPG. Do systému je však možné vložiť aj súbor vo formáte PDF obsahujúci jeden alebo viac obrázkov dodacích listov na samostatných stranách. Takýto PDF súbor je následne konvertovaný na JPG obrázok z každej stránky PDF súboru. Po získaní JPG obrázku dodacieho listu je potrebné už len detekovať tabuľku a vyrezať požadované bunky. To však nie je vždy jednoduché a jednoznačné, nakoľko obrázok dodacieho listu zo skeneru zväčša obsahuje rôzne problémy zhoršujúce kvalitu samotnej detekcie tabuľky a následne aj rozpoznania textu v tabuľke. Z toho dôvodu je pred samotnou detekciou krok predspracovania, ktorý kvalitu obrazu vylepší a detekcia tabuľky a rozpoznania textu bude presnejšia. Celý proces vo fáze



Obr. 5.1: Architektúra systému pre automatický prepis dodacích listov. (a) Vstupný obrázok naskenovaného dodacieho listu. (b) Fáza detekcie a spracovania tabuliek. (c) Fáza rozpoznávania a prepisu ručne písaného textu. (d) Výstupný obsah dodacieho listu uložený v textovom dokumente.

detekcie a spracovania tabuliek dodacích listov je na obrázku 5.2. Ďalej budú jednotlivé kroky podrobnejšie popísané.

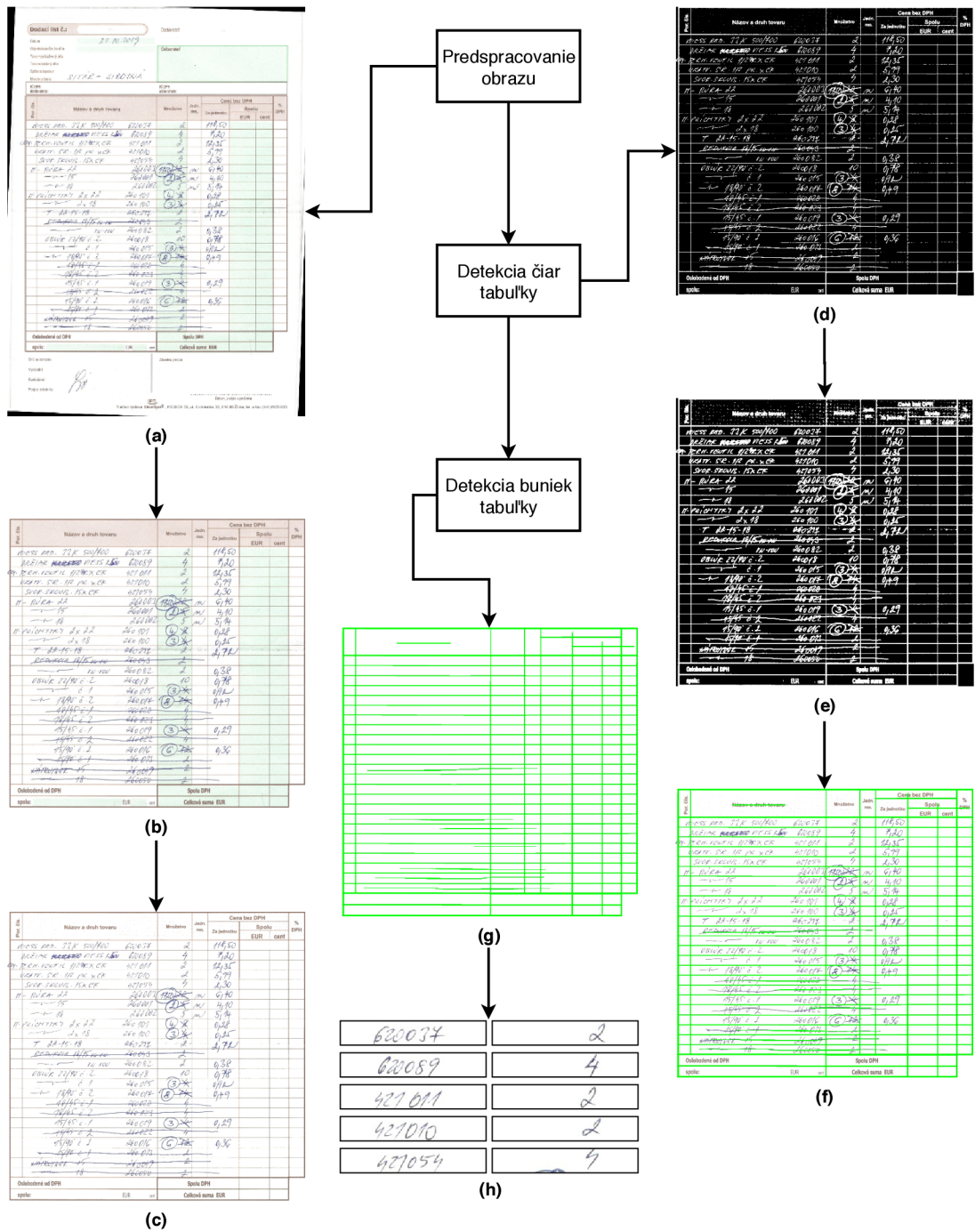
### Predspracovanie obrazu

Ako už bolo spomenuté v kapitole 3, prvou časťou pri detekcii tabuliek v obrázkových dokumentoch je predspracovanie obrazu. Vo vstupnom obrázku dodacieho listu je na začiatku nutné detegovať a odstrániť zošíkmenie, preto je na obrázok aplikovaný proces korekcie zošíkmenia. V prípade použitia skeneru k veľkému zošíkmeniu nedochádza, ale v určitých prípadoch aj menšie zošíkmenie môže spôsobiť chybu pri detekcii čiar tabuliek. Pre detekciu a korekciu zošíkmenia sa v obrázku deteguje najväčší box – najčastejšie hlavná tabuľka dodacieho listu. Zo súradníc rohových bodov sa vypočíta uhol zošíkmenia. Následne sa pomocou získaných bodov a vypočítaného uhlu zošíkmenia obrázok otočí tak, aby bol uhol zošíkmenia nulový – zošíkmenie z obrázku je odstránené alebo minimalizované.

Zároveň sú v tomto kroku odstránené nežiadúce okraje nepatriace k dodaciemu listu vznikajúce väčšou skenovacou plochou skenera oproti skenovanému papieru dodacieho listu, ako aj informácie, ktoré pre rozpoznávanie obsahu dodacieho listu nie sú potrebné. Pre to sú opätovne využité súradnice získané pri detekcii zošíkmenia – súradnice hlavnej tabuľky dodacieho listu. Obrázok sa podľa týchto súradníc oreže a v obrázku ostane len hlavná tabuľka, pozri 5.2 (b).

Z takto orezaného obrázku sa získajú rozmery, ktoré slúžia na výpočet hodnoty, podľa ktorej sa určí, či ide o dodací list typu A4 (2.1 (a)) alebo dodací list typu A5 (2.1 (b)). Na to je využitý fakt, že druhý typ dodacieho listu má pomer šírka / výška väčší ako prvý typ dodacieho listu.





Obr. 5.2: Architektúra časti systému, ktorá vykonáva fázu detekcie a spracovania tabuľiek dodacích listov. Fáza je rozdelená na tri časti – predspracovanie obrazu, detekcia čiar tabuľky a detekcia buniek tabuľky. (a) Proces korekcie zošikmenia. (b) Odstránenie prebytočných okrajov. (c) Vyčistenie pozadia od šmúh a iných nežiadúcich objektov. (d) Prevod obrázku do binárnej podoby. (e) Aplikácia morfológických operácií pre zvýšenie kvality čiar tabuľky. (f) Detekcia čiar tabuľky. (g) Vyčistenie obrázku od textu. (h) Detekcia a výrez buniek s požadovanými dátami.

