VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

AUTOMATICKÉ ROZPOZNÁVÁNÍ A ZPRACOVÁNÍ FAKTUR

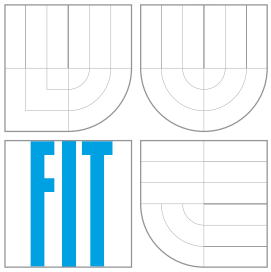
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

VLADIMÍR ŠČEŠŇÁK

BRNO 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

AUTOMATICKÉ ROZPOZNÁVÁNÍ A ZPRACOVÁNÍ FAKTUR

AUTOMATIC INVOICE RECOGNITION AND PROCESSING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — VLADIMÍR ŠČEŠŇÁK
AUTHOR

VEDOUCÍ PRÁCE — Ing. MICHAL ŠPANĚL, Ph.D.
SUPERVISOR

BRNO 2015

Abstrakt

Cílem této práce je navrhnout a implementovat aplikaci pro automatické rozpoznávání a zpracovaní faktur za pomocí využití počítačového vidění. Práce se zabývá analýzou existujících faktur, návrhem a implementaci algoritmů na spravne rozpoznávání výběrem vhodných testovacích vzorkůy a také návrhem a implementací užívatelského rozhraní.

Abstract

This work aims to design and implement application for automatic recognition and processing invoices with assistance using computer vision. The work deals with the analysis of existing invoices, design and implementation of an algorithm for correctly recognition, selection of appropriate test patterns and also design and implementation of the user interface.

Klíčová slova

Automatické, faktura, rozpoznávání, zpracování, optické rozpoznávaní textu, OCR, Tesseract, PDF Parser

Keywords

Automatic, invoice, recognition, processing, optical character recognition, OCR, Tesseract, PDF Parser

Citace

Vladimír Ščešňák: Automatické rozpoznávání a zpracování faktur, bakalářská práce, Brno, FIT VUT v Brně, 2015

Automatické rozpoznávání a zpracování faktur

Prohlášení

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

.....
Vladimír Ščešňák
20. května 2015

Poděkování

Ďakujem vedúcemu práce Ing. Michalovi Španělovi, Ph.D, za poskytnuté konzultácie, vecné návrhy a pripomienky a taktiež za trpezlivosť.

Obsah

Kapitola 1

Úvod

Vo svete plnom papierov a byrokracie nastal čas digitalizácie. Časy keď sa všetko písalo ručne, alebo na písacích strojoch sú dávno preč. Pri práci s počítačmi sa stretávame aj s pojmom optického rozpoznávania textu (OCR). Optické rozpoznávanie textu je proces, pri ktorom z obrázku s výskytom textu vznikne výstup v podobe čistého textu. Táto metóda dala základy aj mojej bakalárskej práci, kde sa venujem automatickému rozpoznávaniu textu na faktúrach a následné spracovanie výstupu.

Táto bakalárska práca popisuje spôsob segmentácie obrázku, teda kroky, vďaka ktorým je možné správne extrahovať text z obrázku, ktorý môže byť preskenovaný, odfotený, alebo vytvorený rôznymi grafickými nástrojmi. Hlavnou náplňou práce je predovšetkým analyzovať jednotlivé faktúry a správne rozoznať, kde sa nachádza dôležitý text, ktorý užívateľovi uľahčí prácu s faktúrami a umožní mu zadávať informácie do systému automaticky. Užívateľovi bude ponúknuté grafické prostredie v podobe webovej aplikácie, kde bude môcť prevádzať úkony ako je rozpoznanie textu na obrázku, ktoré bolo navrhnuté s využitím knižnice Tesseract. Užívateľ môže obrázok pridať pomocou jednoduchej upload stránky, kde si vyberie súbor a jazyk faktúry. Po vykonaní akcie rozpoznávanie je užívateľovi zobrazený výstup, ktorý môže skontrolovať, prípadne upravovať.

V druhej kapitole sa budem venovať operáciam vedúcim k správnemu určeniu oblasti, kde sa v obrázkoch nachádza text. Okrem základných úkonov nad obrázkom predstavím aj metódu založenú na hľadaní ostrých hran v obraze.

V ďalšej kapitole krátko predstavím optické rozpoznávanie znakov (optical character recognizion) a dva princípy, na ktorých to funguje a krátko spomeniem aj modul Tesseract, kde sa dotknem histórie a architektúre a dôvod, prečo som si tento modul vybral. V štvrtej kapitole opíšem už existujúce riešenia, ktoré mi poslúžili aj ako inšpirácia pri implementácii. V piatej kapitole je popísaný návrh riešenia a v šiestej samotná realizácia. Finálne hodnotenie a testovanie a prípadné vylepšenia nájdete v kapitolách sedem a osem.

Kapitola 2

Segmentácia obrazu

Obrázky, ktoré sú predávané na počítačové spracovanie, nie sú dokonalým zachytením reálneho sveta a častokrát sa do obrazu vnášajú šumy, skreslenia a chyby, ktoré mohli vzniknúť napríklad zlým svetlom. Aby boli obrazy správne rozpoznané, je potrebné tieto chyby potlačiť. Na to slúži predspracovanie obrázka, ktoré je dôležitou súčasťou vyhľadávania textu v obrázku a pomocou rôznych techník, ktoré sa aplikujú na každý pixel v obraze, nám pomôžu tieto chyby aspoň čiastočne odstrániť.

2.1 Prahovanie

Prahovanie je najjednoduchšia metóda segmentácie a je to transformácia obrázku do jeho binárnej podoby, najčastejšie reprezentovanej čiernou a bielou farbou. Tento proces sa vykonáva na každom pixeli v obrázku a prepočítava sa jeho hodnota na hodnotu novú podľa použiteho typu prahovania.

Pri metóde jednoduchého prahovania sa predpokladá, že objekty sa v obraze dajú odlíšiť od pozadia na základe hodnoty jasu jednotlivých pixelov. Podľa toho sa zvolí prah, ktorý je potom porovnávaný s jednotlivými pixelmi obrazu. Podľa hodnoty pixelu sa vyhodnotí operácia, ktorá sa má vykonať. Hodnotu pixelu môžme podľa [1] určiť pomocou vzorca 2.1. Kde $g(x, y)$ označuje výslednú hodnotu pixelu, t je hodnota určeného prahu a $f(x, y)$ je zistená hodnota pixelu.

$$g(x,y) = \begin{cases} 255 & \text{if} \quad \text{f(x,y)} > \text{t} \\ 0 & \text{if} \quad \text{f(x,y)} \leqq \text{t} \end{cases} \tag{2.1}$$

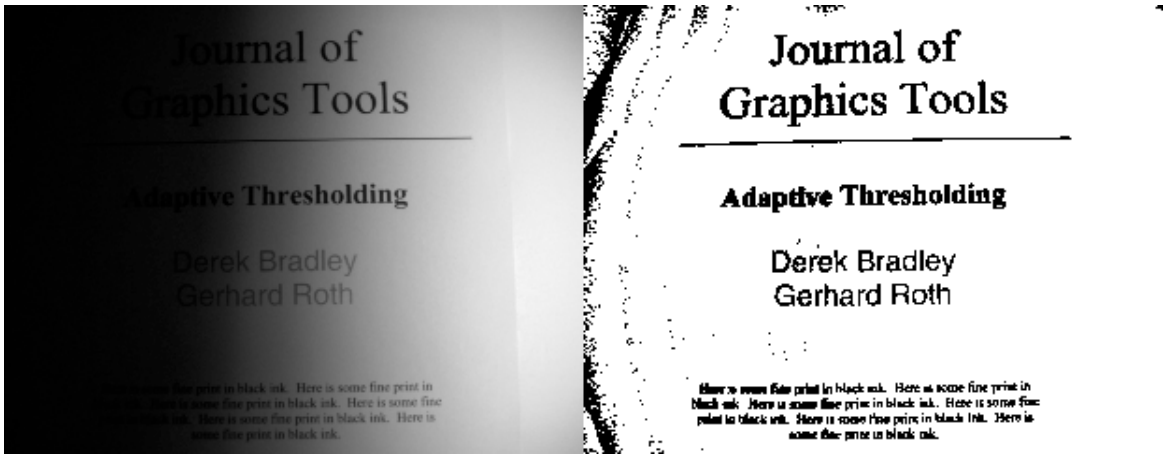
Ďalším typom prahovania je `adaptívne prahovanie`, ktoré je podľa [6] a tiež [2] modifikáciou klasického prahovania. Toto prahovanie sa používa zvyčajne vtedy, ak jednoduché prahovacie metódy zlyhajú a to kvôli tomu, že sa v obrázku nachádzajú objekty, ktoré sú zle osvetlené, alebo vrhajú tieň. Táto skutočnosť vplýva na to, že prah môže byť premenlivý. Pre každý pixel obrázku sa vypočíta priemerná hodnota pixelov v jeho okolí. Výsledné hodnoty môžu byť vyhodnotené pomocou Gaussovej funkcie vzhľadom k vzdialenosti od stredu okolia, alebo sa môže stať, že pixely v okolí budú vyhodnotené rovnakým spôsobom prahovania a s rovnakým prahom.

Úspešnosť prahovania závisí od šírky a hĺbky údolia medzi vrcholmi. Čím viac su vrcholy od seba vzdialené, tým je väčšia šanca na separáciu. Pri zvýšenom obsahu šumu

[1] Z internetovej stránky http://www.codeproject.com/.

Obrázek 2.1: Príklad použitia jednoduchého prahovania [1]

v obraze môže dôjsť k rozširovaniu vrcholou. Úspešnosť tiež závisí od relatívnej veľkosti zdroja nasvietenia a rovnomerných vlastností odrazivosti objektov.



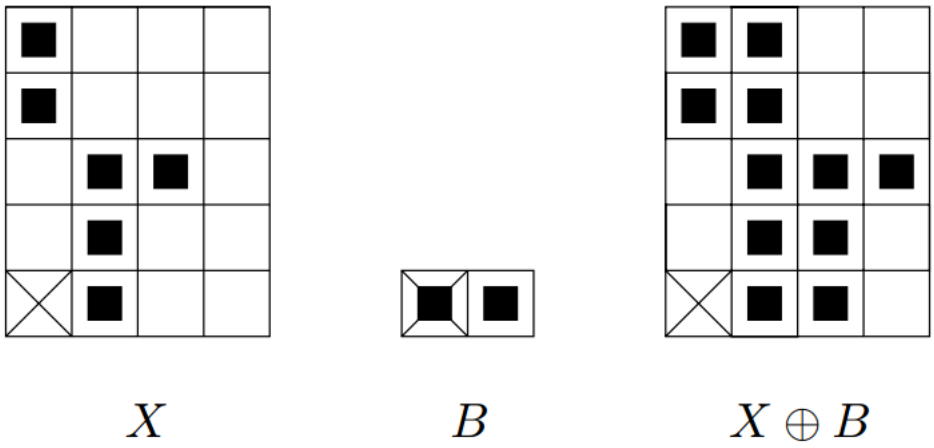
Obrázek 2.2: Príklad použitia adaptívneho prahovania [2]

2.2 Vyčistenie obrazu

Pri práci s obrázkom je možné, že sa do neho zanesie určité množstvo šumu, čo môže byť pre ďalšiu prácu s obrázkom nepríjemné. Túto skutočnosť sa snaží zlepšiť vyčistenie obrazu. Na vyčistenie obrazu sa používajú rôzne metódy. Medzi najjednoduchšie metódy patrí dilatácia a erózia [4] [3], ktoré sa v krátkosti opíšem.

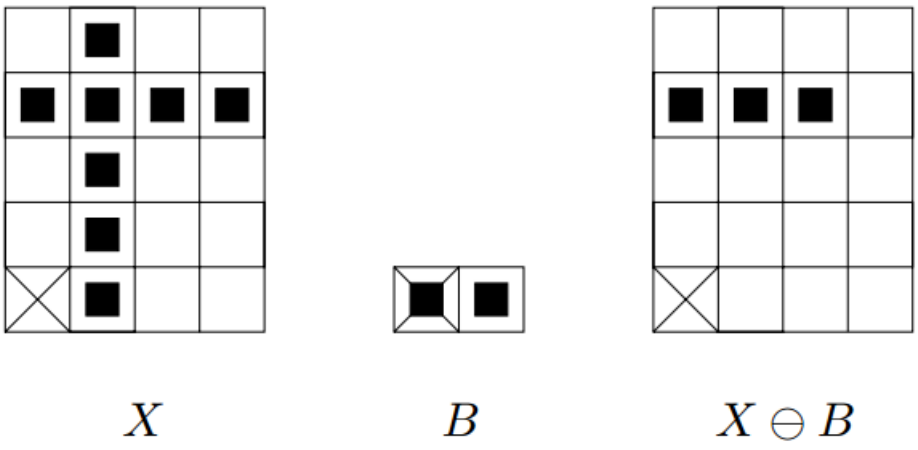
[2] Z internetovej stránky http://blog.inspirit.ru/.

- **Dilatácia** - je jednou zo základných binárnych morfológii, používa sa k zaplneniu malých dier a úzkych zálivov v objektoch. Vstupom je binárny obraz, nad ktorým je vykonaná dilatácia s určitým elementom. Tento spôsob čistenia obrazu je demonštrovaný na obrázku 2.3. Pri použití dilatácie dôjde k zväčšeniu veľkosti objektu, pre zachovanie veľkosti sa dilatácia kombinuje s eróziou.



Obrázek 2.3: Dilatácia obrazu

- **Erózia** - rovnako ako u dilatácie je vstupom binárny obraz. Používa sa k zjednodušeniu štruktúry (rozloženie objektu na jednoduchšie časti). Zaujímavosťou je, že operáciou erózia máme možnosť získať obrysy objektu. Tento spôsob čistenia obrazu je demonštrovaný na obrázku 2.4.



Obrázek 2.4: Erózia obrazu

Existujú ďalšie metódy na odstránenie šumu a to napríklad použitie nízkofrekvenčných filtrov, metóda Gaussovho filtru, metóda anizotropnej difúzie a ďalšie.