Posledným krokom je vyčistenie pozadia, nakoľko dodacie listy často obsahujú šmuhy, zašednutie obrazu a iné rušivé prvky. Vyčistenie pozadia prebieha tak, že sa nastaví RGB hodnota a pokiaľ má pixel RGB hodnotu vyššiu ako je predom nastavená hodnota, jeho farba sa zmení na bielu. To sa opakuje pre všetky pixle v obrázku. Výstupný obrázok tak obsahuje zarovnanú hlavnú tabuľku dodacieho listu, zvyšné objekty mimo nej a okraje sú odrezané, pozadie tabuľky je bielo a farba ostatných objektov ostáva nezmenená.

## Detekcia čiar tabuľky

Ďalším, nie často využívaným, krokom je použitie segmentácie obrazu na základe farby [28]. Segmentácia obrazu na základe farby je využitá pre zvýraznenie čiar tabuliek s využitím farebného histogramu. Pre každý druh dodacieho listu získam jeden farebný histogram. Tento histogram sa počíta zo všetkých bodov čiar, a to pre tabuľky všetkých vybraných obrázkov dodacích listov konkrétneho druhu.

Po použití segmentácie obrazu na základe farby je s využitím prahovania obrázkov prevedený do binárnej podoby, aby sa odstránilo pozadie obrázku a zvýraznili čiary tabuľky a text. Následne využijem morfológické operácie z dôvodu, že počas skenovania, čistenia pozadia, segmentácie obrazu, prevodu obrázku do binárnej podoby alebo chybou pri tlači dodacích listov mohlo dôjsť k medzerám v čiarach tabuliek, čo je veľmi nežiadúce. Morfológické operácie objekty v obrázku upraví (rozšíria) do takej podoby, aby boli chyby pri procese detekcie čiar úplne odstránené alebo aspoň minimalizované. Výstupom týchto krokov je binárny obrázok obsahujúci biele čiary tabuľky, biely text a čierne pozadie.

Na tento obrázok je následne aplikovaná metóda pre identifikáciu čiar tabuliek, ktorá nájde úsečky spĺňajúce určité podmienky. Pre identifikáciu čiar tabuliek sú k dispozícii dve metódy. Štandardná Houghova transformácia je navrhnutá pre detekciu priamok, čo v prípade detekcie tabuliek nie je vždy vhodné. Preto je využitá druhá metóda – probabilistická Houghova transformácia, ktorá sa od štandardnej varianty líši tým, že navyše zisťuje hranice nájdených čiar. Nedetekuje teda priamky, ale ohraničené úsečky. Z týchto úsečiek sú získané ich súradnice a na základe súradníc sú zakreslené čiary prekrývajúce pôvodné čiary tabuľky na obrázku. Po tomto kroku je možné z obrázku „vyčistiť“ – nastaviť farbu pixlov na bielu – všetko okrem čiar zakreslených na súradniciach získaných z použitej metódy. Tento postup je zvolený hlavne preto, že výsledkom je biely obrázok obsahujúci len farebne rozlíšené čiary tabuľky, pozri 5.2 (g). Na takýto obrázok sa aplikuje prahovanie a získa sa binárny obrázok obsahujúci len pixle čiar.

## Detekcia buniek tabuľky

Posledným krokom pri spracovaní tabuliek je detekcia a „vyrezanie“ buniek obsahujúcich požadované dáta, a to konkrétne bunky s registračným číslom tovaru a jeho množstvom.

V tabuľke sa detegujú všetky bunky, a to tým spôsobom, že v binárnom obrázku z predošlého kroku sa vyhľadajú všetky kontúry a na základe ich súradníc sa vytvoria boxy – objekty tvorené štyrmi spojenými čiarami – a z týchto boxov sa získajú súradnice ich rohových bodov. V prípade, že tieto boxy spĺňajú predom určené rozmery a umiestnenie v tabuľke zodpovedajúce požadovaným bunkám, súradnice ľavého horného a pravého dolného rohového bodu sa uložia pre ďalšie spracovanie. Následne sa vytvoria dvojice „boxov“ obsahujúcich bunku registračného čísla a bunku množstva. Nakoľko sa stále pracuje so súradnicami, pre vytvorenie dvojíc sa používajú súradnice na y-ovej osi.

Jeden druh dodacích listov však využíva jednu bunku pre zápis názvu tovaru a zároveň aj registračného čísla tovaru. V tom prípade sa výrezy buniek ešte ďalej spracovávajú,

kedy je počas toho z výrezu bunky názov tovaru odstránený a vo výreze je ponechané len registračné číslo. Tento proces spočíva v tom, že v bunke sú pomocou morfológických operácií rozšírené pixle textu tak, aby názov tovaru tvoril jeden objekt a registračné číslo tovaru tvorilo druhý objekt. Následne sa už len získajú súradnice týchto objektov a obrázok bunky sa oreže podľa toho objektu, ktorý má x-ovú súradnicu ľavého horného rohového bodu väčšiu, nakoľko názov tovaru sa do bunky píše vľavo, registračné číslo vpravo – v obrázku bunky je tak ponechané len registračné číslo. Finálnym krokom je už len „vyrezanie“ buniek z tabuľky a ich uloženie ako samostatné obrázky.

Výstupom tohoto kroku sú obrázky – výrezy buniek s registračným číslom tovaru alebo množstvom tovaru. Tieto výrezy buniek sa ďalej využijú v druhej hlavnej fáze systému.

## 5.2 Rozpoznávanie a prepis ručne písaného textu

Druhá fáza systému ako vstup využíva obrázky buniek z výstupu predošlej fázy. Na začiatku je potrebné normalizovať rozmery obrázkov. Normalizácia veľkosti obrázku bunky funguje na princípe, že sa najprv vypočíta pomer  $r$  medzi požadovanou výškou a skutočnou výškou. Následne sa vypočíta nová šírka ako súčin  $r$  a skutočnej šírky obrázku bunky. Nakoľko však nie sú všetky výrezy rovnako široké, tak je po tejto zmene veľkosti obrázku ešte obrázok rozšírený bielym okrajom na ľavú aj pravú stranu. Po tomto kroku majú všetky obrázky buniek rovnakú – normalizovanú – výšku a šírku, pričom počas tejto normalizácie nedôjde k deformácii textu v obrázku.

Následne sa tieto obrázky použijú pre vstup do neurónovej siete, ktorá zabezpečí rozpoznanie a prepis textu z obrázkov do digitálnej podoby. Všetky obrázky – výrezy buniek z jedného dodacieho listu sú spracované v jednom behu neurónovej siete, aby bol čas rozpoznávania čo najkratší. Pre účely rozpoznávania numerických znakov (0 až 9), desatinnej čiarky a znaku „+“ je využitá CRNN [39], ktorá bola podrobnejšie popísaná v 4.3. Popis dátovej sady pre tréning neurónovej siete je uvedený v nasledujúcej podsekcii.

Výstupy neurónovej siete pre každý obrázok (refazce obsahujúce prepis textu z obrázka) sú postupne vkladané do textového dokumentu vo formáte `registračné číslo,množstvo`. Po skončení behu neurónovej siete – prepisu všetkých obrázkov vyrezaných buniek – je obsah textového dokumentu ďalej upravený. V prípade, že nie je rozpoznané registračné číslo alebo množstvo a tak netvorí platnú dvojicu čísel, je tento riadok z textového dokumentu odstránený. Systém o odstránení riadku užívateľa neinformuje, avšak pri finálnej kontrole je v prípade odstránenia riadku, ktorý mal byť súčasťou prepisu, tento fakt zistený užívateľom a riadok tak môže ručne doplniť.

Ak prepísaná hodnota obsahuje znak „+“ (Znak „+“ sa využíva pri hodnote množstva, napríklad `4+2,5`, iné matematické značky sa v dodacích listoch nevyužívajú.), je táto hodnota ďalej spracovaná ako matematický výraz a vo výstupnom textovom dokumente sa na jeho mieste nachádza jeho výsledok. Vo finálnom kroku sa celý spracovaný obsah vloží do textového dokumentu `vydaj.txt`, čo je zároveň aj výstupom navrhnutého riešenia systému.

### Dátová sada pre tréning a testovanie neurónovej siete

Pre účely tréningu a testovania neurónovej siete [39] som zvolil vytvorenie vlastnej dátovej sady, a to z dôvodu, že mám k dispozícii väčšie množstvo fyzických dodacích listov a dodacie listy v prevádzke Remeslo Stavmat, s.r.o. sú vyplňané len piatimi osobami, preto nie je potrebné tréningovať sieť na verejne dostupných dátových sadoch so stovkami tisíc obrázkov.

Obrázky pre dátovú sadu sú získavané už popísaným systémom, priebeh získania obrázkov pre dátovú sadu je rovnaký ako priebeh spracovania obrázku dodacieho listu. Obrázky v dátovej sade obsahujú jeden až osem znakov, konkrétne číselné znaky 0 až 9, znak desatinnej čiarky a znak „+“. Rozmery obrázkov sú vyššie uvedeným spôsobom normalizované na šírku 192 pixlov a výšku 32 pixlov. Dátová sada obsahuje 8 891 obrázkov – výrezov buniek, pričom sú rozdelené na tréning, testovanie a validáciu v pomere 80:10:10. Každý obrázok je označený anotáciou – prepisom obsahu obrázka. Všetky obrázky v dátovej sade som anotoval ručne. Dátová sada je vytvorená podľa dátovej sady MJSynth [20], na ktorú je optimalizovaná mnou použitá neurónová sieť. To znamená, že je vytvorených niekoľko textových dokumentov a priečinkov obsahujúci všetky obrázky – výrezy buniek. Textový dokument `imlist.txt` obsahuje názvy obrázkov, `lexicon.txt` obsahuje všetky anotované hodnoty z obrázkov. Najdôležitejší textový dokument `annotation.txt` obsahuje názov obrázka a pozíciu anotovanej hodnoty v dokumente `lexicon.txt`. Tri posledné textové dokumenty `annotation_train.txt`, `annotation_test.txt` a `annotation_val.txt` obsahujú náhodné hodnoty z `annotation.txt`, na základe ktorých sa následne určuje, ktoré obrázky budú slúžiť na tréning, ktoré na testovanie a ktoré na validáciu neurónovej siete.

Následne som využil nástroj `write_tfrecords` z GIT repozitára, ktorý z dátovej sady vytvorí tensorflow záznamy (*tensorflow records*). Tie sú následne využité neurónovou sieťou. Neurónová sieť je natrénovaná na 50 000 priechodoch (*epochs*), miera učenia (*learning rate*) je 0,01. Presnosť rozpoznávania samostatných znakov je 95 % a celých sekvencií znakov 92 %.

# Kapitola 6

## Implementácia

Táto kapitola sa zameriava na popis implementácie systému pre automatický prepis dodacích listov, ktorého návrh bol popísaný v kapitole 5.

### 6.1 Použité nástroje

Systém a jeho pomocné skripty sú implementované v programovacom jazyku Python<sup>1</sup> vo verzii 3.7. Väčšina algoritmov pre spracovanie obrazu je implementovaných za použitia knižnice OpenCV<sup>2</sup>, ktorá sa zameriava na počítačové videnie a spracovanie obrazu. Pre prácu s vektormi, maticami a viacrozmernými polami je v implementácii systému využitá knižnica NumPy<sup>3</sup>.

Pre rozpoznávanie ručne písaného textu je využitá konvolučná rekurentná neurónová sieť [39], konkrétne implementácia v jazyku Python za použitia softvéru TensorFlow dostupná v podobe GIT repozitára<sup>4</sup>.

Implementácia systému prebiehala na OS Windows 10 za použitia softvéru Anaconda<sup>5</sup>, a to až do doby implementácie neurónovej siete, kedy sa mi nepodarilo nainštalovať všetky potrebné balíky softvéru TensorFlow a preto bola implementácia presunutá na OS Linux Ubuntu, kde bol výsledný systém aj testovaný.

### 6.2 Implementácia systému pre automatický prepis dodacích listov

V nasledujúcej časti tejto kapitoly budú stručne uvedené zaujímavé a dôležité časti implementácie systému. Viac podrobností o jednotlivých funkciách v implementácii je možné nájsť v komentároch zdrojových kódov.

V prípade, že je vstupný súbor vo formáte PDF a obsahuje jeden alebo viac obrázkov dodacích listov, je súbor na začiatku spracovaný a obrázky z tohoto súboru sú konvertované do formátu JPG. Pre to je využitá funkcia `convert_from_path()` z knižnice `pdf2image`. Tieto obrázky sú následne uložené do dočasného priečinka pre ďalšie spracovanie. Väč-

---

<sup>1</sup><https://www.python.org/>

<sup>2</sup><https://opencv.org/>

<sup>3</sup><https://numpy.org/>

<sup>4</sup>Implementácia konvolučnej rekurentnej neurónovej siete podľa [39] dostupná pod MIT licenciou na odkaze [https://github.com/MaybeShewill-CV/CRNN\\_Tensorflow](https://github.com/MaybeShewill-CV/CRNN_Tensorflow).

<sup>5</sup><https://www.anaconda.com/>

šina vytvorených skriptov pri spustení využíva vstupné argumenty, ktoré sú spracované funkciami knižnice `argparse`.

Nakoľko je fungovanie systému komplexnejšie a je obsiahnuté vo viacerých ďalej popísaných skriptoch, všetky hlavné skripty sú spúšťané pomocou jedného hlavného skriptu `delivery_note_recognition.py`, ktorý v prípade potreby zabezpečí konverziu PDF súboru na JPG obrázky, následné spracovanie obrázku a výrez buniek, normalizáciu veľkosti buniek a aj finálny prepis textu do digitálnej podoby. To znamená, že celý navrhnutý systém je možné spustiť pomocou tohoto skriptu, ktorý ako potrebný vstupný parameter vyžaduje len vstupný súbor.

## Detekcia a spracovanie tabuliek

Proces detekcie a spracovania tabuliek je obsiahnutý v skripte `table_recognition.py`. Ako bolo napísané v návrhu riešenia, fáza detekcie a spracovania tabuliek začína procesom predspracovania. Tento proces začína vo funkcii `skew_corr_and_noisy_bg_rem()`, ktorej vstupný parameter je vstupný obrázok dodacieho listu. Funkcia s využitím vstavaných funkcií z knižnice `OpenCV` zabezpečuje detekciu zošikmenia podľa popisu z predošlej kapitoly a následne aj jeho odstránenie s využitím funkcií `getRotationMatrix2D()` a `warpAffine()`. Súradnice získané pre tieto dve funkcie sú zároveň využité pre odstránenie okrajov tabuľky dodacieho listu, ktoré nie sú pre proces detekcie tabuliek dôležité. Na to je využitá funkcia `getRectSubPix()`. Na konci funkcie je z rozmerov vyrovnaného obrázka bez okrajov zistený typ dodacieho listu, nakoľko v prevádzke Remeslo Stavmat, s.r.o. pracujú s dvomi typmi dodacích listov – dodací list vo formáte A4 (pozri 2.1 (a)) a dodací list vo formáte A5 (pozri 2.1 (b)). Výstupom tejto funkcie je už spomenutý vyrovnaný obrázok tabuľky s orezanými okrajmi a typ dodacieho listu. Proces predspracovania pokračuje vo funkcii `clear_background()`, kde sa za použitia masky vytvorenej funkciou `inRange()` nastaví pozadie tabuľky na bielu farbu. Výstupom funkcie je obrázok s vyčisteným pozadím bez zmeny ostatných objektov v obrázku.

Ďalším krokom vo fáze detekcie a spracovania tabuliek je proces detekcie čiar tabuľky. Tento proces začína funkciou `draw_lines()`. V systéme je pre detekciu čiar využitá spätná projekcia, konkrétne funkcia `CalcBackProject()`, ktorá na základe histogramu deteguje body obrázku, ktoré majú hodnotu farby obsiahnutú v danom histograme.