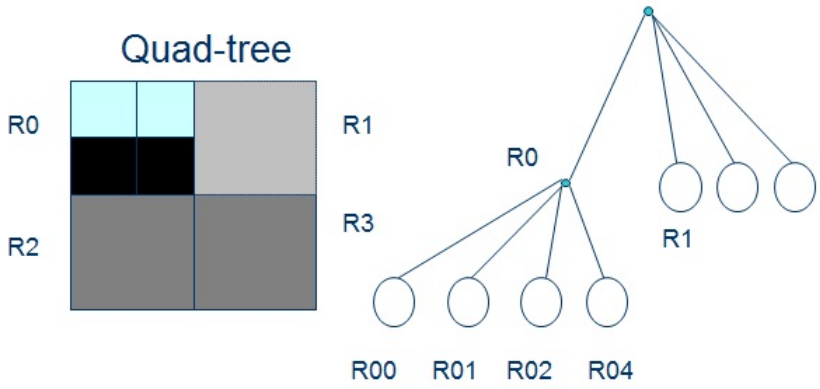
2.3 Metóda spojitých oblasti

Metóda spojitých oblastí (connected component method) je označenie procesu, kedy sa z obrázku skúmajú malé oblasti a na základe predom definovaných pravidiel spájame tieto oblasti do väčších celkov. Tento proces je ukončený v dobe, kedy sa susedné oblasti nedajú spojiť podľa predom definovaných pravidiel.

Ďalej popíšem niektoré z algoritmov pre hľadanie spojitých oblastí.

Štiepenie a spájanie oblastí

Metódou štiepenia a spájania oblastí (split and merge) sa dá dosiahnuť rôznych výsledkov pre rôzne definície počiatočných oblastí, kritérii, počiatky spájania, poradie predpokladaných oblastí, alebo pre rôzne postupy spájania. Táto metóda je kombináciou dvoch algoritmov, rozdelenia pôvodného obrázku na niekoľko oblastí a následné spájanie tychto oblastí, ktoré sa vykonávajú na základe dopredu definovaných pravidiel. Algoritmus reprezentuje obrázok v podobe pyramídy. Jednotlivé oblasti obrázku bývajú prevažne štvorcového charakteru a odpovedajú jednotlivým úrovniam pyramídy. Tento algoritmus je zobrazený na obrázku 2.5.

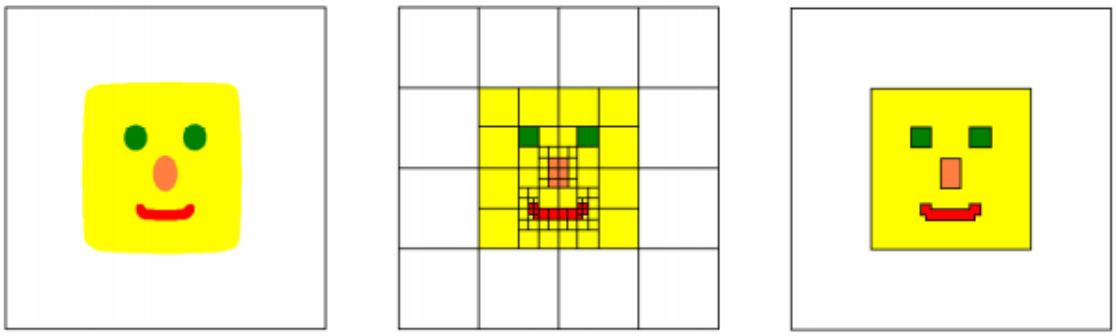


Obrázek 2.5: Split and merge

Dôležitým faktorom, podľa ktorého sa oblasti rozdeľujú a následne zlučujú je kritérium homogenity, ktoré musí byť správne určený. Za toto kritérium sa dá považovať pozícia, intenzita, farba, alebo akákoľvek iná dôležitá informáci o pixeli. Algoritmus ako prvé skúma oblasť, ktorá odpovedá celému obrázku a pokiaľ nie je homogénny rozdelí sa na štyri podoblasti s rovnakými rozmermi. V prípade, že tieto oblasti už existujú v akomkoľvek stupni pyramídy s aspoň približnou hodnotou homogenity, sú zlúčené do jednej oblasti vo vyššej vrstve pyramídy. Táto metóda tiež ukladá informácie o svojich susedoch do grafu. Ako také štiepenie a spájanie vyzerá je demonštrované na obrázku 2.6.

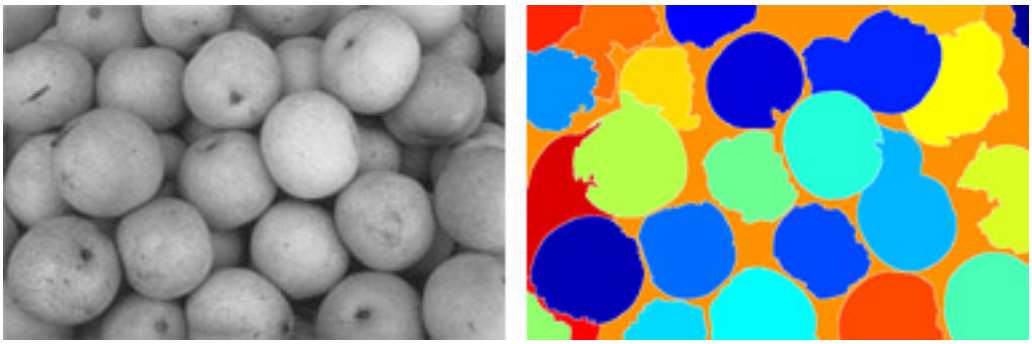
Watershed

Watershed je ďalšou metódou pre segmentáciu obrázku. Pri tejto metóde sa vychádza z geografie. Obraz je chápaný ako terén, alebo topografický reliéf, kde jas vstupného obrázku



Obrázek 2.6: Zľava: originál, štiepenie, spojovanie

určuje výšku terénu, pričom čierna je najnižšou položenou oblasťou a biela najvyššiou. Princíp algoritmu je založený na postupnom zaplavovaní tohoto terénu, alebo aj stúpaním hladiny vody. Výsledný obraz je rozdelený do jednotlivých oblastí, ktoré sú oddelené hrádzami a všetky body danej hladiny sú označené rovnakým unikátnym indexom. Algoritmus tiež dovoľuje aby oblasť pozadia obrázku bola vybraná užívateľom, čo môže segmentáciu značne uľahčiť. Obrázok 2.7 zobrazuje ako metóda pracuje.



Obrázek 2.7: Demonštrácia Watershedingu

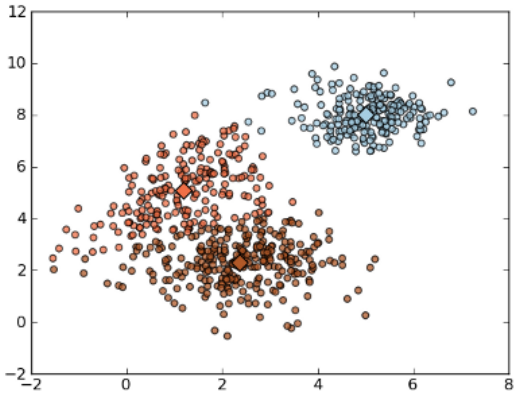
Zhluková analýza (cluster analysis)

Táto metóda sa používa ku klasifikácii objektov a je viacrozmerná štatistická metóda a slúži k triedeniu jednotiek do zhlukov tak, aby jednotky patriace do rovnakej skupiny boli podobnejšie než objekty, ktoré patria do iných skupín. Úlohou tejto metódy je automaticky určiť polohu zhlukov a priradiť jednotlivým vzorkám najbližší zhluk. Najpoužívanejšou metódou je metóda k-means, ktorá je zobrazená na obrázku 2.8.

2.4 Detekcia hrán

Táto metóda je jednou z najstarších segmentačných techník v obraze a dodnes sa využíva ako jedna z popredných metód a je úzko spojená s prahovaním. Hlavným účelom detekcie

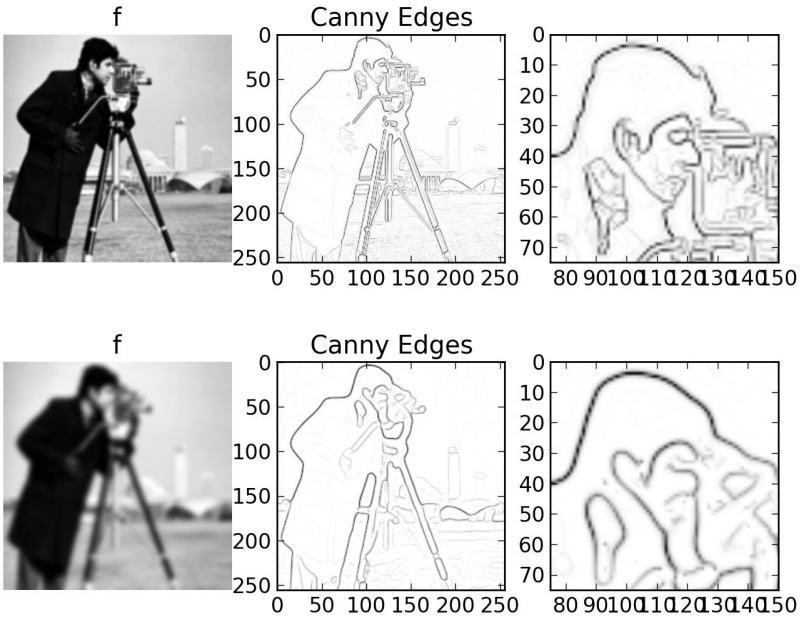
[3]Z internetovej stránky http://mlpy.sourceforge.net/docs/3.5/.



Obrázek 2.8: Metóda k-means [3]

hrán je oddelenie jednotlivých objektov od seba. Pri rozpoznávaní objektov v obraze sú niektoré objekty dôležitejšie ako ostatné. Pomocou hrán je možné určiť, kde sa objekty v obraze nachádzajú, a aj to kde začínajú a končia. V obraze sa hrana reprezentuje ako oblasť, kde sa skokovo mení hodnota jasu. Táto oblasť má špecifickú šírku a úroveň zmeny intenzity. Čím je šírka užšia a zmena jasu väčšia, tým je hrana ostrejšia. Na metódy rozpoznávania hrán sú kladené požiadavky ako presnosť, jednoznačnosť a nízka chybovosť, čím je možné vyhnúť sa detekcii oblastí, ktoré hranami vôbec nie sú, nepresnému určeniu polohy, alebo nechcenému zdvojeniu hrán.

Na detekovanie hrán existuje veľké množstvo detektorov, ktoré vyhľadávajú hrany v dvoch, štyroch, alebo dokonca viacerých smeroch. Najpoužívanejším detektorom hrán je Cannyho detektor.



Obrázek 2.9: Ukážka použitia Cannyho detektoru hrán [4]

Obrázok 2.9 demonštruje fungovanie tohoto detektoru. Jeden z rozdielov medzi Cannyho detektorom hrán a jednoduchšími spôsobmi detekovania hrán, napríklad Laplacov algoritmus, je spočítanie prvej derivácie v x a y a následné spojenie do štyroch smerov derivácie. Lokálne maximá, ktoré body tieto smerové derivácie dosahujú, sa označujú ako kandidáti pre nájdené hrany. Ako uvádza [1], tento detektor bol navrhnutý tak, aby spĺňal tri hlavné požiadavky:

- **detekcia** - je kritériom pre nájdenie významných hrán, ktoré nesmú byť v obrázku prehliadnuté a tiež je potrebné sa vyhnúť falošným znakom, ktoré netvoria hranu
- **lokalizácia** - tá hovorí o tom, že medzi aktuálnou a lokalizovanou hranou by mala byť minimálna vzdialenosť
- **jednoznačnosť** - rozpoznávanie hrán tak, aby nedochádzalo k zdvojeniu, a tiež aby sa vysporiadal s problémom detekcie zlej hrany spôsobenej šumom a nehladkými operátormi hrany.

Fungovanie algoritmu pre nájdenie hrán pomocou Cannyho detektoru je popísaný v algoritme:.

1. Odstráň šum z obrazu f pomocou Gaussoveho filtru σ.
2. Ohodnoť smer lokálnej hrany n použitím rovnice 2.2 pre každý pixel v obraze, ktorá slúži k zvýrazneniu hrán pomocou Robertsovho gradientu, ktorý prepočíta hodnoty $DH(i,j)$ z pôvodných hodnôt pixelu $f(i,j)$.

$$DH(i,j) = [f(i,j) - f(i+1,j+1)] + [f(i+1,j) - f(i,j+1)] \tag{2.2}$$

3. Lokalizuj hranu pomocou rovnice 2.3, ktorý odpovedá vzorcu pre zhlukovú analýzu. Na začiatku si určíme množinu atribútov, podľa ktorých budeme data z obrázku segmentovať do tzv. zhlukov. Majme množinu objektov $O = \{O_1, \dots O_n\}$ a mieru vzdialenosti V, potom sa zhluk uvádza ako podmnožina $X \in Y$.

$$max(V(O_i, O_j)) < min(V(O_k, O_i)), \qquad O_i, O_j \in X, O_k \notin X \tag{2.3}$$

4. Vypočítaj veľkosť hrany pomocou rovnice 2.4, ktorá spočíta kolmý smer k hrane n na základe gradientu Δ, Gaussovenj funkcie G a obrázku f.

$$n = \frac{\Delta(G * f)}{|\Delta(G * f)|} \tag{2.4}$$

5. Využi prahovanie na hrany v obraze s hysteréziou za účelom eleminácie niekoľkonásobného označenia hrán.
6. Kroky 1 až 5 opakuj pre vzrast hodnôt smerodatnej odchýlky σ.
7. Vyhodnoť výsledné informácie o hranách vo viacerých ohľadoch na základe spoločných rysov.

[4] Z internetovej stránky https://staff.fnwi.uva.nl/.

Ako nájsť lokálne maximu v smere kolmému k hrane, kde δ predstavuje parciálnu deriváciu funkcie G k n nám popisuje vzorec 2.5.

$$\frac{\delta^2}{\delta n^2} G * f = 0 \tag{2.5}$$

Pre úplnosť treba ešte vedieť rovnicu pre výpočet veľkosti hrany 2.6.