Následne sa obrázok prevedie do invertovanej binárnej podoby a na obrázok sa aplikujú funkcie pre vyplnenie a zväčšenie objektov, konkrétne dilatácia a funkcia `filter2D()`. Po tom už nasleduje najdôležitejšia časť tejto funkcie, a to detekcia čiar za použitia probabilistickej Houghovej transformácie. Pre využitie probabilistickej Houghovej transformácie je v knižnici `OpenCV` dostupná funkcia `HoughLinesP()`. Houghova transformácia vráti súradnice úsečiek, na základe ktorých sa do obrázku zakreslia farebne odlíšené čiary, pozri obrázok 5.2 (c). Výstupom funkcie `draw_lines()` a teda procesu detekcie čiar je obrázok tabuľky dodacieho listu s farebne zvýraznenými čiarami tvoriace kostru tabuľky.

Posledným krokom vo fáze detekcie a spracovania tabuliek je proces detekcie buniek tabuľky. Tento proces začína funkciou `get_cells()`. V nej sa na začiatku pôvodný obrázok tabuľky (5.2 (f)), v ktorom sú z predošlého kroku zakreslené farebne zvýraznené čiary na pozícii čiar tabuľky, vyčistí pomocou masky získanej funkciou `inRange()`. Následne sa pre získanie binárneho obrázku použije funkcia `threshold()` z knižnice `OpenCV`. V tomto binárnom obrázku sa následne nájdu kontúry pomocou funkcie `findContours()` opäť z knižnice `OpenCV`. Kontúry z `findContours()` sa v pomocnej funkcii `sort_contours()` pomocou funkcie `boudingRect()` zoradia podľa y-ovej osi a opätovným použitím tejto funkcie sú z každej

kontúry získané štyri hodnoty, a to x-ová súradnica, y-ová súradnica, šírka boxu a výška boxu. Tieto hodnoty sa ďalej využijú pre testovanie boxov a ak boxy spĺňajú podmienky na veľkosť, polohu v obrázku a počet nebielych bodov, súradnice boxov sa uložia. Vo funkcii `sort_cropped_imgs()` sa vytvoria dvojice – konkrétne súradnice boxu obsahujúceho registračné číslo výrobku a súradnice boxu obsahujúceho množstvo výrobku.

Po vytvorení dvojíc sa proces vracia do predošlej funkcie, kde sa kontroluje, či box pre registračné číslo výrobku neobsahuje zároveň aj jeho názov, nakoľko jeden typ dodacieho listu (pozri obrázok 5.1 (a)) využíva pre zápis názvu výrobku a jeho registračné číslo jednu bunku. V prípade, že box obsahuje obe položky, vo funkcii `crop_number_from_cell()` sa obrázok bunky spracuje. Na obrázok sa aplikuje funkcia `threshold()` a následne sa pixle objektov rozšíria funkciou `dilate()` pre získanie dvoch objektov, kde jeden objekt tvorí názov výrobku a druhý objekt je registračné číslo výrobku. Následne sa opäť získajú kontúry týchto objektov funkciou `findContours()` a z nich následne štyri vyššie spomenuté hodnoty. Pomocou nich sa určí objekt tvoriaci registračné číslo výrobku a podľa tohoto objektu (a jeho súradníc) sa obrázok bunky oreže. Nakoniec sa na základe súradníc boxov vyrežú jednotlivé bunky z tabuľky a uložia sa v podobe JPG obrázkov pre ďalšie spracovanie. Výstupom tohoto procesu sú výrezy požadovaných buniek z celého dodacieho listu.

## Rozpoznávanie a prepis ručne písaného textu

Prvým krokom, ktorý treba vo fáze rozpoznávania a prepisu textu v bunkách tabuliek urobiť, je normalizácia veľkosti vstupných obrázkov – výrezov buniek. Normalizácia veľkosti prebieha postupom popísaným v 5.2 v skripte `resize_image.py`. Pre zmenu veľkosti sa v skripte využíva funkcia `resize()` z knižnice `OpenCV`, na pridanie bielych okrajov na ľavú a pravú stranu obrázku bunky je využitá funkcia `copyMakeBorder()` opäť z knižnice `OpenCV`. Po behu tohoto skriptu majú všetky výstupné obrázky buniek veľkosť 192 pixlov na šírku a 32 pixlov na výšku.

Po tomto kroku už nasleduje hlavná časť systému, a to rozpoznávanie textu. Ako bolo napísané v návrhu riešenia, pre prepis textu je využitá konvolučná rekurentná neurónová sieť [39], konkrétne implementácia v jazyku Python za použitia softvéru TensorFlow dostupná ako GIT repozitár<sup>6</sup>. Implementácia je uverejnená pod MIT licenciou, čo dovoľuje dielo využívať a zdrojové kódy upravovať. Implementácia bola prevzatá z uvedeného GIT repozitára a následne čiastočne upravená. Pôvodná implementácia umožňuje predikciu iba pre jeden obrázok na jeden beh neurónovej siete, čo je v prípade desiatok obrázkov z každého dodacieho listu veľmi časovo náročné (beh programu pri približne 40 obrázkoch trval 10 minút), preto bolo nutné upraviť načítanie obrázkov do neurónovej siete.

Načítanie obrázkov bolo upravené v skripte `test_shadownet.py`. Pôvodná implementácia rozpoznávanie textu na obrázku realizovala tak, že obrázok bol načítaný zo vstupného argumentu skriptu a predikcia prebiehala iba pre tento obrázok. Preto bolo upravené spustenie skriptu, kedy vstupný argument nie je jeden obrázok, ale celý priečinok, v ktorom sa nachádzajú vytvorené výrezy buniek. Následne je vo funkcii `recognize()` spustený beh neurónovej siete za použitia softvéru TensorFlow a až potom sú zo vstupného priečinku načítané jednotlivé obrázky. Keď je obrázok – výrez bunky – načítaný, vykoná sa predikcia, výsledok sa zapíše a načíta sa ďalší obrázok. Keď sú rozpoznané všetky obrázky, beh neurónovej siete sa ukončí. Rozpoznávanie obrázkov po tejto úprave je značne rýchlejšie, pri 40 obrázkoch je čas behu skriptu približne 20 sekúnd. K ďalším úpravám v implementácii neurónovej siete nedošlo.

<sup>6</sup>[https://github.com/MaybeShewill-CV/CRNN\\_Tensorflow](https://github.com/MaybeShewill-CV/CRNN_Tensorflow).

Predikcie všetkých obrázkov z priečinku sa v priebehu behu procesu rozpoznávania textu ukladajú do dočasného textového dokumentu. Potom sa obsah textového dokumentu načíta a skontroluje sa platnosť rozpoznaných hodnôt. V prípade, že hodnota obsahuje znak „+“, napríklad 40+20, je s hodnotou pracované ako s matematickým výrazom, tento výraz je pomocou funkcie `eval()` vypočítaný a zapísaný je jeho výsledok. Finálne hodnoty sú zoradené do dvojíc, kde jedna hodnota predstavuje registračné číslo a druhá hodnota množstvo. V prípade, že jedna alebo druhá hodnota chýba, je celý riadok z finálneho prepisu dodacieho listu odstránený. Na konci sa predikcia všetkých výrezov buniek zapíše do textového súboru `vydaj.txt`, ktorý sa ďalej využije pre vloženie obsahu dodacieho listu do informačného systému prevádzky.

### 6.3 Pomocné nástroje

Počas implementácie systému bolo potrebné vytvoriť aj niekoľko pomocných nástrojov (skriptov), ktoré sa pri používaní systému k automatickému prepisu dodacích listov priamo nevyužívajú.

Ako bolo spomenuté v 6.2, pre detekciu čiar je využitá spätná projekcia, ktorá potrebuje farebný histogram. Na to, aby sa mohol daný histogram využiť, ho však najprv treba vytvoriť. Na to slúži skript `calc_hist.py`. Pred vytvorením histogramu som vybral šesť dodacích listov z každého typu a pre tieto dodacie listy som vytvoril takzvané skelety tabuliek, to znamená obrázky s priehľadným pozadím obsahujúce iba čiary tabuliek na rovnakých pozíciách ako na pôvodnom – vzorovom – obrázku. V skripte sú následne vzorové obrázky a obrázky obsahujúce skelety tabuliek načítané. Vo funkcii `create_new_image_from_scelet()` sa vytvorí nový obrázok o veľkosti vzorového obrázku, pričom všetky jeho pixle majú RGB hodnotu (0, 0, 0). Následne sa získajú súradnice každého bodu čiar z obrázku so skeletom tabulky. Vzápätí sa pre každý bod čiary získa farba bodu na danej súradnici zo vzorového obrázku a tento bod sa vloží do obrázku, ktorý bol na začiatku funkcie vytvorený. Týmto procesom vznikne obrázok obsahujúci iba čiary tabulky, ktoré majú rovnakú farbu ako čiary zo vzorového obrázku, zvyšok obrázku ostáva čierny. Z obrázku tak je možné jednoducho vypočítať, akej farby sú čiary týchto tabuliek. Tento proces sa vykoná pre všetky vzorové obrázky. V poslednom kroku sa vo funkcii `calc_histogram()` vypočíta jeden spoločný histogram pre všetky obrázky jedného typu dodacieho listu. Na to slúži funkcia `calcHist()` z knižnice `OpenCV`. Histogram sa následne uloží do textového súboru, aby ho bolo možné načítať a využiť pri segmentácii obrazu pomocou farby. Krok výpočtu histogramu a hlavne získavanie obrázkov pre výpočet histogramu je možné riešiť viacerými spôsobmi, avšak počas implementácie systému vyšiel tento postup ako najjednoduchší, preto je výpočet histogramu implementovaný týmto spôsobom.

Ďalší pomocný skript `xml_feeder.py` bol naimplementovaný pre anotáciu dát. Dôvodom pre vytvorenie daného skriptu bolo to, že dodacie listy, ktoré boli manuálne prepísané do informačného systému prevádzky, je možné spätne exportovať vo formáte XML, pričom z tohoto XML súboru je možné pomocou tohoto skriptu získať registračné číslo a množstvo z každej položky dodacieho listu. Tieto údaje sa následne mali priradovať k obrázkom výrezov buniek z konkrétnych dodacích listov. Tento spôsob sa však ukázal ako časovo veľmi náročný, preto som ho pri tvorbe dátovej sady nevyužil. Všetky obrázky z dátovej sady som anotoval ručne.

Funkčnosť skriptu `create_tfrecords_dataset.py` už bola z väčšej časti popísaná v 5.2. Skript slúži na vytvorenie dátovej sady pre tréning, testovanie a validáciu neurónovej siete. Proces tvorby dátovej sady začína načítaním textového súboru, ktorý obsahuje cestu a



názov každého obrázka z dátovej sady a ich prepisy – anotácie. Z tohoto textového súboru sú následne vytvorené všetky potrebné textové dokumenty. Výber obrázkov pre tréning, testovaciu a validačnú dátovú sadu je náhodný, na „zamiešanie“ obrázkov v datasete je využitá funkcia `shuffle()` z knižnice `random`.

Zároveň boli vytvorené dva skripty pre testovanie presnosti pri vyrezávaní buniek. Prvý, pomocný skript `get_coordinates_of_boxes.py` slúži na získanie súradníc buniek, ktoré majú byť vyrezané z tabuľky konkrétneho dodacieho listu. Na začiatku sa vyberie konkrétny dodací list, ktorý sa počas behu skriptu zobrazí v okne. Následne je potrebné manuálne označiť bunky na obrázku dodacieho listu, ktoré by mali byť vyrezané. Za použitia funkcie `setMouseCallback()` a pomocnej funkcie `click_and_crop()` sa na akciu `EVENT_LBUTTONDOWN` získajú počiatočné súradnice bunky a pri akcii `EVENT_LBUTTONUP` sú získané koncové súradnice bunky. Súradnice sa uložia do pola. Rovnakým spôsobom sa pokračuje na ďalšie bunky až kým nie sú označené všetky bunky z tabuľky. Následne je zobrazenie obrázku ukončené a obsah pola sa zapíše do textového súboru. Tento textový súbor obsahujúci referenčné súradnice buniek v dodacom liste je ďalej využitý v skripte `compare_cells.py`, ktorý slúži na porovnávanie manuálne označených „referenčných“ buniek so skutočne vyrezanými bunkami popísaným systémom z dodacieho listu. Na zistenie súradníc skutočne vyrezaných buniek je v skripte `table_recognition.py` zároveň implementované zapisovanie súradníc vyrezaných buniek do textového súboru. Ten je využitý pre popisované testovanie. Proces porovnávania súradníc buniek prebieha tak, že sa porovnávajú jednotlivé hodnoty zo súradníc, pričom je počítané aj s odchýlkou 10 pixlov. Po porovnaní všetkých súradníc skutočne vyrezaných buniek so súradnicami referenčných buniek sa vyhodnotí úspešnosť a chybovosť vyrezávania.

Posledný vytvorený skript pre testovanie je `compare_recognition.py`, ktorý porovnáva a vyhodnocuje úspešnosť prepisu obsahu dodacieho listu. Skript porovnáva referenčné hodnoty z prepisovaných buniek a skutočne získané hodnoty zo súboru `vydaj.txt` získaného po behu systému. Aktuálne tento skript vyhodnocuje úspešný prepis riadkov, a to v tom prípade, že riadok obsahuje dve hodnoty – registračné číslo a množstvo – a obe hodnoty sú správne.

## Kapitola 7

# Experimenty a výsledky

Testovanie systému prebiehalo v troch fázach. Prvé testovanie je na úspešné vyrezávanie buniek, v čom je zároveň zahrnutá aj správna detekcia samotnej tabuľky a čiar tejto tabuľky. Druhé testovanie – testovanie prepisu textu – je rozdelené do dvoch častí, a to testovanie na validačných dátach a testovanie rozpoznávania textu s finálnym modelom neurónovej siete na novo pripravených obrázkoch dodacích listov. Posledné testovanie je testovanie samotnej podstaty bakalárskej práce – časové porovnanie manuálneho a automatizovaného prepisovania dodacích listov do informačného systému.

Zatiaľ čo prvé dve kategórie testov je možné vykonávať automatizovane na počítači, tretiu kategóriu testov bolo potrebné vykonávať priamo v prevádzke Remeslo Stavmat, s.r.o. s pracovníkmi, ktorí budú tento systém obsluhovať aj v budúcnosti.

### 7.1 Detekcia buniek

Prvým krokom pri testovaní detekcie buniek je získanie referenčných súradníc buniek v každom testovanom dodacom liste. Pre to je použitý skript `get_coordinates_of_boxes.py` popísaný v 6.3. Pre testovanie detekcie buniek som vybral 22 dodacích listov typu A4 a 8 dodacích listov typu A5, to znamená dokopy 30 náhodných testovaných obrázkov. V každom obrázku sa teda ručne označia požadované bunky a súradnice z týchto buniek sa uložia do textového súboru. Následne je obrázok použitý ako vstupný obrázok pre systém a počas spracovania tabuľky a výrezu buniek v skripte `table_recognition.py` sú získané a uložené do textového súboru aj súradnice skutočne vyrezaných buniek. Tento proces sa opakuje pre všetkých 30 testovaných obrázkov.

Po získaní referenčných aj reálnych súradníc buniek všetkých dodacích listov sa využije skript `compare_script.py`, ktorý je opäť popísaný v 6.3. Pre testovanie boli dodacie listy naskenované skenerom v prevádzke Remeslo Stavmat, s.r.o., aby výsledky zobrazovali reálnu funkčnosť systému po nasadení do tejto prevádzky. Tabuľka 7.1 zobrazuje výsledky tohoto testu. Z tabuľky je vidieť, že pri vhodne naskenovaných obrázkoch dodacích listov systém deteguje všetky požadované bunky. Problémy pri detekcii buniek však nastávajú v prípade, že vstupný obrázok je príliš „bledý“ a čiary tabuliek sú nevýrazné, alebo naopak vtedy, keď je obrázok príliš „tmavý“. Vhodné nastavenie skeneru je pri detekcii buniek veľmi dôležité.