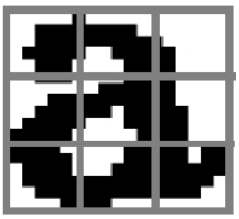
$$|G_n * f| = |\Delta(G * f)| \tag{2.6}$$

Kapitola 3

Rozpoznávanie textu

Optické rozpoznávanie znakov (ďalej OCR) je komplexná technológia, ktorá dokáže konvertovať obrázky s textom do editovateľných formátov, teda najčastejšie do podoby textu. OCR umožňuje spracovávať odfotografované, alebo v lepšom prípade preskenované knihy, snímky, fotografie s textom, v tomto prípade faktúry a rozpoznať v nich text. Táto technológia sa používa v mnohých oblastiach a najnovšie a najmodernejšie systémy OCR dokážu rozpoznať už aj komplikované typy súborov ako sú naskenované strany časopisov, alebo dokumenty odfotené mobilným zariadením. Tiež dokážu pracovať so širokým spektrom formátov ako napríklad `BMP`, `JPEG`, `PNG`, `GIF`, či `TIFF`. Ako uvádza [5] OCR pre klasifikáciu vzorov používa tieto metódy:

- **Rozdelenie do pásiem** - políčko, kde je lokalizovaný znak je rozdelené na niekoľko oblastí a skúma sa histogram tmavých miest v jednotlivých oblastiach. Pre správne rozpoznanie sa potom tieto histogramy porovnávajú s rysmi jednotlivých znakov, ktoré vzídu z tzv. tréningových dát. Toto delenie je demonštrované na obrázku 3.1.



Obrázek 3.1: Metóda rozdelenia do pásiem

- **Priesečníky** - metóda je založena na počte priesečníkov vektorov, ktoré sú predom zvolené v políčku, kde sa nachádza znak. Metóda rozpoznávajúca na základe špecifických rysov sa nazýva štrukturálnou analýzou. V tejto analýze sú jednotlivé znaky popisované geometrickou a topologickou štruktúrou znakov. Táto metóda je však ešte predmetom výskumu. Metóda je demonštrovaná na obrázku 3.2.

Aby bol text správne identifikovaný, je potrebné si určit, v akej podobe sa text v obrázku bude zobrazovať. Táto práca je ale zameraná na faktúry, čo je účtovný doklad zachytávajúci určitý výmenný vzťah, napríklad predaj tovaru dvoch právnych subjektov (napríklad medzi dvoma právnymi osobami, medzi právnickou osobou a fyzickou osobou - podnikateľom a pod.). Tento doklad býva vo väčšine tlačený na bielom podklade. Môže, ale obsahovať nejaké

Obrázek 3.2: Metóda priesečníkov

pozadie reprezentujúce farby firmy, či samotné logo firmy. Na faktúre sa zvykne používať čierne písmo, no nie je to nutnosťou a stretneme sa aj s inými použitými farbami. Čo sa týka použitých fontov, pri faktúrach sa stretávame so základnými fontmi, ktoré sa používajú v knihách, novinách a rôznych reklmaných predmetoch. Kedže ide skôr o oficíalny dokument nezvyknú sa používať písane, ozdobné, zvláštne, futuristické a znakové [1]. Táto skutočnosť dovoľuje konštatovať, že pri rozpoznávani jednotlivých písmen by nemalo dôjsť k zámene znakov, a teda že bude text prečítaný správne.

Arial
Arial Black
Calibri
Candara
Century Gothic
Corbel
Euphemia
Franklin Gothic Medium
Tahoma
Verdana

Obrázek 3.3: Najčastejšie použité fonty na faktúrach

Na samotné rozoznávanie textu z obrázkov sa dnes používajú už pokročilé systémy OCR. Zo širokej ponuky som si vybral tesseract OCR.

3.1 Modul Tesseract

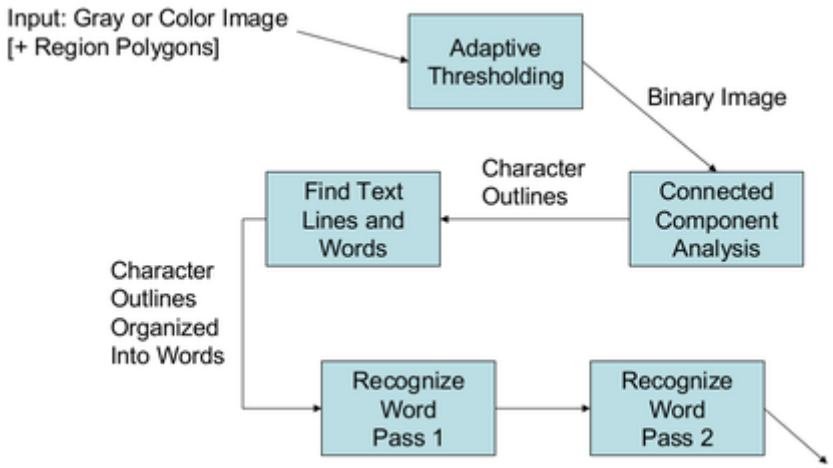
Ako uvádza [7] Tesseract je open-source knižnica na optické rozpoznávanie znakov, ktorý bol vytvorený spoločnosťou HP medzi rokmi 1984 a 1994. Každoročné testy UNLV presnosti OCR sa ukazovali vo výbornom svetle, no nakoniec sa nad týmto OCR roztiahol plášť mlčanlivosti. Tesseract vznikol ako dizertačná práca [8] v laboratóriach HP v Bristole a jeho výskum sprevádzalo nadšenie z vývinu plochých skenerov spoločnosti HP. Motivácia bola podnietetá aj skutočnosťou, že ostatné komerčné OCR engine boli ešte v plienkach. Po tom čo sa skĺbili projekty v HP Labs v Bristole a HP divíziou skenerov, ktorá sa nachádzala v Colorade, Tesseract mal výrazný náskok v presnosti rozpoznávania pred ostatnými komerčnými produktmi, no nikdy sa sám produktom nestal.

Ďalšia fáza vývoja prebiehala v laboratóriach HP v Bristole ako výskum OCR kompresie. Vývoj sa sústreďoval na zlepšenie schopnosti odmietnutia viac než na presnosť základnej

[1] Viz http://tvorim.net/typografia/70-pisma-delenie-fontov-a-ich-primerane-pouzitie.

úrovne. Pri konci fázy koncom roka 1994, vývoj prestal úplne. Engine bol v roku 1995 poslaný na každoročný UNLV test presnosti OCR, kde sa osvedčil v konkurencii komerčných enginov. Koncom roku 2005 spoločnosť HP spravila Tesseract voľne dostupným širokej verejnosti.

Tesseract na vstupe predpokladá binárny obraz s vymedzenými textovými regiónmi. Jeho architektúru môžme vidieť na obrázku 3.4.



Obrázek 3.4: Architektúra Tesseract OCR

Ide o multiplatformové riešenie, ktoré je možne využiť na operačných systémoch ako Linux, Mac OSX, Windows a aj Android, kedže sa jedná o OS, ktorý je postavený na jadre linuxu. Tesseract sa dá používať cez terminál, rôzne aplikácie, alebo ako v tomto prípade cez knižnicu. Knižnica bola zvolená z dôvodu voľnej dostupnosti, podpory UTF-8 znakov a rôznych jazykov, a to konkrétne slovenčiny a češtiny. Dosahuje uspokojivé výsledky a tiež je veľmi populárna, vďaka čomu je rozšírená a dostupná pri práci s jazykmi ako je PHP. Pri tejto aplikácii bola však zvolená knižnica napísaná v jazyku C/C++.

Kapitola 4

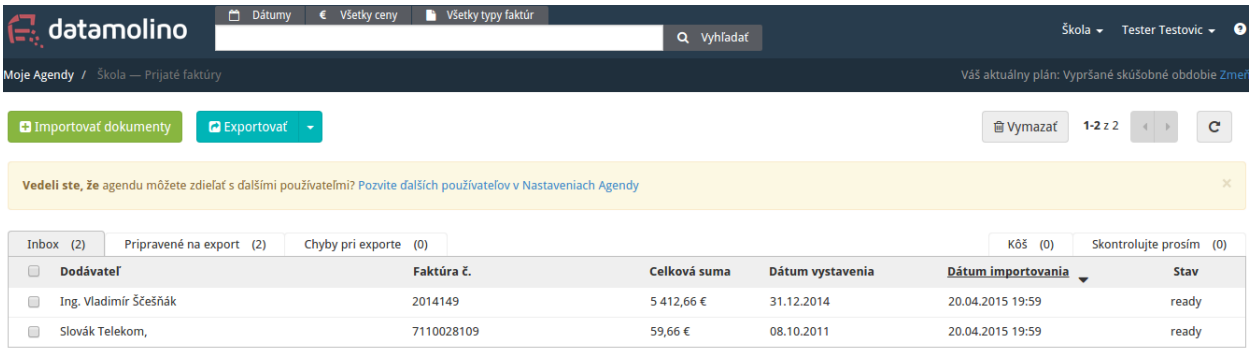
Existujúce riešenia

Na trhu existuje veľké množstvo produktov, ktoré ponúkajú podobnú funkcionalitu prípadne ju ešte rozširujú. Tieto produkty sa zameriavajú na účtovníkov, podnikateľov, ale aj veľké korporácie. Líšia sa však cenou, kde sú niektoré základné funkcie zadarmo, alebo sú sprístupnené len na určitú skúšobnú dobu. Tieto riešenia ponúkajú užívateľom príjemné užívateľské rozhranie pre správu faktúr a bločkov, ich korektné rozpoznanie a export do rôznych účtovných softvérov. V tejto kapitole spomeniem pár riešení, na ktoré som natrafil pri skúmaní už existujúcich riešení.

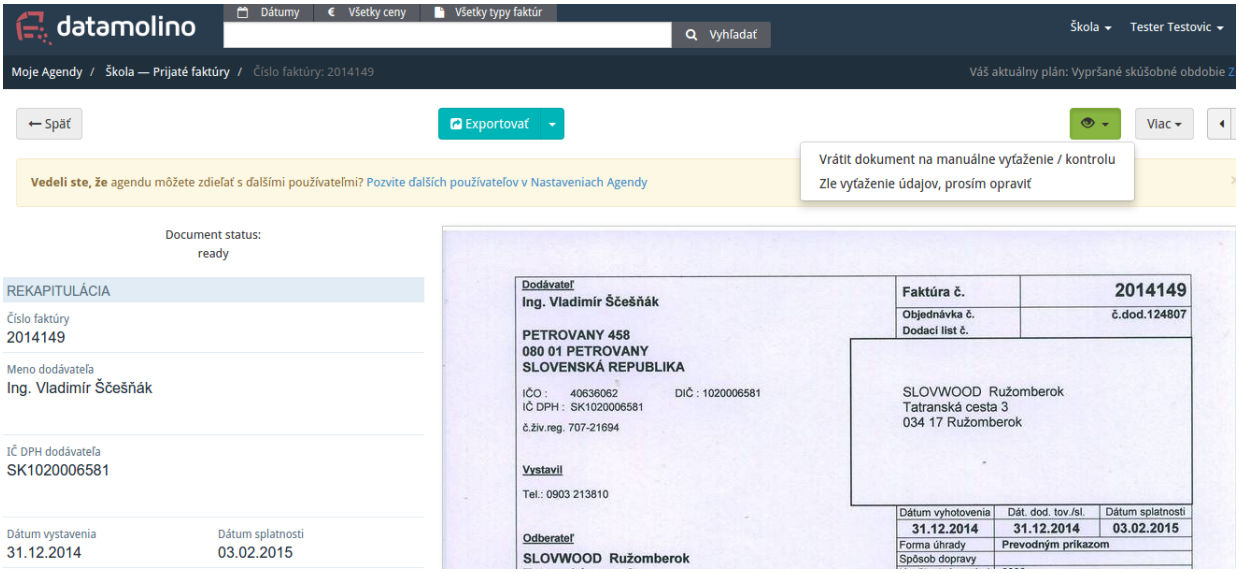
4.1 Datamolimo

Datamolimo je slovenské riešenie zamerané na efektívnejšiu prácu s faktúrami a účtnovnými dokladmi. Ich cieľom sú predovšetkým malé a stredné firmy. Toto riešenie ponúka registráciu zadarmo, kde sú poskytnuté základné funkcionality aplikácie, avšak len na skúšobnú dobu. Po prihlásení užívateľa privíta príjemné užívateľské rozhranie, obrázok 4.1, v ktorom sa užívateľ dokáže rýchlo zorientovať. Užívateľ si môže jednoducho pridať agendu s množstvom faktúr za mesiac. Po pridaní agendy sa užívateľ môže presunúť do danej agendy, kde už môže využívať jednotlivé funkcie aplikácie. V agende sa zobrazuje aktuálny počet dokumentov, kde sú zobrazené základné údaje o dokumente. Ďalej môže užívateľ exportovať faktúry do všeobecných formátov ako sú CSV a Excel, alebo môže využiť export do konkrétneho účtovného programu, ktorý je v ponuke.

Po pridaní nového dokumentu môže užívateľ sledovať stav spracovania svojej faktúry. Ten sa mení v závislosti od faktúry. Ak sa jedna o klasickú faktúru, výsledok je spracovaný behom 5 minút a stav sa zmení na ready. V prípade, že ide o komplikovanejšiu faktúru, ktorá obsahuje nejaké obrázky poprípade logo firmy, alebo sú údaje v inom ako klasickom poradí, výsledok sa spracuje do 24 hodín. Po spracovaní faktúry je užívateľovi zobrazena rekapitulácia, ktorá obsahuje základné údaje, ktoré boli získané z obrázka a vedľa nahrávaný obrázok. V prípade, že dôjde k nesprávnemu rozpoznaniu užívateľ môže faktúru odoslať na opätovné rozpoznanie, alebo na manuálne rozpoznanie.



Obrázek 4.1: Uživateľské rozhranie Datamolimo



Obrázek 4.2: Užívateľské rozhranie Datamolimo - spracovanie výsledkov

4.2 Smartsoft Invoices

Ide o zahraničné riešenie, ktorého cieľom sú malé, stredné ale predovšetkým veľké korporácie, čo dokazuje aj ich zoznam klientov !!footnote!!. Na rodiel od Datamolimo 4.1, nejde o webovú aplikáciu ale desktopovú. Smartsoft Invoices automatizuje proces skenovania faktúry, extrakciu a klasifikáciu dát. Rozpoznáva údaje zo skenovaných papierových, alebo PDF faktúr. Získané údaje prechádzajú validáciou užívateľa a všetky získané data sú následné triedené, pre zachovanie prehľadnosti o vložených faktúrach. Tiež umožňuje export faktúr do všeobecných formátov. Aplikácia poskytuje základné funkcie na skúšobnú dobu, pre úplnú funkcionalitu je potrebné si za túto aplikáciu zaplatiť. Viac o tejto aplikáci môže napovedať toto video https://www.youtube.com/watch?v=A8P1GLE2FT8.

Kapitola 5

Návrh riešenia pre automatické rozpoznávanie a spracovanie faktúr

Na rozdiel od ostatných aplikácii pre správu faktúr, úlohou tejto aplikácie bude automatické spracovávanie faktúr, bez čo možno najmenšej nutnosti zadávať potrebné informácie do systému manuálne. Aplikácia dokáže spracovávať faktúry a to rôznych typov písaných strojovým písmom a to vo formáte `.JPEG`, `.PNG`, `.GIF`, ale aj `.PDF`. Kedže existujú rôzne šablóny faktúr, aplikácia by mala byť schopná reagovať na väčšinu použitých šablón. Program by mal taktiež podporovať importovanie faktúr s rôznymi jazykovými mutáciami.

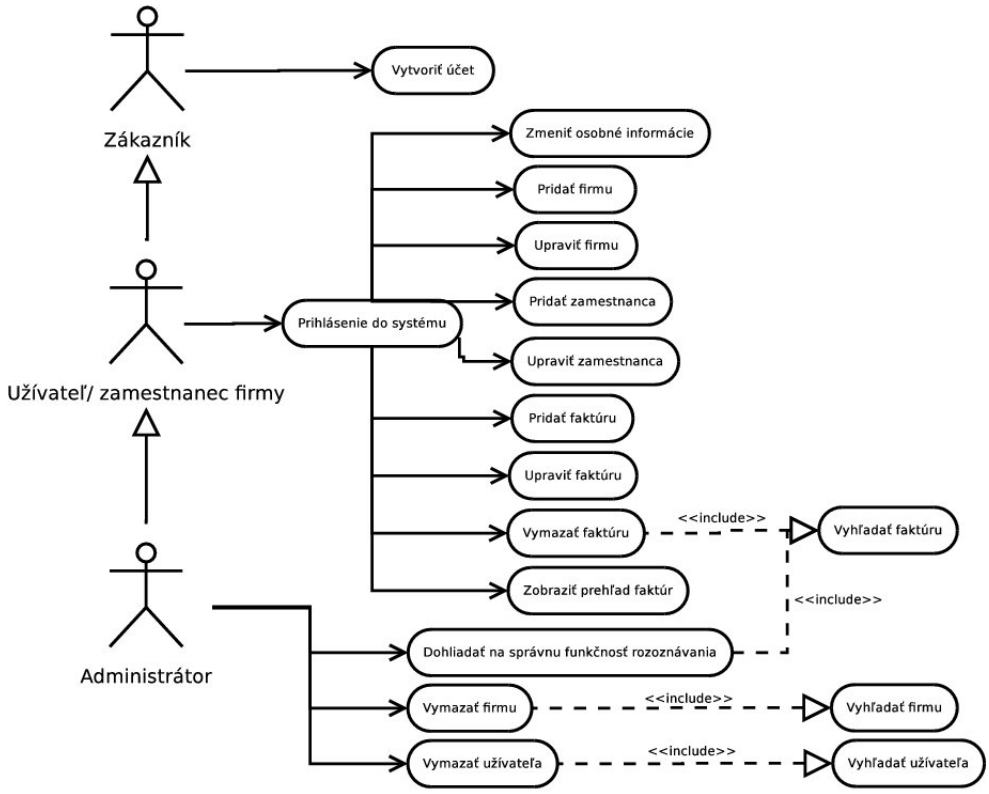
5.1 Špecifikácia

Pri preskúmaní existujúcich technológii som sa rozhodol pre aplikáciu pomocou webového rozhrania, ktoré ponúka jednoduché užívateľské rozranie. Aplikácia bude užívateľom ponúkať systém pre správu faktúr. Pod pojmom správa faktúr sa chápe vloženie novej faktúry, editácia faktúry, zmazanie faktúry a taktiež export faktúr vo formáte XML. Aplikácia na vstupe očakáva obrázok, ktorý bude ďalej spracovaný a budú sa vyhľadávať dôležité informácie. Medzi tieto informácie patrí:

- Meno dodávateľa
- IČO dodávateľa
- Adresa dodávateľa
- Celková suma faktúry

Užívateľovi bude ponúknuta možnosť výberu výsledkov, ktoré aplikácia vyhodnotila ako pravdepodobné výsledky. Vo výbere sa výsledky zobrazia vzostupne od najpravedpodobnejšieho výsledku po najnepravdepodobnejší výsledok. V prípade, že program nevyhodnotí údaje, užívateľovi bude ponúknuta možnosť editácie daných polí. Užívateľovi bude k dispozícii možnosť zobraziť si kalendár, kde bude môcť zistiť dátum a čas vloženia faktúry respektíve úpravu faktúry. V prípade, že budú viacerí užívatelia spravovať rovnakú firmu, zobrazí sa aj meno užívateľa, ktorý faktúru nahral. Taktiež bude užívateľovi zobrazený celkový prehľad, ktorý bude okrem zoznamu faktúr zobrazovať aj celkovú sumu faktúr. Prehľad môže byť zobrazený textovou formou, alebo v grafickej podobe a to napríklad grafom.

V tomto systéme sú zahrnuté tri role užívateľov a to administrátor, užívateľ/zamestnanec a zákazník, ktorý si systém len prezerá. Ako je možné vidieť na obrázku 5.1 najnižšiu rolu v systéme predstavuje zákazník, ktorý si celý systém prezerá a môže vykonať len akciu registrácie, po ktorej bude môcť s aplikáciou pracovať plnohodnotne. Ďalšou rolou, rola užívateľa/zamestnanca, ktorým sa zákazník stane po registrácii, má možnosť prihlásiť sa do systému, kde mu budú poskytnuté ďalšie možnosti pre prácu so systémom.
Užívateľ môže zmeniť informácie poskytnuté pri registrácii, pridať firmu, pre ktorú sa budú faktúry spracovávať, alebo firmu upraviť. Ďalej môže pridať zamestnanca, ktorý bude pre firmu pracovať poprípade upraviť existujúceho zamestnanca. Hlavnou výhodou registrovaného zákazníka je pridanie faktúry na automatické spracovanie a rozpoznávanie faktúry, alebo úpravu už existujúcej faktúry. Užívateľovi bude ponúknuta akcia na zmazanie faktúry v prípade, že bola do systému nahraná zlá faktúra a taktiež akcia pre vyhľadávanie faktúry. Celkový prehľad vložených faktúr a celková suma budú zobrazované po prihlásení do systému.
Najviac kompetencií má však administrátor, ktorý môže okrem už hore spomenutých udalostí vyhľadať firmu respektíve zamestnanca a zmazať ich. Administrátor dohliada aj na správnu funkčnosť rozoznávania.



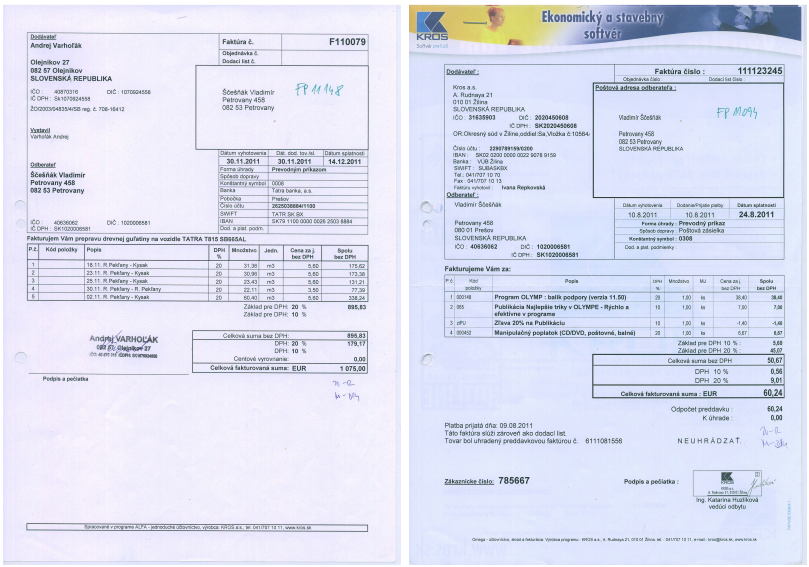
Obrázek 5.1: Use-case diagram

5.2 Analýza faktúry

Vhľadom na to, že som sa rozhodol pre webovú aplikáciu, celková analýza faktúry bude prebiehať na strane servera. Užívateľovi bude poskytnutá možnosť nahrať súbor a to buď vo formáte `.JPEG`, `.PNG`, `.GIF`, alebo vo formáte `.PDF`. Tieto súbory budú ukladané na server do špeciálneho adresára, ktorý bude mať obmedzené prístupové práva, aby sa predišlo úniku citlivých informácii. Po vykonaní hlavnej funckie aplikácie bude užívateľ môcť získané informácie uložiť. Tieto informácie sa budú uschovávať do databázy, konkrétne do tabuľkiek `company` a `invoice`.

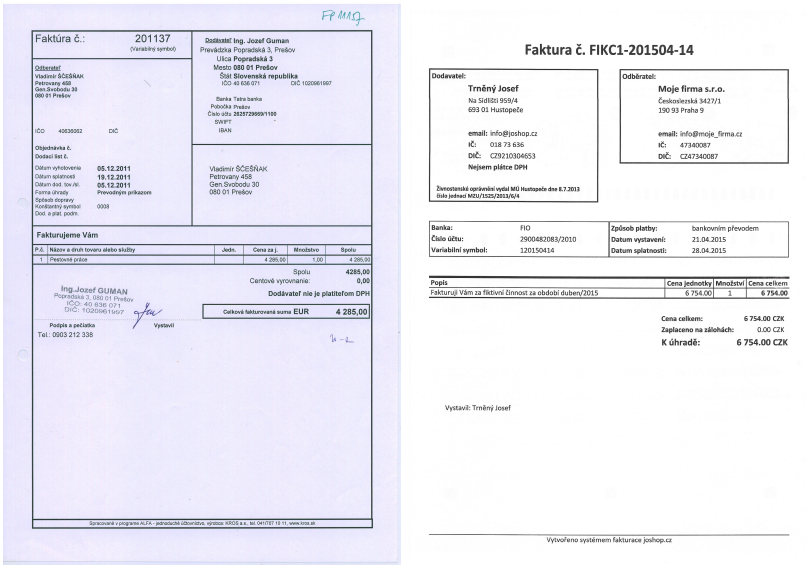
Pri jej štúdiu som narazil na rôzne typy faktúr, ktoré by mala aplikácia dokázať rozpoznať a správne spracovať údaje, ktoré sa na nej nachádzajú. Pri analýze som si tieto faktúry rozdelil do pár skupín.

- **Prvú skupinu faktúr** si pre ľahší popis rozdelím na štyri kvadranty. Prvý kvadrant obsahuje údaje ako je meno dodávateľa, adresa dodávateľa a IČO. V tom istom kvadrante môžme nájsť aj meno odberateľa, adresu odberateľa a aj IČO odberateľa. V druhom kvadrante sú obsiahnuté kontaktné údaje na odberateľa, ako je bankové spojenie, dátum splatnosti faktúry. Tretí kvadrant obsahuje čiastočné informácie o položkách. Štvrtý kvadrant obsahuje celkovú čiastku a menu a to aj s DPH a bez DPH. Druhý a tretí kvadrant, však nie je pre implementáciu tejto aplikácie dôležitý a pri tejto skupine faktúr sa zameriam hlavne na prvý a štvrtý kvadrant.



Obrázek 5.2: Prvá skupina faktúr

- **Druhú skupinu faktúr**, ktorú môžeme vidieť na obrázku 5.3 môžeme taktiež pre ľahší popis rozdeliť na štyri časti. Tentokrát sa ale v prvom a druhom kvadrante môžu vyskytovať už vyššie spomenuté údaje o dodávateľovi, alebo o odberateľovi. Ďalšie dve časti sú podobné prvej skupine faktúr. Pri tejto skupine budem pracovať s prvým, druhým ale aj so štvrtým kvadrantom.



Obrázek 5.3: Druhá skupina faktúr

- **Tretia skupina faktúr** je viac odlišná od prvých dvoch skupín. Ako je možné vidieť na obrázku 5.4 tento typ faktúry je špecifický tým, že meno dodávateľa ako aj jeho adresa a IČO sa nachádzajú bezprostredne za sebou vo vodorovnom smere, kdež to v prvých dvoch prípadoch sa tieto údaje nachádzajú za sebou v horizontálnom smere.

Obrázek 5.4: Tretia skupina faktúr

Existuje však množstvo rôznych šablón na faktúry, ale tieto tri skupiny sú dôležité pre implementáciu a správne fungovanie aplikácie. Od týchto typov šablón môžem zovšeobecniť

moje riešenie aplikácie, nakoľko od týchto skupín sú odvodené aj iné šablóny faktúr, ktoré by mala aplikácia zvládnuť rozpoznať.

5.3 Návrh spracovania faktúry

Pre rozpoznávanie textu na faktúre som použil modul tesseract 3.1. Tesseract očakáva na vstupe obrázok, ktorý spracuje a na výstupe sa objavia výsledky v podobe textu a je možné zobraziť aj pozíciu textu a percento úspešnosti čítania. Pri testovaní tohoto modulu som však prišiel na skutočnosť, že občas sa niektoré znaky prečítaju nesprávne a to viedlo k tomu, že som si obrázok rozdelil na menšie časti. Toto rozhodnutie viedlo k zvýšeniu presnosti čítania znakov. Ako už bolo spomenuté vyššie, spracovanie faktúry bude prebiehať na strane servera. Aplikácia, ktorá beží na serveri rozdelí obrázok čo možno v najlepšej miere, aby nedochádzalo k strate dôležitých údajov, podľa typu šablóny faktúry. Ďalej sú jednotlivé časti obrázkov spracovávané modulom tesseract.

V každej časti sa tento modul snaží nájsť výskyt kľúčových slov. Ako prvé je potrebné nájsť slovo **IČO**, od ktorého môžme vo vyhľadávaní posunúť ďalej. Ak sa toto slovo na danej časti obrázku vyskytuje, uloží sa jeho pozícia a dané slovo. Od pozície tohto slova, začne vyhľadávanie ďalších slov, konkrétne **Dodávateľ** a **Odberateľ**. V prípade, že je nájdený výskyt prvého slova, uložíme si jeho pozíciu a slovo a je možné s určitosťou povedať, že sa jedná o IČO dodávateľa. V opačnom prípade je možné prehlásiť, že sa jedná o IČO odberateľa, ktorého výskyt však pre túto aplikáciu nie je až tak prioritný.