Typ dodacích listov	Počet dodacích listov	Počet referenčných buniek	Úspešne vyrezané bunky	Neúspešne vyrezané bunky	Celková úspešnosť vyrezávania [%]
Typ A4 (2.1 (a))	22	844	844	0	100
Typ A5 (2.1 (b))	8	138	138	0	100

Tabuľka 7.1: Testovanie úspešnosti vyrezávania buniek dodacích listov

## 7.2 Rozpoznávanie textu

Prvá časť testovania rozpoznávania (prepisu) textu – testovanie na validačných dátach z dátovej sady – prebiehala počas tréningovania neurónovej siete. Testovanie na validačných dátach je počas tréningovania automatizované a vykonáva sa pravidelne po predom nastavenom počte krokov tréningovania. Testovanie je zabezpečené skriptom `evaluate_shadownet.py` dostupnom v implementácii popísanej v 6.2. Tabuľka 7.2 zobrazuje výsledky získané z tohoto testovania. Z tabuľky je vidieť, že najlepšie výsledky rozpoznávania ručne písaného textu neurónová sieť dosahuje pri použití modelu získaného pri 41 000 tréningovacích epochoch.

Počet epoch	Presnosť prepisu samostatných znakov	Presnosť prepisu sekvencií znakov
5 000	0.916926	0.856027
10 000	0.942549	0.910714
20 000	0.945432	0.918527
40 000	0.950826	0.924107
41 000	0.951012	0.925223
50 000	0.950640	0.924107

Tabuľka 7.2: Testovanie neurónovej siete na validačných dátach počas jej učenia.

Druhá časť testovania rozpoznávania textu spočíva v samotnom vyhodnocovaní úspešnosti prepisu textu. Pre testovanie bolo vybraných 47 dodacích listov. Týchto 47 testovaných dodacích listov obsahuje 1 670 relevantných výrezov buniek, to znamená 835 riadkov obsahujúcich dve kategórie vyrezaných buniek, a to bunky obsahujúce registračné číslo, ktoré má vždy dĺžku 6 číselných znakov a bunky obsahujúce množstvo. Tabuľka 7.3 zobrazuje výsledky tohoto testovania. Pri teste bolo zistené, že 67 riadkov nebolo prepísaných správne. V týchto riadkoch môže byť zlý prepis registračného čísla alebo množstva, prípadne oboch čísiel, alebo tento riadok nebol do finálneho textového súboru zahrnutý vôbec, a to z dôvodu, že jedno alebo obe čísla z dvojice neboli rozpoznané – prepísané – vôbec. Tento údaj teda udáva, koľko riadkov vyžaduje úpravu (alebo doplnenie) pracovníkom pred tým, než bude výsledný textový dokument použitý pre vloženie údajov dodacieho listu do informačného systému.

Počet testovaných dodacích listov	Počet referenčných riadkov	Úspešne prepísané riadky	Neúspešne prepísané riadky	Celková úspešnosť prepisu [%]
47	835	768	67	91.97605

Tabuľka 7.3: Testovanie úspešnosti prepisu ručne písaného textu – úspešne prepísaný riadok je vtedy, keď registračné číslo a aj množstvo sú prepísané (rozpoznané) správne.

### 7.3 Automatizácia prepisu dodacích listov

Posledné testovanie, a to testovanie priamo v prevádzke Remeslo Stavmat, s.r.o. je zamerané na zistenie urýchlenia prepisu dodacích listov do informačného systému pri použití navrhnutého systému pre jeho automatizáciu oproti v súčasnosti využívanému manuálnemu prepisovaniu. Testovanie prebiehalo u dvoch pracovníkov prevádzky. U oboch pracovníkov testovanie prebiehalo na piatich dodacích listoch.

Pri manuálnom prepisovaní dodacieho listu bolo meranie času spustené od stlačenia prvej klávesy na numerickej klávesnici, to znamená od začiatku prepisovania obsahu dodacieho listu. Meranie času bolo zastavené po uložení dodacieho listu v informačnom systéme.

Pre automatický prepis dodacích listov boli získavané dva časové údaje. Prvý časový údaj zobrazuje rýchlosť spracovania dodacieho listu samotným systémom. Čas sa začal merať od spustenia systému so vstupným obrázkom a meranie času bolo ukončené vtedy, keď systém vytvoril finálny textový súbor vydaj.txt. Proces počas tohoto merania zahŕňa celý už popísaný proces automatického prepisu dodacieho listu do textového súboru. Meranie druhého časového údaje nadväzuje na to prvé. Čas sa začne merať pri vložení obsahu dodacieho listu do informačného systému a ukončuje sa pri uložení dodacieho listu v informačnom systéme. Proces v druhom časovom úseku zahŕňa vloženie obsahu textového dokumentu do informačného systému (postup bol popísaný v 2.1) a kontrolu prepisu dodacieho listu v textovom dokumente, prípadne úpravu alebo doplnenie chýbajúcich dát. Tabuľka 7.4 zobrazuje výsledky tohoto testu. Z tabuľky je vidieť, že pri využití systému pre automatizáciu prepisu dodacích listov je priemerný čas, ktorý musí pracovník venovať prepisu dodacieho listu do informačného systému kratší viac ako o 50%. Tento fakt je umocnený ešte aj tým, že počas spracovania dodacieho listu systémom môže pracovník vykonávať inú činnosť.

		Pracovník 1	Pracovník 2	Systém + Pracovník 1	Systém + Pracovník 2
Dodací list 1	Čas spracovania	-	-	00:21	00:21
	Čas prepisu a kontroly	01:49	02:17	00:53	01:02
Dodací list 2	Čas spracovania	-	-	00:19	00:19
	Čas prepisu a kontroly	02:04	02:21	00:39	00:46
Dodací list 3	Čas spracovania	-	-	00:20	00:20
	Čas prepisu a kontroly	02:48	03:03	00:47	00:53
Dodací list 4	Čas spracovania	-	-	00:19	00:19
	Čas prepisu a kontroly	02:25	02:37	00:43	00:50
Dodací list 5	Čas spracovania	-	-	00:20	00:20
	Čas prepisu a kontroly	02:13	02:30	00:41	00:49
Priemerný čas na dodací list		02:16	02:34	01:04	1:11

Tabuľka 7.4: Testovanie zrýchlenia procesu prepisu dodacieho listu pri využití vytvoreného systému.

# Kapitola 8

## Záver

Cieľom tejto bakalárskej práce bolo navrhnuť a vytvoriť systém, ktorý automaticky prepíše dodacie listy. Výsledný systém na vstupe vyžaduje obrázok vo formáte JPG alebo dokument PDF obsahujúci jeden alebo viac obrázkov dodacích listov. V prípade, že je na vstupe dokument PDF, prevedie sa dokument na samostatné obrázky. Na začiatku sa vstupný obrázok spracuje, nájde sa tabuľka a samostatné bunky. Požadované bunky sa z obrázku vyrežú a predajú sa k rozpoznaniu textu. Po rozpoznaní textu sa vytvorí dvojica **registračné číslo** a **množstvo**. Výsledkom behu systému je textový dokument obsahujúci prepisy požadovaných údajov o konkrétnych položkách z tabuľky dodacieho listu. Tento textový dokument je následne vizuálne skontrolovaný pracovníkom, v prípade chýb v prepise textu sa textový dokument upraví a nakoniec sa využije pre vloženie obsahu dodacieho listu do informačného systému prevádzky Remeslo Stavmat, s.r.o.

Vstupný obrázok musí obsahovať naskenovaný dodací list. V prípade, že skener pri skenovaní papiera dodacieho listu príliš presvieti a vznikne obrázok s nižším kontrastom, je presnosť detekcie čiar tabuľky a samotných buniek značne znížená. Väčšina dodacích listov pre testovanie systému a vytvorenie dátovej sady bola skenovaná priamo v spomenutej prevádzke, aby boli výsledky čo najpresnejšie a systém mohol byť bez ďalších úprav nasadený priamo do tejto prevádzky. Presnosť vyrezávania buniek na testovaných dodacích listoch je 100 %.