Ako ďalšie je spustené vyhľadávanie **konkrétneho čísla**, ktoré následuje po slove IČO. Zpravidla býva toto číslo osem miestne. Pri výskyte tohoto čísla sa uloží pozícia tohoto čísla a číslo samotné, tiež sa ukladá váha čísla. Váha čísla v tomto prípade znamená, či sa v okolí tohoto čísla vyskystuje slovo Dodávateľ, alebo Odberateľ. Ak sa v okolí vyskytuje slovo Dodávateľ je váha čísla väčšia, ako v prípade, že sa v okolí nachádza slovo Odberateľ. Je možné, že aplikácia nájde číslo, ktoré vyhovuje vyššie uvedeným podmienkam, avšak bez výskytu slov dodávateľ a odberateľ. V tomto prípade sa uloží pozícia čísla, číslo samotné a váha, ktorá sa rovná trom. V prípade, že sa IČO už nachádza v databáze a obsahuje už všetky údaje, vyhľadávanie sa ukončí a nižšie popísané údaje budú doplnené z databázy. Týmto spôsobom sa dá celkové vyhľadávanie urýchliť, avšak len v tom prípade, že v databáze sa už budú nachádzať nejaké údaje.

Aplikácia ďalej pokračuje vyhľadávanim **Mena dodávateľa** a to tak, že aplikácia berie v úvahu pozíciu slova dodávateľ, od ktorej prebieha vyhľadávanie v okolí tohto slova. Aplikácia berie do úvahy, že sa v mene dodávateľa môže vyskytovať názov spoločnosti, ako aj či ide o spoločnosť s ručením obmedzeným, alebo o akciovú spoločnosť. Tiež je schopná reagovať, ak na faktúre vystupuje ako dodávateľ živnostník vystupujúci pod svojím menom a to tak, že sa v mene môže vyskytovať titul(pred a aj za menom), meno a priezvisko, poprípade viac priezvísk. Uložená je opäť pozícia mena a tiež celé meno dodávateľa a váha, ktorá tentoraz reprezentuje vzdialenosť od výskytu slova dodávateľ.

Ďalej aplikácia pokračuje hľadaním adresy. Na faktúrach býva vo väčšine prípadov pod menom dodávateľa **ulica**. Ulica sa vo väčšine prípadov uvádza vo formáte: názov ulice a číslo bloku a tomuto formátu je prispôsobená aj aplikácia. Program si zapamätá pozíciu slova, slovo a tiež váhu, ktorá aj v tomto prípade znamená vzdialenosť, no teraz od výskytu mena dodávateľa. Za ulicou sa uvádza **poštové smerovacie číslo** (ďalej PSČ) mesta a taktiež názov mesta. PSČ sa používa predovšetkým pre identifikáciu mesta a na území Slovenskej a Českej republiky to býva päťciferné číslo, členené na trojčíslie a dvojčíslie. Aplikácia využíva túto skutočnosť pre správnu identifikáciu PSČ a vyhľadáva sa päťciferné

číslo, avšak môže byť členené aj iným spôsobom ako je členenie na trojčísle a dvojčíslie. Aj v tomto prípade sa ukladá pozícia, číslo a váha. Bezprostredne za PSČ sa uvádza mesto. Aplikácia teda vyhľadáva najbližšie slovo po výskyte PSČ a uloží pozíciu a slovo.

Pre identifikáciu dodávateľa sa ako posledné vyhľadáva **krajina**. Tá sa však na faktúre nemusí uvádzať, nakoľko niekedy môže byť jasné o akú krajinu ide. V prípade, že však uvedená je, algoritmus vyhľadávania pokračuje od poslednej zistenej pozície PSČ a mesta a ukladá slovo, ktoré sa vyskytuje za touto pozíciou a to aj jeho pozíciu a váhu.

Informácie o dodávateľovi sú teda spracované. Posledné vyhľadávanie sa vykoná pre nájdenie **celkovej ceny** faktúry. Pri analýze šablón je možné určiť v ktorej časti sa celková čiastka nachádza, čo nám uľahčuje vyhľadávanie a možnosť predísť vyhľadávaniu mýlnych informácii. Aplikácia teda v tejto časti vyhľadáva číslo, ktoré môže byť viacciferné a cifry môžu byť oddelené medzerou alebo čiarkou. Na faktúrach sa tiež uvádza cena s DPH a bez DPH, avšak aplikácia využíva skutočnosť, že celková cena býva uvádzana ako posledná a to sa môže vďaka pozícii ľahko určiť.

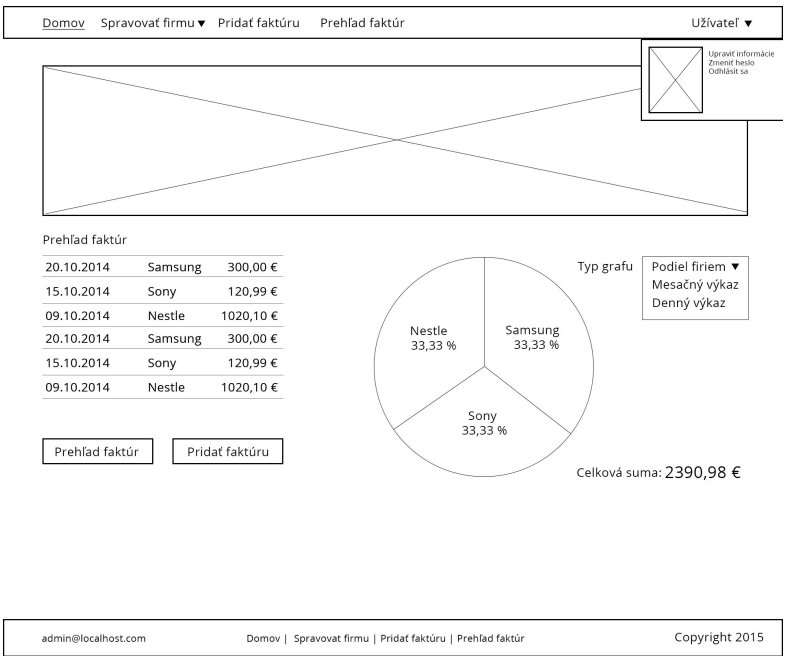
Aplikácia však nemusí vo všetkých prípadoch rozpoznať všetky znaky správne. Vykompenzovať sa to snaží tým, že ponúka užívateľovi možnosť výberu s viacerých možnosti a taktiež ručnú editáciu.

5.4 Návrh užívateľského rozhrania

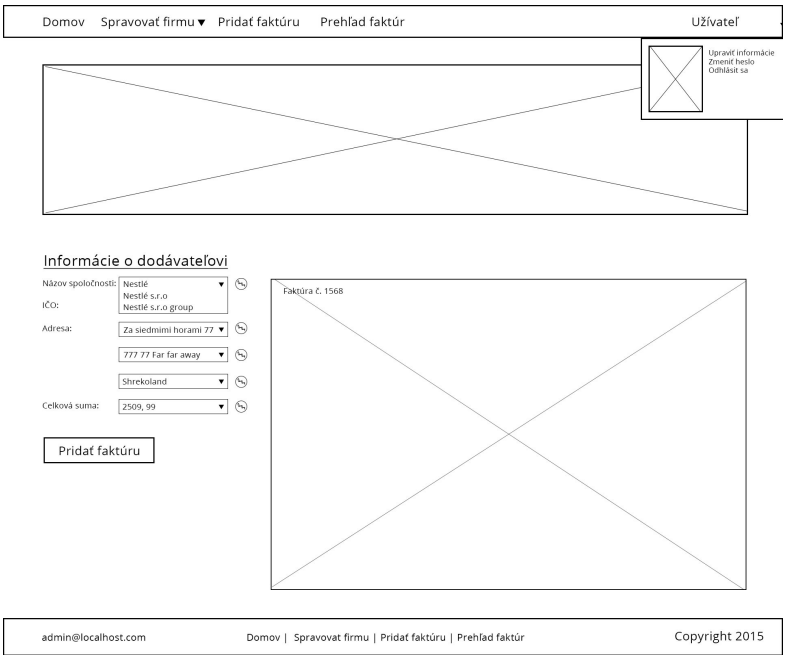
Kedže som sa rozhodol, že pôjde o webovú aplikáciu, aplikácia bude využívať pre svoje zobrazenie webové rozhranie, typicky nejaký webový prehliadač. Tomuto som sa snažil prispôsobiť aj uživateľské rozhranie, ktoré je zobrazené na obrázku. Snažil som sa o jednoduché a prehľadné užívateľské rozhranie, v ktorom sa užívateľ bude jednoducho orientovať a zároveň bude prístupné z akejkoľvek platformy. Užívateľa privíta prihlasovacia stránka, kde sú ponúknute dve polia na zadanie e-mailu a hesla. Obrázok 5.5 zobrazuje aplikáciu, ktorá sa bude užívateľom zobrazova po prihlásení do aplikácie. Podobne budú vyzerať aj ostatné stránky. Táto šablóna je teda rozdelená na:

- hlavičku - ktorá bude obsahovať menu s jednoduchou navigáciou pre využitie všetkých potrebných funkcií, ktoré aplikácia poskytuje, logo aplikácie a meno prihláseneho užívateľa
- obsah - bude obsahovať text, ktorý sa bude meniť podľa stránky, na ktorej sa užívateľ bude nachádzať
- pätička - bude obsahovať kontakt na administrátora a mapu stránok

Užívateľovi bude tiež umožnené vybrať typ faktúry, pred tým ako nahrá obrázok, ktorý chce rospoznať, čo by urýchlilo a upresnilo samotné rozoznávanie informácií. Obrázok 5.6 zobrazuje ako bude vyzerať a ako sa vypíšu výstupné data po spracovaní faktúry. Z vyššie uvedených častí však dôjde k zmene len v obsahovej časti. Tu sa užívateľovi zobrazia polia s možnosťou výberu korektných informácii. V prípade, že sa užívateľovi nebude pozdávať žiadna možnosť s poskytnutého výberu, bude mu poskytnutá možnosť upraviť tieto informácie manuálne. Vedľa týchto polí sa zobrazí obrázok faktúry, ktorú užívateľ poslal na spracovanie.



Obrázek 5.5: Návrh užívateľského rozhrania



Obrázek 5.6: Návrh užívateľského rozhrania

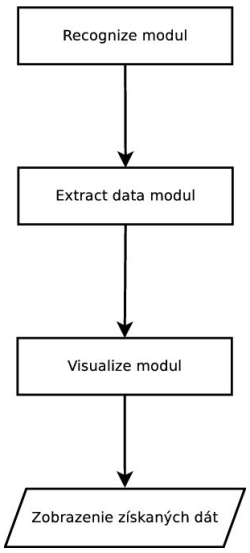
Kapitola 6

Realizácia

V prechádzajúcej kapitole bol uvedený celkový návrh, ako by mohla aplikácia vyzerať a tiež bol popísaný možný spôsob, ako by mohol fungovať algoritmus pre vyhľadávanie kľúčových slov. V tejto kapitole budem popisovať tvorbu a prepojenie jednotlivých modulov do celku.

6.1 Návrh aplikácie

Aplikácia bude fungovať ako webová aplikácia, založená na technológiach `PHP`, `MySQL`. K samostatnému získavaniu dát z faktúry by mala slúžiť aplikácia naprogramovaná v jazyku `C++` využívajúca knižnicu Tesseract 3.1. Užívateľské rozhranie bude implementované pomocou technológii `HTML`, `CSS` a `Javascriptu`. Celá aplikácia bude bežať na vzdialenom serveri a užívateľom sa bude zobrazovať prostredníctvom webového prehliadača. Aplikácia bude rozčelnená na jednotlivé moduly 6.1, kde každý bude mať svoju úlohu:



Obrázek 6.1: Postup spracovania faktúry

- `recognize modul` - má na starosti spracovanie obrázku a získanie potrebných dát, ktoré budú zobrazené užívateľovi, je zložený z `recogme.cc` a časť je implementovaná v `functions.php`

- `extract data modul` - má na starosti volanie `recognize modulu` a následné spracovanie dát, tvorí ho `functions.php` a doplňa ho `process_result.php`
- `visualize modul` - zobrazí užívateľovi získané výsledky a zoradí ich podľa najväčšej pravdepodobnosti zistenia po najmenšiu, implementácia v `functions.php` a `show_result.php`

Na server budú ukladané obrázky faktúr do špeciálneho adresára a do databázy sa zapíše len odkaz na tieto obrázky.

6.2 Návrh databázy

Samostatná databáza je rozdelená na tabuľky `address`, `company`, `invoice`, `users`, kde má každá tabuľka rôzne atribúty.
V tabuľke `address` sú uložené adresy jednotlivých firiem. Môže ísť o adresy, ktoré boli uložené pri registrácii firmy do systému, alebo adresy, ktoré boli získané pri spracovávaní faktúr, Tabuľka obsahuje tieto atribúty:

- `ID_Address`, `street`, `zip`, `city`, `country`

Tabuľka `company` obsahuje taktiež data už registrovaných firiem, alebo firiem, ktoré boli uložené pri spracovávaní faktúr. Jednotlivé atribúty tabuľky:

- `ID_Comp`, `Nazov`, `ICO`, `Email`, `Password`, `Phone`, `ID_Adress`

Informácie o jednotlivých faktúrach sú uložené v tabuľke `invoice`, ktoré sa získavajú pri rozoznávani s spracovaní faktúry, kedy sa tieto informácie uložia do databázy potvrdením užívateľa. Tabuľka obsahuje atribúty:

- `ID_Invo`, `datumSum`, `ID_Comp`

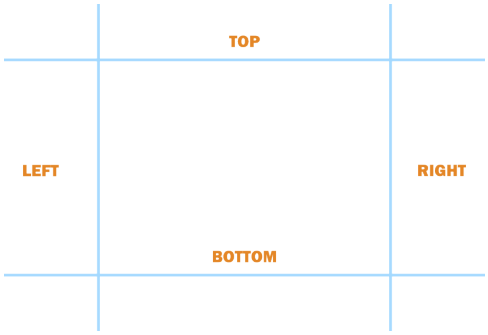
Jednotlivý užívatelia systému sú uložený v tabuľke `users`. Tí sú do tabuľky pridaný po registrácii do systému. Atribúty tejto tabuľky sú:

- `ID_User`, `ID_Comp`, `Meno`, `Priezvisko`, `Email`, `Password`, `Phone`

6.3 Recognize modul

Tento modul tvorí jadro celej aplikácie. Počas prvotnej implementácie bol tento modul naprogramovaný v jazyku `PHP` s využitím `TesseractOCR wrapperu`[1]. Tento wrapper však neobsahoval všetky potrebné funkcie a tak nepostačoval na implementáciu aplikácie. Nakoniec som sa rozhodol tento modul naprogramovať v jazyku `C++` za použitia knižnice `Tesseract` 3.1, ktorá je rozšírenejšia ako samotný `wrapper` pre jazyk `PHP` a obsahuje funkcie, potrebné pre správny chod tejto aplikácie. Spomeniem napríklad funkciu, ktorá zisťuje pozíciu nájdeného textu. Pozícia sa ukladá do dátovej štruktúry zvanej Bounding Box 6.2, ktorá obsahuje štyri údaje o pozícii a to vzdialenosť zľava, zhora, zprava, zdola. Ponúka tiež možnosť rozpoznávať a zobrazovať výsledky po blokoch, paragrafoch, riadkoch, slovách, alebo symboloch. Pri implementovaní aplikácie ma zaujímali z pozície len prvé dve možnosti a to vzdialenosť zľava, zhora a zobrazovanie výsledkov po slovách a riadkoch.
Modul na vstupe očakáva obrázok, ktorý podľa parametrov vyhľadáva potrebné informácie.

[1] Viz https://github.com/thiagoalessio/tesseract-ocr-for-php.



Obrázek 6.2: Bounding box

Ako prvé prebieha vyhľadávanie kľúčového slova **IČO**. Program si ukladá všetky nájdené výskyty aj s ich pozíciami. Následne program zaháji vyhľadávavanie samotného čísla. To je sprevádzané vyhľadávanim kľúčových slov **Dodávateľ**, alebo **Odberateľ**, čo umožní nastaviť váhu výsledného čísla. Od uloženej pozície program hľadá vo vzdialenosti o desať jednotiek zľava číslo vyhovujúce regulárnemu výrazu, ktorý odpovedá číslu o veľkosti osem, kde číslo môže obsahovať aj biele znaky. Uloží sa výskyt všetkých takýchto čísel spolu s ich váhou. Váha sa v tomto prípade nastavuje následovne:

- v prípade, že sa pred IČO nachádza slovo Dodávateľ váha sa rovná číslu 5
- v prípade, že sa pred IČO nachádza slovo Odberateľ váha sa rovná číslu 1
- v prípade, že sa pred IČO nenachádza ani jedno z vyššie uvedených slov váha sa rovná číslu 3

Na výstup sa zobrazí znak reprezentujúci výskyt Dodávateľa, Odberateľa, alebo možnosť, že sa tam žiadne z týchto slov nenachádza, pozície týchto slov, alebo pozícia IČO a samostatné výsledné číslo.
Od výskytu tohto slova je odvodené aj ďalšie vyhľadávanie. Ako ďalšie sa vyhľadáva **Meno dodávateľa**. Začína sa od pozície, ktorá bola určená v predchádzajúcom výstupe a vyhľadáva sa regulárny výraz, ktorý zahrňuje možnosť, že je meno zostavené z viacerých slov. Ak sa na faktúre vyskytuje slovo Dodávateľ, alebo Odberateľ, vyhľadávanie sa vykonáva vo vzdialenosti o 60 jednotiek smerom nahor. Program taktiež reaguje na situáciu, že tieto slová na faktúre nie sú a vyhľadávanie sa vykonáva do vzdialenosti o tristo jednotiek. Táto vzdialenosť bola zvolená zo štúdie, že niektoré faktúry majú medzi slovami odstavce, ktoré majú veľkost väčšiu ako 50 jednotiek (čo je zvyčajne približná veľkosť medzi riadkami). Túto skutočnosť som znázornil na obrázku 6.3 Pre čo najlepší odhad som vyhľadávanie obmedzil na tri najpravdepodobnejšie výsledky. Na výstup sa zobrazí opäť výsledok obsahujúci pozíciu, zistený reťazec a váha, ktorá tentokrát odpovedá vzdialenosti od začiatku vyhľadávania.
Nasleduje vyhľadávania **adresy**, ktorá je rozdelená na štyri časti.

- V prvej časti prebieha vyhľadávanie **ulice**. Vyhľadávanie prebieha rovnako ako v predchadzajúcich prípadoch, avšak vyhľadáva sa do vzdialenosti 250 jednotiek, čím som chcel pokryť možnosť, ak by bola adresa napísaná na viac riadkov. Opäť je vyhľadávanie obmedzené na tri najpravdepodobnejšie výsledky a na výstupe sa ukáže výsledok obsahujúci pozíciu, zistený reťazec a váha.

Obrázek 6.3: Vzdialenosť medzi jednotlivými riadkami

- V druhej časti sa vyhľadáva **PSČ**. PSČ sa vyhľadáva od zistenej pozície ulice a vyhľadávanie prebieha do vzdialenosti 55 jednotiek a hľadá sa reťazec zložený z čísel. V prípade úspešného hľadania, sa na výstup zobrazí pozícia výskytu tohto slova a samotný reťazec.
- V tretej časti sa hľadá **názov mesta**, alebo obce. Vyhľadávanie a výstup je podobný predchadzajúcim hľadaniam.
- V štvrtej časti sa vyhľadáva posledný záznam, potrebný pre kompletizáciu údajov o dodávateľovi. Vyhľadáva sa **krajina** a toto hľadanie je totožné s predchádzajúcimi. Na niektorých faktúrach sa krajina pôvodu nevyskytuje. Aplikácia to ale dokáže vykompenzovať interakciou, užívateľa, ktorý vyberie jazyk faktúry.

Ako posledné prebieha vyhľadávanie **ceny**. Tu vyhľadávanie prebieha po slovách a výsledky porovnáva s regulárnym výrazom, kde sa testuje výskyt čísel. Na výstupe sa zobrazia výsledky, ktoré budú zobrazované od najväčšieho po najmenší, kvôli predpokladu, že najvyššia suma je sumou aj celkovou.

6.4 Extract data modul

Modul je naprogramovaný v jazyku `PHP`. Spúšta sa v momente, keď užívateľ nahrá faktúru na server. Súbory sa ukladajú do priečinku `/imgs/`. Ďalej tento modul rozdelí obrázok na časti, ktoré už boli spomenuté pri analýze faktúry 5.2. Tieto časti sa po dobu vyhodnocovania ukladajú do špeciálneho priečinku `/imgs/temp/`. Každá časť začína názvom `"fa-n-string"`, kde `n` značí číslo časti a `string` automaticky vygenerovaný reťazec, aby sa predišlo možnému prepisovaniu týchto súborov. Každá z týchto častí je podrobená vyhľadávaniu. Tento modul spúšta `modul recognize` 6.3 pomocou príkazu `shell_exec`[1]. Výsledky jednotlivých vyhľadávaní sa ukladájú do poľa, kde su zoraďované zostupne podľa váhy výsledku a posielajú sa ďalej na spracovanie.

6.5 Visualize modul

Modul visualize pozostáva z viacerých častí. Jeho funkcionalita je písaná tiež v jazyku `PHP` a na výsledné zobrazenie užívateľovi sa používa `HTML`, `CSS` a `Javascript`. Modul pomocou funkcie získa data, ktoré majú byť poskytnuté užívateľovi, z modulu `extract data`. Všetky

tieto informácie sú uložené v dátovom type `array`[2]. Jednotlivé informácie o dodávateľovi a cene sú v tomto poli tiež uložené v poli a to z dôvodu, viacerých výsledkov. Tento modul, ale tieto výsledky rozoberie a rozdelí. Pole s možnými IČO číslami je na pozícii 0, mená dodávateľa na pozícii 1. Možné ulice na pozícii 2, poštové smerovacie čísla na pozícii 3, názvy miest a obcí na pozícii 4 a z informácii o dodávateľovi je krajina dodávateľa na pozícii 6. Ako posledný prvok sa v tomto poli nachádzajú pravdepodobné výsledky celkovej ceny faktúry a to teda na pozícii číslo 7.

Ďalšou časťou tohto modulu je zobraziť tieto výsledky do políčok výberu a v prípade chyby ponúknuť užívateľovi možnosť manuálnej úpravy. V prípade, že sú všetky uvedené informácie skontrolované užívateľom správne, modul tieto informácie uloží do tabuľky `invoice` a `company`.

6.6 Užívateľské rozhranie

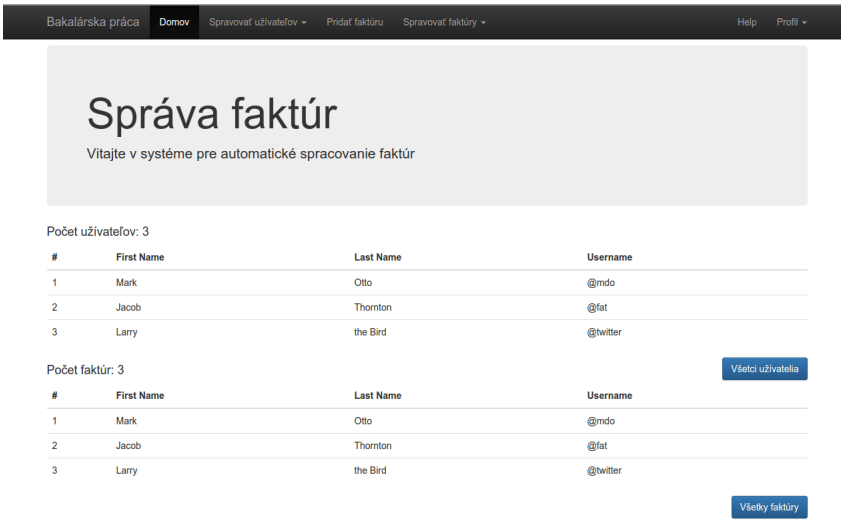
Kedže ide o webovú aplikáciu, pre implementáciu užívateľského rozhrania použijem `HTML`, `CSS` a knižnicu `jQuery`. Ako už bolo spomenuté v návrhu, užívateľské rozhranie by malo byť jednoduché, priehľadné a aplikácia by mala byť prístupná z akejkoľvek platformy. Aby som vyhovel všetkým požiadavkam siahol som po riešení, ktoré bude využívať `framework Bootstrap`, ktorý dokonale pokrýva všetky požiadavky.

Návštevníka stránky ako prvé privíta prihlasovacia stránka. Tá bola vytvorená použitím `Bootstrap` šablóny a konečné riešenie je možné vidieť na obrázku 6.4. Po prihlásení

Obrázek 6.4: Prihlasovacia stránka

sa užívateľovi zobrazí už klasická stránka, ktorá je členená na hlavičku, ktorá obsahuje jednoduchú navigáciu, obsah stránky, ktorá sa mení podľa stránky, na ktorej sa užívateľ nachádza. V prípade prihlásenia užívateľa, obsah tvorí prehľa vložených faktúr a tiež ich celková sumarizácia. Ďalšia a posledná časť stránky je pätička, ktorá obsahuje kontaktné informácie na administrátora a mapu stránky. Táto stránka by mala mať podobu podobnú tomuto obrázku 6.5.

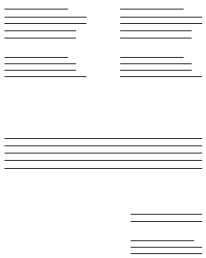
[1] Viz http://php.net/manual/en/function.shell-exec.php.
[2] Viz http://php.net/manual/en/language.types.array.php.



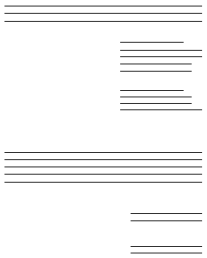
Obrázek 6.5: Hlavná stránka

Pred samotným zobrazením stránky s nahrávaním obrázka na server, sa užívateľovi ponúknu možnosti pre upresnenie typu faktúry. Hlavným využitím výberu týchto možnosti je upresnenie, ktorými smermi má aplikácia vyhľadávať informácie. Tiež to prispeje k lepším výsledkom pri vyhľadávaní a aj k lepšej rýchlosti aplikácie. Užívateľovi sa zobrazí stránka s dvomi možnosťami výberu a to:

- Prvý výber - šablóna je typu, kedy sú údaje o dodávateľovi zapísané horizontálne. Užívateľ si môže vybrať z obrázkových možností 6.6.
- Druhý výber - šablóna je typu, kedy sú údaje o dodávateľovi zapísané vertikálne. Tu si užívateľ môže vybrať 6.7.



Obrázek 6.6: Prvá možnosť výberu - horizontálne usporiadanie



Obrázek 6.7: Druhá možnosť výberu - vertikálne usporiadanie

Samotné zobrazenie výsledkov je riešené na ďalšej samostatnej stránke, kde je užívateľ presmerovaný po dokončení a spracovaní rozpoznávania. Užívateľovi sa zobrazia najlepšie zistené výsledky v rozklikávaciom tlačítku. Užívateľ si môže vybrať možnosť, ktorá najviac odpovedá hľadanému výrazu. Ak však aplikácia neponúka žiadnu zo správnych možností,

užívateľ môže túto informáciu upraviť a to je riešené tlačítkom upraviť, pri každom políčku. Vedľa zistených výsledkov sa užívateľovi zobrazí obrázok faktúry, nad ktorou chcel vykonať toto rozpoznávanie.

Obrázek 6.8: Stránka na zobrazovanie výsledkov

Užívateľ má tiež k dispozícii tlačítko pre pridanie faktúry do databázy, alebo na zahodenie vykonanej extrakcie. Po oboch akciách je užívateľ presmerovaný späť na stránku s prehľadom vložených faktúr, kde má k dispozícii číslo faktúry zaevidované v systéme, názov dodávateľa a cenu a tiež dátum vloženia.

Kapitola 7

Testovanie

V tejto časti sa zamerám na testovanie výslednej aplikácie. Za testovacie data som použil faktúry, ktoré boli použité počas implementácie aplikácie a budem na nich skúmať správnosť rozpoznania dát. Pre lepšie zohľadnenie výsledkov som použil tabuľku hodnôt, na základe ktorých bude prepočítana percentuálna úspešnosť jednotlivých testov 7.1. Touto tabuľkou bude prepočítavaná každá získaná hodnota, ktorých je dokopy 6 a konečné hodnotenie testu bude suma bodov a suma percent.