Pre rozpoznávanie ručne písaného textu je použitá konvolučná rekurentná neurónová sieť [39] popísaná v 4.3. Pred učením neurónovej siete bola vytvorená dátová sada obsahujúca 8 891 obrázkov – výrezov buniek. Neurónová sieť bola trénovaná na 50 000 epochov, pričom najlepšie výsledky vykazuje pri použití modelu získaného pri 41 000 epochoch, a to viac ako 95 % pri rozpoznaní samostatných znakov, viac ako 92 % pre rozpoznanie sekvencií znakov a skoro 92 % pre rozpoznanie „riadkov“ – dvojice registračné číslo a množstvo.

Testovaním bolo zistené, že pri použití systému môže pracovník ušetriť viac ako 50 % času na každom prepisovanom dodacom liste. Zároveň môže pracovník počas behu systému – spracovania dodacieho listu – vykonávať inú činnosť, čo môže ušetriť ešte viac času. Okrem toho pracovník nemusí do informačného systému údaje z dodacieho listu ručne prepisovať, ale len skontrolovať obsah, ktorý bol vytvorený systémom.

Ďalšie rozšírenie systému bude zahŕňať vytvorenie jednoduchého grafického užívateľského rozhrania, aby bola práca so systémom pohodlnejšia a intuitívnejšia. V grafickom užívateľskom rozhraní bude zobrazený obrázok spracovávaného dodacieho listu a po behu systému aj prepis textu jednotlivých buniek, kde sa prepis skontroluje a prípadne upraví.

# Literatúra

- [1] AVILA, B. a LINS, R. A new algorithm for removing noisy borders from monochromatic documents. In: *Proceedings of the 2004 ACM Symposium on Applied Computing*. Association for Computing Machinery, Marec 2004, s. 1219–1225. DOI: 10.1145/967900.968149.
- [2] BENGIO, Y., SIMARD, P. a FRASCONI, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*. 1994, zv. 5, č. 2, s. 157–166.
- [3] CAMERON, J. P. *A cognitive model for tabular editing*. Technical report OSU-CISRC6/89-TR 26. Columbus, Ohio, USA: Computer and Information Science Research Centre, Ohio State University, jún 1989.
- [4] COÜASNON, B. a LEMAITRE, A. Recognition of Tables and Forms. In: *Handbook of Document Image Processing and Recognition*. 2014. DOI: 10.1007/978-0-85729-859-1.
- [5] DHIRAN, T. a SHARMA, R. Table Detection and Extraction from Image Document. In: *International Journal of Computer & Organization Trends, IJCOT*. Seventh Sense Research Group, August 2013, sv. 3, č. 4, s. 6–9. ISSN 2249-2593.
- [6] DUE TRIER Øivind, JAIN, A. K. a TAXT, T. Feature extraction methods for character recognition-A survey. *Pattern Recognition*. 1996, zv. 29, č. 4, s. 641–662. DOI: 10.1016/0031-3203(95)00118-2. ISSN 0031-3203.
- [7] DUNN, C. E. a WANG, P. S. P. Character segmentation techniques for handwritten text-a survey. In: *11th IAPR International Conference on Pattern Recognition. Vol.II. Conference B: Pattern Recognition Methodology and Systems*. Institute of Electrical and Electronics Engineers (IEEE), 1992, s. 577–580. DOI: 10.1109/ICPR.1992.201844. ISBN 0-8186-2915-0.
- [8] EMBLEY, D. W., HURST, M., LOPRESTI, D. a NAGY, G. Table-processing paradigms: a research survey. *IJDAR International Journal of Document Analysis and Recognition*. Máj 2006, zv. 8, č. 2, s. 66–86. DOI: 10.1007/s10032-006-0017-x. ISSN 1433-2825.
- [9] ESPAÑA-BOQUERA, S., CASTRO-BLEDA, M. J., GORBE-MOYA, J. a ZAMORA-MARTINEZ, F. Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Institute of Electrical and Electronics Engineers (IEEE). 2011, zv. 33, č. 4, s. 767–779. DOI: 10.1109/TPAMI.2010.141. ISSN 1939-3539.

- [10] GATOS, B., DANATSAS, D., PRATIKAKIS, I. a PERANTONIS, S. Automatic Table Detection in Document Images. In: *Pattern Recognition and Data Mining*. August 2005, sv. 3686, s. 609–618. DOI: 10.1007/11551188\_67.
- [11] GATOS, B., PRATIKAKIS, I. a PERANTONIS, S. An Adaptive Binarization Technique for Low Quality Historical Documents. In: *Document Analysis Systems VI*. September 2004, sv. 3163, s. 102–113. DOI: 10.1007/978-3-540-28640-0\_10.
- [12] GATOS, B. G. Imaging Techniques in Document Analysis Processes. In: DOERMANN, D. a TOMBRE, K., ed. *Handbook of Document Image Processing and Recognition*. London: Springer London, 2014, s. 73–131. DOI: 10.1007/978-0-85729-859-1\_4. ISBN 978-0-85729-859-1. Dostupné z: [https://doi.org/10.1007/978-0-85729-859-1\\_4](https://doi.org/10.1007/978-0-85729-859-1_4).
- [13] GRAVES, A., LIWICKI, M., FERNÁNDEZ, S., BERTOLAMI, R., BUNKE, H. et al. A Novel Connectionist System for Unconstrained Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2009, zv. 31, č. 5, s. 855–868. DOI: 10.1109/TPAMI.2008.137. ISSN 1939-3539.
- [14] GRAVES, A., MOHAMED, A. a HINTON, G. Speech recognition with deep recurrent neural networks. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2013, s. 6645–6649.
- [15] GUPTA, A., SRIVASTAVA, M. a MAHANTA, C. Offline handwritten character recognition using neural network. In: *2011 IEEE International Conference on Computer Applications and Industrial Electronics (ICCAIE)*. Institute of Electrical and Electronics Engineers (IEEE), 2011, s. 102–107. DOI: 10.1109/ICCAIE.2011.6162113. ISBN 978-1-4577-2059-8.
- [16] HANMANDLU, M., MOHAN], K. M., CHAKRABORTY, S., GOYAL, S. a CHOUDHURY, D. Unconstrained handwritten character recognition based on fuzzy logic. *Pattern Recognition*. 2003, zv. 36, č. 3, s. 603–623. DOI: 10.1016/S0031-3203(02)00069-9. ISSN 0031-3203.
- [17] HOCHREITER, S. a SCHMIDHUBER, J. Long Short-term Memory. *Neural computation*. December 1997, zv. 9, s. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
- [18] HSIN-CHIA FU, HUNG-YUAN CHANG, YEONG YUH XU a PAO, H. . User adaptive handwriting recognition by self-growing probabilistic decision-based neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2000, zv. 11, č. 6, s. 1373–1384. DOI: 10.1109/72.883451. ISSN 1941-0093.
- [19] IL-SEOK OH, JIN-SEON LEE a SUEN, C. Y. Analysis of class separation and combination of class-dependent features for handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1999, zv. 21, č. 10, s. 1089–1094. DOI: 10.1109/34.799913. ISSN 1939-3539.
- [20] JADERBERG, M., SIMONYAN, K., VEDALDI, A. a ZISSERMAN, A. Synthetic Data and Artificial Neural Networks for Natural Scene Text Recognition. In: *Workshop on Deep Learning, NIPS*. 2014.
- [21] JAIN, A. K., DUIN, R. P. W. a JIANCHANG MAO. Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2000, zv. 22, č. 1, s. 4–37. DOI: 10.1109/34.824819. ISSN 1939-3539.