Akcia	**Počet bodov**	**Percentuálna úspešnosť**
Správe rozpoznanie	10	100%
Nutnosť výberu z ponuky	5	50%
Zle prečítané data, nutnosť zásahu	3	30%
Data chýbajú	0	0%

Tabulka 7.1: Tabuľka úspešnosti

Všetky testovacie data sú uložené v priečinku `tests/`.

Test 1

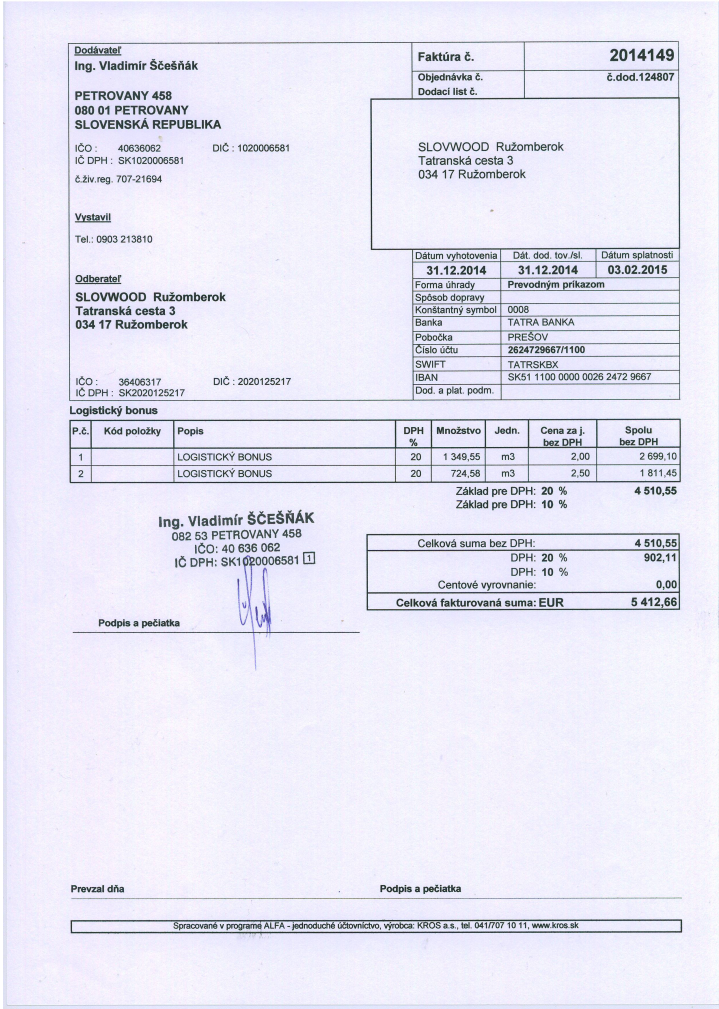
Pri tomto teste bol použitý obrázok `fakt1.jpg`. Pri rozpoznávaní bolo ohodnotených plným počtom bodov 4 hodnôt a to výber mena dodávateľa, IČO, Ulica a PSČ. Dve hodnoty, mesto a krajinu bolo potrebné vybrať z ponuky, ktorá bola ohodnotená počtom bodom 5.

Celkom bodov: 50b
Percentuálna úspešnosť: 83,33%

Test 2

Pri tomto teste bol použitý obrázok `fakt2.jpg`. Pri rozpoznávaní bolo ohodnotených plným počtom bodov 4 hodnôt a to výber mena dodávateľa, IČO, Ulica a PSČ. Dve hodnoty, mesto a krajinu bolo potrebné vybrať z ponuky, ktorá bola ohodnotená počtom bodom 5.

Celkom bodov: 50b
Percentuálna úspešnosť: 83,33%



Dodávateľ
Ing. Vladimír Ščešňák

PETROVANY 458
080 01 PETROVANY
SLOVENSKÁ REPUBLIKA

IČO : 40636062 DIČ : 1020006581
IČ DPH : SK1020006581
č.živ.reg. 707-21694

Vystavil

Tel.: 0903 213810

Odberateľ
SLOVWOOD Ružomberok
Tatranská cesta 3
034 17 Ružomberok

IČO : 36406317 DIČ : 2020125217
IČ DPH : SK2020125217

Faktúra č. 2014149
Objednávka č. č.dod.124807
Dodací list č.

SLOVWOOD Ružomberok
Tatranská cesta 3
034 17 Ružomberok

Dátum vyhotovenia	Dát. dod. tov./sl.	Dátum splatnosti
31.12.2014	31.12.2014	03.02.2015
Forma úhrady	Prevodným príkazom	
Spôsob dopravy		
Konštantný symbol	0008	
Banka	TATRA BANKA	
Pobočka	PREŠOV	
Číslo účtu	2624729667/1100	
SWIFT	TATRSKBX	
IBAN	SK51 1100 0000 0026 2472 9667	
Dod. a plat. podm.		

Logistický bonus

P.č.	Kód položky	Popis	DPH %	Množstvo	Jedn.	Cena za j. bez DPH	Spolu bez DPH
1		LOGISTICKÝ BONUS	20	1 349,55	m3	2,00	2 699,10
2		LOGISTICKÝ BONUS	20	724,58	m3	2,50	1 811,45

Základ pre DPH: 20 % 4 510,55
Základ pre DPH: 10 %

Ing. Vladimír ŠČEŠŇÁK
082 53 PETROVANY 458
IČO: 40 636 062
IČ DPH: SK1020006581

Celková suma bez DPH:	4 510,55
DPH: 20 %	902,11
DPH: 10 %	
Centové vyrovnanie:	0,00
Celková fakturovaná suma: EUR	5 412,66

Podpis a pečiatka

Prevzal dňa Podpis a pečiatka

Spracované v programe ALFA - jednoduché účtovníctvo, výrobca: KROS a.s., tel. 041/707 10 11, www.kros.sk

Obrázek 7.1: Testovací súbor č.1

Test 3

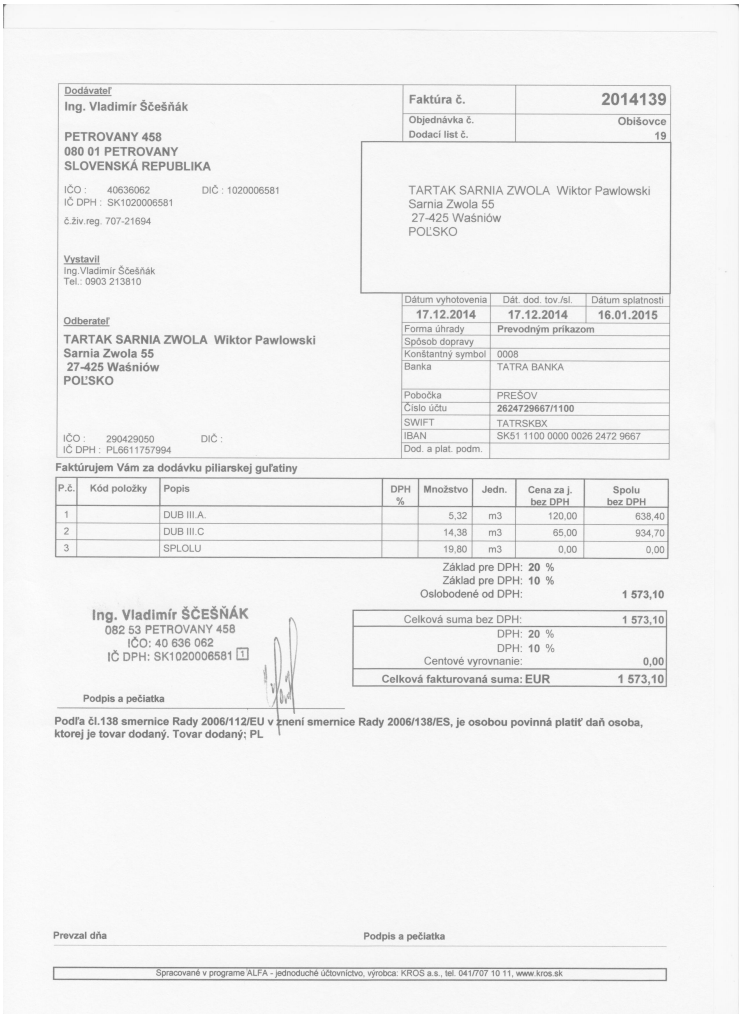
Pri tomto teste bol použitý obrázok `fakt3.jpg`. Pri rozpoznávaní bola ohodnotená plným počtom bodov len 1 hodnota a to výber IČO. Meno dodávateľa, ulica, PSČ a krajina neboloi rozoznané vôbec a boli ohodnotené 0 bodmi.

Celkom bodov: 5b
Percentuálna úspešnosť: 5,00%

Test 4

Pri tomto teste bol použitý obrázok `fakt4.jpg`. Pri rozpoznávaní boli ohodnotené plným počtom 4 hodnoty, ktorými boli meno dodávateľa, IČO, mesto, PSČ. K výberu boli dostupné, nie však na prvom mieste mesto a krajina, ktoré boli ohodnotené 5 bodmi.

Celkom bodov: 50b
Percentuálna úspešnosť: 83,33%



Dodávateľ
Ing. Vladimír Ščešňák

PETROVANY 458
080 01 PETROVANY
SLOVENSKÁ REPUBLIKA

IČO : 40636062 DIČ : 1020006581
IČ DPH : SK1020006581
č.živ.reg. 707-21694

Vystavil
Ing.Vladimír Ščešňák
Tel.: 0903 213810

Odberateľ
TARTAK SARNIA ZWOLA Wiktor Pawlowski
Sarnia Zwola 55
27-425 Waśniów
POĽSKO

IČO : 290429050 DIČ :
IČ DPH : PL6611757994

Faktúra č. 2014139
Objednávka č. Obišovce
Dodací list č. 19

TARTAK SARNIA ZWOLA Wiktor Pawlowski
Sarnia Zwola 55
27-425 Waśniów
POĽSKO

Dátum vyhotovenia	Dát. dod. tov./sl.	Dátum splatnosti
17.12.2014	17.12.2014	16.01.2015

Forma úhrady	Prevodným príkazom
Spôsob dopravy	
Konštantný symbol	0008
Banka	TATRA BANKA
Pobočka	PREŠOV
Číslo účtu	2624729667/1100
SWIFT	TATRSKBX
IBAN	SK51 1100 0000 0026 2472 9667
Dod. a plat. podm.	

Faktúrujem Vám za dodávku piliarskej guľatiny

P.č.	Kód položky	Popis	DPH %	Množstvo	Jedn.	Cena za j. bez DPH	Spolu bez DPH
1		DUB III.A.		5,32	m3	120,00	638,40
2		DUB III.C		14,38	m3	65,00	934,70
3		SPLOLU		19,80	m3	0,00	0,00

Základ pre DPH: 20 %
Základ pre DPH: 10 %
Oslobodené od DPH: 1 573,10

Ing. Vladimír ŠČEŠŇÁK
082 53 PETROVANY 458
IČO: 40 636 062
IČ DPH: SK1020006581

Celková suma bez DPH:	1 573,10
DPH: 20 %	
DPH: 10 %	
Centové vyrovnanie:	0,00
Celková fakturovaná suma: EUR	**1 573,10**

Podpis a pečiatka

Podľa čl.138 smernice Rady 2006/112/EU v znení smernice Rady 2006/138/ES, je osobou povinná platiť daň osoba, ktorej je tovar dodaný. Tovar dodaný: PL

Prevzal dňa

Podpis a pečiatka

Spracované v programe ALFA - jednoduché účtovníctvo, výrobca: KROS a.s., tel. 041/707 10 11, www.kros.sk

Obrázek 7.2: Testovací súbor č.2

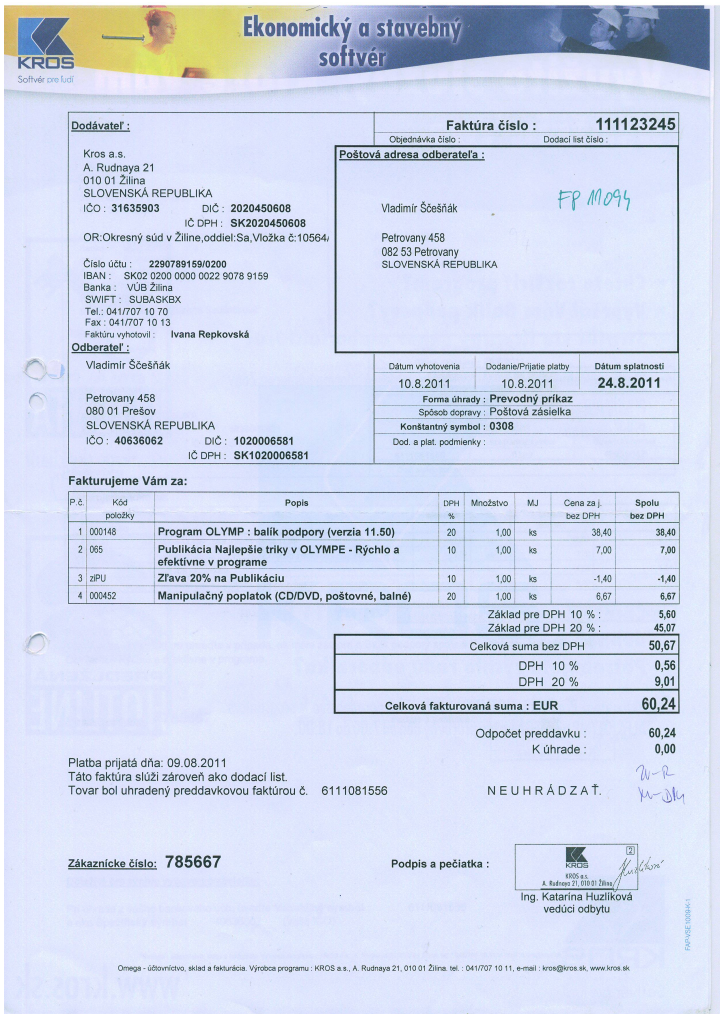
Test 5

Pri tomto teste bol použitý obrázok `fakt5.jpg`. Pri rozpoznávaní bola plným počtom ohodnotená jedna hodnota a to IČO. Pri ostatných hodnotách boli nutné zásahy, alebo sa nezobrazilo nič. Konkrétne pri názve dodávateľa a názve ulici, ktoré boli ohodnotené 3 bodmi. Názov mesta, PSČ a krajina rozoznané neboli a tieto rozoznania boli ohodnotené 0 bodmi.