- [22] KARISHMA PATEL, M. G. Offline Handwritten Character Recognition: A Review. *International Journal of Scientific & Engineering Research*. 2016, zv. 7, č. 5, s. 193–196. ISSN 2229-5518.
- [23] LAM, L., LEE, S. . a SUEN, C. Y. Thinning methodologies-a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Institute of Electrical and Electronics Engineers (IEEE). 1992, zv. 14, č. 9, s. 869–885. DOI: 10.1109/34.161346. ISSN 1939-3539.
- [24] LLADÓS, J. a RUSIÑOL, M. Graphics Recognition Techniques. In: *Handbook of Document Image Processing and Recognition*. London, UK: Springer London, 2014, s. 489–521. DOI: 10.1007/978-0-85729-859-1\_18. ISBN 978-0-85729-859-1.
- [25] MAHASUKHON, P., MOUSAVINEZHAD, H. a SONG, J. Hand-printed English character recognition based on Fuzzy theory. In: *2012 IEEE International Conference on Electro/Information Technology*. 2012, s. 1–4.
- [26] MARINAI, S., GORI, M. a SODA, G. Artificial neural networks for document analysis and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Institute of Electrical and Electronics Engineers (IEEE). 2005, zv. 27, č. 1, s. 23–25. DOI: 10.1109/TPAMI.2005.4. ISSN 1939-3539.
- [27] MEHFUZ, S. a KATIYAR, G. Intelligent Systems for Off-Line Handwritten Character Recognition: A Review. In: *International Journal of Emerging Technology and Advanced Engineering*. 2012, sv. 2, č. 4. ISSN 2250-2459.
- [28] PALUS, H. Color Image Segmentation. In: *Color image processing: methods and applications*. Október 2006, s. 103–128. DOI: 10.1201/9781420009781.ch5.
- [29] PAPAVALASSIOU, V., STAFYLAKIS, T., KATSOUROS, V. a CARAYANNIS, G. Handwritten document image segmentation into text lines and words. *Pattern Recognition*. 2010, zv. 43, č. 1, s. 369–377. DOI: 10.1016/j.patcog.2009.05.007. ISSN 0031-3203.
- [30] PATEL, C., PATEL, A. a PATEL, D. Optical Character Recognition by Open source OCR Tool Tesseract: A Case Study. *International Journal of Computer Applications*. Október 2012, zv. 55, s. 50–56. DOI: 10.5120/8794-2784.
- [31] PERANTONIS, S. J., GATOS, B. a PAPAMARKOS, N. Block decomposition and segmentation for fast Hough transform evaluation. *Pattern Recognition*. 1999, zv. 32, č. 5, s. 811–824. ISSN 0031-3203.
- [32] PETERLIN, P. *Morphological Operations: An Overview* [online]. 1996. Aktualizované 19. 7. 1996 [cit. 4. júna 2020]. Dostupné z: <http://www.inf.u-szeged.hu/ssip/1996/morpho/morphology.html>.
- [33] PLAMONDON, R. a SRIHARI, S. N. Online and off-line handwriting recognition: a comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2000, zv. 22, č. 1, s. 63–84. DOI: 10.1109/34.824821. ISSN 1939-3539.
- [34] PRASAD, D., GADPAL, A., KAPADNI, K., VISAVE, M. a SULTANPURE, K. *CascadeTabNet: An approach for end to end table detection and structure recognition from image-based documents*. Apríl 2020.

- [35] PRASAD, J. R. a KULKARNI, U. V. Trends in Handwriting Recognition. In: *2010 3rd International Conference on Emerging Trends in Engineering and Technology*. Institute of Electrical and Electronics Engineers (IEEE), 2010, s. 491–495. DOI: 10.1109/ICETET.2010.92. ISBN 978-1-4244-8481-2.
- [36] SANJIV KUMAR SINGH, D. Review of Online & Offline Character Recognition. *International Journal of Engineering and Computer Science*. 2015, zv. 4, č. 5, s. 11729–11732. ISSN 2319-7242.
- [37] SERRA, J. Morphological filtering: An overview. *Signal Processing*. 1994, zv. 38, č. 1, s. 3–11. DOI: [https://doi.org/10.1016/0165-1684\(94\)90052-3](https://doi.org/10.1016/0165-1684(94)90052-3). ISSN 0165-1684. Mathematical Morphology and its Applications to Signal Processing. Dostupné z: <http://www.sciencedirect.com/science/article/pii/0165168494900523>.
- [38] SHAFAIT, F., KEYSERS, D. a BREUEL, T. M. Efficient implementation of local adaptive thresholding techniques using integral images. In: *Document Recognition and Retrieval XV*. SPIE, 2008, sv. 6815, s. 317–322. DOI: 10.1117/12.767755.
- [39] SHI, B., BAI, X. a YAO, C. An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2017, zv. 39, č. 11, s. 2298–2304. DOI: 10.1109/TPAMI.2016.2646371.
- [40] SLOVENSKO. Zákon č. 431 z 18. júna 2002 o účtovníctve. In: *Zbierka zákonov Slovenskej republiky*. 2002, čiastka 168, s. 4426–4427.
- [41] SMITH, R. An Overview of the Tesseract OCR Engine. In: *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*. Institute of Electrical and Electronics Engineers (IEEE), 2007, sv. 2, s. 629–633. DOI: 10.1109/ICDAR.2007.4376991. ISBN 978-0-7695-2822-9.
- [42] TRIER, O. D. a JAIN, A. K. Goal-directed evaluation of binarization methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Institute of Electrical and Electronics Engineers (IEEE). 1995, zv. 17, č. 12, s. 1191–1201. DOI: 10.1109/34.476511. ISSN 1939-3539.
- [43] VINCIARELLI, A. Noisy text categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2005, zv. 27, č. 12, s. 1882–1895. DOI: 10.1109/TPAMI.2005.248. ISSN 1939-3539.
- [44] YIN, P.-Y. Skew detection and block classification of printed documents. *Image and Vision Computing*. Máj 2001, zv. 19, č. 8, s. 567–579. DOI: 10.1016/S0262-8856(00)00098-6.

# Príloha A

## Obsah pamäťového média

Priložené pamäťové médium obsahuje všetky zdrojové súbory spolu s dátovou sadou, obrázky dodacích listov pre otestovanie vytvoreného systému, text bakalárskej práce vo formáte PDF a zdrojové súbory pre vysádzanie tohoto textu. Zároveň je k dispozícii video prezentujúce prácu systému, ako aj súbory README.md, requirements.txt a manual.pdf. Adresárová štruktúra priloženého pamäťového média je nasledujúca:

- `delivery_notes/`
  - `images_DN/` ukážkové JPG obrázky dodacích listov
  - `pdf_DN/` ukážkové PDF dokumenty dodacích listov
- `source_codes/` zdrojové kódy systému
  - `CRNN_directory/` zdrojové kódy a ostatné náležitosti k použitej CRNN
  - `DN_recognition/` skripty a ostatné náležitosti pre spracovanie dodacích listov
    - `dataset/` dátová sada na vytvorenie modelu pre CRNN
    - `tests/` pomocné materiály a výsledky testov
- `text/` zdrojové súbory  $\text{\LaTeX}$  pre vysádzanie PDF textu a samotný PDF text práce
- `video/` video prezentujúce prácu systému
- `manual.pdf` dokument obsahujúci návod na obsluhu systému
- `README.md` dokument obsahujúci návod na inštaláciu
- `requirements.txt` dokument s potrebnými balíčkami pre fungovanie systému

## Príloha B

# Inštalácia a návod k použitiu

Systém bol vyvíjaný a testovaný na OS Linux Ubuntu. Pre inštaláciu všetkých závislostí systému je možné využiť príkaz `pip install -r requirements.txt`.

Súbor `requirements.txt` je dostupný v prvej úrovni adresárovej štruktúry systému.

### Požadované parametre a spustenie systému

`-i input_img` – cesta k vstupnému súboru (JPG alebo PDF)

`$ python3 delivery_note_recognition.py -i cesta/k/vstupnému/obrázku`

### Voliteľné parametre

`-t tmp_dir` – cesta k priečinku na uloženie výrezov buniek

`-d tmp_pdf_dir` – cesta k priečinku na uloženie JPG obrázkov z PDF súboru

`-s predict_script_path` – cesta ku skriptu na predikciu textu – `test_shadownet.py`

`-w weights_path` – cesta k súboru s váhami trénovaných tried

`-c char_dict_path` – cesta k priečinku so slovníkom znakov – `char_dict.json`

`-r ord_map_dict_path` – cesta k mape znakov – `ord_map.json`

`-v visualize` – zobrazenie obrázku prechádzajúceho predikciou – *True/False*

`-o output_file` – výstupný súbor obsahujúci prepis textu – `./vydaj.txt`