Celkom bodov: 16b
Percentuálna úspešnosť: 26,66%

Súhrn testovania

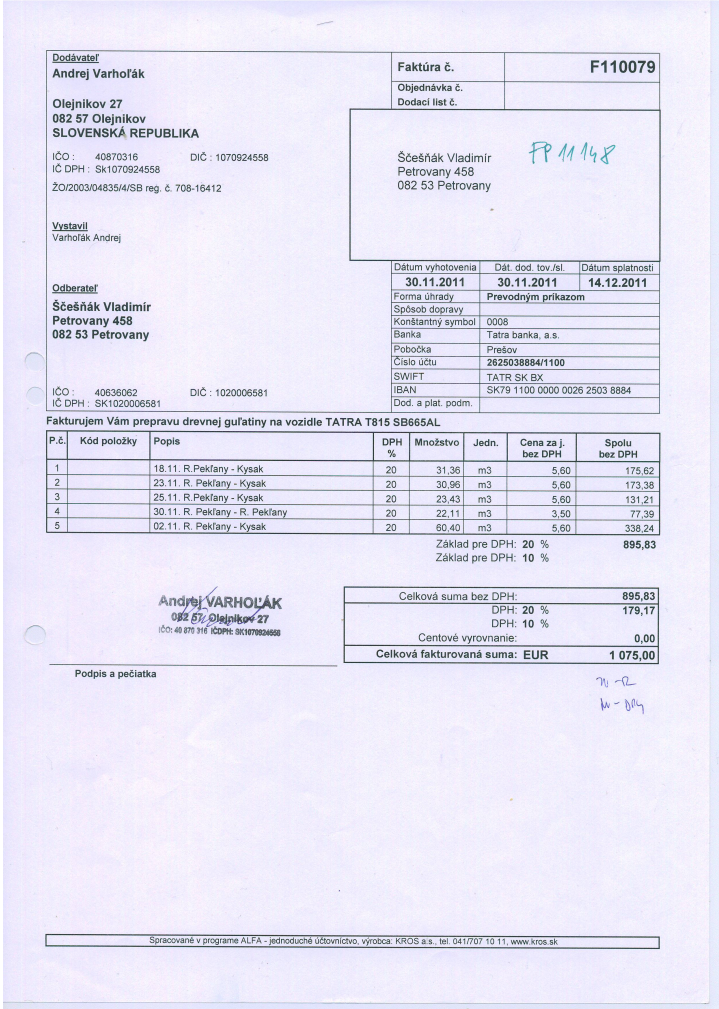
Testovanie neprebehlo podľa očakávania, nakoľko sa jednalo o podobné typy faktúr. Dá sa predpokladať, že vyhľadávací algoritmus nie je nastavení optimálne a nepohybuje sa v správnych medziach. Na niektorých faktúrach majú hodnoty iné pozície, na ktoré sa pri implementácii nebral ohľad. Naopak tesseract sa javil ako správne riešenie, nakoľko



Obrázek 7.3: Testovací súbor č.3

boli znaky rozpoznané správne a ak dochádzalo k nepresnostiam tak v prípadoch, kedy to prioritné hodnoty neovplyvňovalo.

Celkový súčet bodov: 171b z možných 600b
Percentuálna úspešnosť aplikácie: 28,50%



Dodávateľ
Andrej Varhoľák

Olejnikov 27
082 57 Olejnikov
SLOVENSKÁ REPUBLIKA

IČO : 40870316 DIČ : 1070924558
IČ DPH : Sk1070924558

ŽO/2003/04835/4/SB reg. č. 708-16412

Vystavil
Varhoľák Andrej

Odberateľ
Ščešňák Vladimír
Petrovany 458
082 53 Petrovany

IČO : 40636062 DIČ : 1020006581
IČ DPH : SK1020006581

Faktúra č.	F110079
Objednávka č.	
Dodací list č.	

Ščešňák Vladimír
Petrovany 458
082 53 Petrovany

FP 11148

Dátum vyhotovenia	Dát. dod. tov./sl.	Dátum splatnosti
30.11.2011	30.11.2011	14.12.2011
Forma úhrady	Prevodným príkazom	
Spôsob dopravy		
Konštantný symbol	0008	
Banka	Tatra banka, a.s.	
Pobočka	Prešov	
Číslo účtu	2625038884/1100	
SWIFT	TATR SK BX	
IBAN	SK79 1100 0000 0026 2503 8884	
Dod. a plat. podm.		

Fakturujem Vám prepravu drevnej guľatiny na vozidle TATRA T815 SB665AL

P.č.	Kód položky	Popis	DPH %	Množstvo	Jedn.	Cena za j. bez DPH	Spolu bez DPH
1		18.11. R.Pekľany - Kysak	20	31,36	m3	5,60	175,62
2		23.11. R. Pekľany - Kysak	20	30,96	m3	5,60	173,38
3		25.11. R.Pekľany - Kysak	20	23,43	m3	5,60	131,21
4		30.11. R. Pekľany - R. Pekľany	20	22,11	m3	3,50	77,39
5		02.11. R. Pekľany - Kysak	20	60,40	m3	5,60	338,24

Základ pre DPH: **20** % **895,83**
Základ pre DPH: **10** %

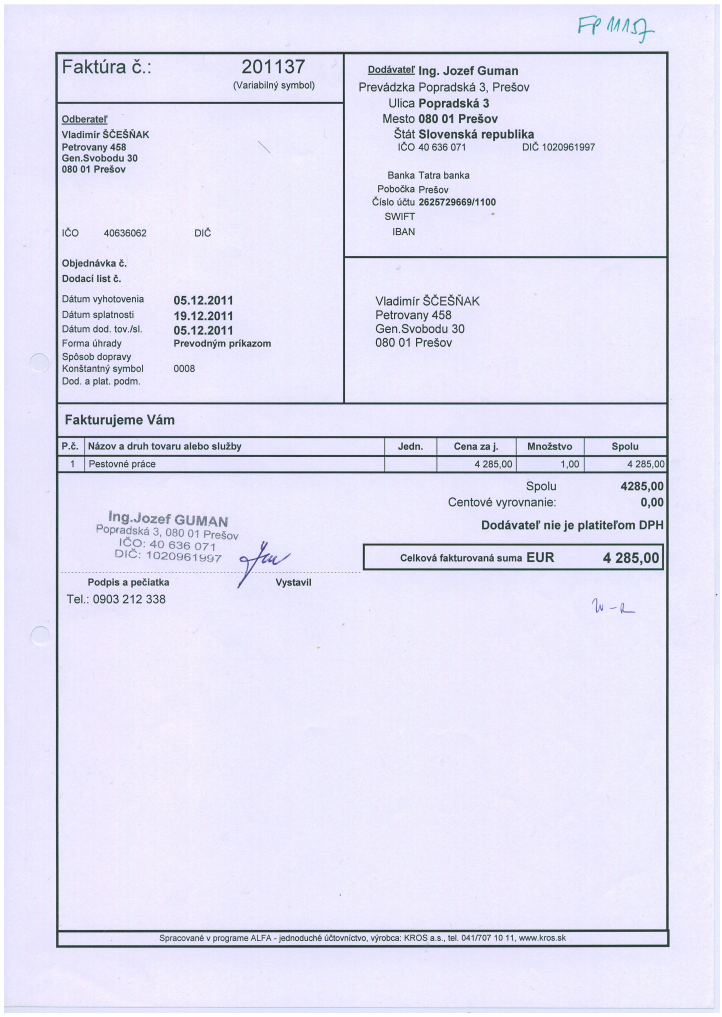
Andrej VARHOĽÁK
082 57 Olejnikov 27
IČO: 40 870 316 IČDPH: SK1070924558

Podpis a pečiatka

Celková suma bez DPH:	895,83
DPH: **20** %	179,17
DPH: **10** %	
Centové vyrovnanie:	0,00
Celková fakturovaná suma: EUR	**1 075,00**

Spracované v programe ALFA - jednoduché účtovníctvo, výrobca: KROS a.s., tel. 041/707 10 11, www.kros.sk

Obrázek 7.4: Testovací súbor č.4



Faktúra č.: 201137
(Variabilný symbol)

Odberateľ
Vladimír ŠČEŠŇAK
Petrovany 458
Gen.Svobodu 30
080 01 Prešov

IČO 40636062 DIČ

Objednávka č.
Dodací list č.

Dátum vyhotovenia 05.12.2011
Dátum splatnosti 19.12.2011
Dátum dod. tov./sl. 05.12.2011
Forma úhrady Prevodným príkazom
Spôsob dopravy
Konštantný symbol 0008
Dod. a plat. podm.

Dodávateľ Ing. Jozef Guman
Prevádzka Popradská 3, Prešov
Ulica Popradská 3
Mesto 080 01 Prešov
Štát Slovenská republika
IČO 40 636 071 DIČ 1020961997

Banka Tatra banka
Pobočka Prešov
Číslo účtu 2625729669/1100
SWIFT
IBAN

Vladimír ŠČEŠŇAK
Petrovany 458
Gen.Svobodu 30
080 01 Prešov

Fakturujeme Vám

P.č.	Názov a druh tovaru alebo služby	Jedn.	Cena za j.	Množstvo	Spolu
1	Pestovné práce		4 285,00	1,00	4 285,00

Spolu 4285,00
Centové vyrovnanie: 0,00
Dodávateľ nie je platiteľom DPH

Celková fakturovaná suma EUR 4 285,00

Ing.Jozef GUMAN
Popradská 3, 080 01 Prešov
IČO: 40 636 071
DIČ: 1020961997

Podpis a pečiatka Vystavil

Tel.: 0903 212 338

Spracované v programe ALFA - jednoduché účtovníctvo, výrobca: KROS a.s., tel. 041/707 10 11, www.kros.sk

Obrázek 7.5: Testovací súbor č.5

Kapitola 8

Závěr

Cieľom tejto práce bolo zoznámiť sa so základmi spracovania obrazu a rozpoznávania textu, čo bolo popísané v kapitolách 2 a 3. Ďalej bolo potrebné analyzovať už existujúce riešenia pre automatické spracovanie faktúr s využitím skeneru, alebo fotoaparátu. Zhrnúť ich vlastnosti a výhody a nevýhody, ktoré som opísal v kapitole 4. Ďalším bodom bolo zoznámiť sa s vhodnými technológiami, ktoré by splňovali požiadavky na automatický systém pre rozpoznávanie a spracovanie faktúr. Existuje mnoho riešení, ja som sa zameral na webové technológie spojené s knižnicou Tesseract pre jazyk C. Toto riešenie sa ukázalo ako vhodné, či už pri implementácii, ale hlavne pri testovaní, kedy sa potvrdilo, že Tesseract zvláda prácu s utf-8 znakmi a tiež so slovenskými a českými znakmi.

Nepodarilo sa úplne nasimulovať informačný systém, ktorý by bol schopný uspokojiť náročného užívateľa, preto má táto aplikácia len pár funkcií ako prihlásenie a pridanie novej faktúry a editácia tejto faktúry. Z hľadiska implementácie automatického rozpoznávania dôležitých dát na faktúre sa mi nepodarilo správne identifikovať cenu na faktúre, čo je ďaleko od mojich očakávaní nakoľko cena býva dosť dôležitá. Tiež som sa pri implementácii snažil zamerať, aby bola aplikácia schopná rozpoznávať informácie na faktúrach, ktoré sú pozostavené z rôznych šablón. Tento bod bol dotiahnutý len čiastočne, pretože občas sa stane, že sú niektoré dáta prečítané nesprávne, alebo nie sú nájdené vôbec.

Samotné testovanie ukázalo, že aplikácia nespĺňa základné požiadavky pre aplikáci na automatické rozpoznávanie faktúr, avšak dáva postačujúci základ tomu, aby sa vyvýjala ďalej. Či už ide o informačný systém, ktorý má veľké nedostatky a môže sa pracovať na jeho vzhľade, ktorý by užívateľa lákal k použitiu a tiež na funkcionalitách, alebo ide o rozpoznávací systém. Rozpoznávací systém nebol navrhnutý najlepšie a pri experimentoch bolo zistené, že vzdialenosti, ktoré som popísal pri implementácii nie sú najvhodnejšie a tiež môže byť chyba v regulárnych výrazoch, ktoré mali pôvodne uľahčiť vyhľadávanie. Dôsledkom toho konštatujem, že sa aplikáciu nepodarilo vypracovať podľa zadania no dáva slušný základ k tomu, aby bola rozvýjaná ďalej. Ako napríklad urýchlenie celého spracovávania súboru, by mohla slúžiť funckia, ktorá skontroluje existenciu IČO v databáze, a v prípade, že existuje tak sa nespracováva obrázok ďalej, ale doplnia sa údaje z databázy. Ďalej by bolo možné zisťovať dátum splatnosti faktúry, ktorý by sa ukladal do kalendára a kalendár by nás upozornil na dátum splatnosti. Načítanie jednotlivých položiek, a ich cien a rôzne triedenie faktúr. V pokročilejšom štádiu implementácie by nesmela chýbať možnosť nahrávať faktúry do systému napríklad hneď z telefónu.

Literatura

[1] Blázsovits, G.: Segmentácia. [Online], 2006.
URL http://dip.sccg.sk/segmenta/segment.htm

[2] Gary Bradski, A. K.: *Learning OpenCV*. O'Reilly Media, 2008, ISBN 978-059-6516-130, 555 s.

[3] Hakulin, L.: *OCR na platofme iOS*. bakalářská práce, FIT VUT v Brně, 2012.

[4] Hlaváč, V.: Matematická morfologie. [Online].
URL http://cmp.felk.cvut.cz/~hlavac/TeachPresCz/11DigZprObr/71-3MatMorpholBinCz.pdf

[5] Jan Viktora, P. V.: Optical Character Rocognition - OCR. [Online], 2008.
URL http://geo3.fsv.cvut.cz/vyuka/kapr/SP/2008_2009/vymetalek_viktora/index.html

[6] Milan Sonka, R. B., Vaclav Hlavac: *Image Processing, Analysis, and Machine Vission*. United States of America : THOMSON, 2008, ISBN 0-495-08252-X, 829 s.

[7] Smith, R.: An Overview of the Tesseract OCR Engine. [Online].
URL http://static.googleusercontent.com/media/research.google.com/cs//pubs/archive/33418.pdf

[8] Smith, R.: *The Extraction and Recognition of Text from Multimedia Document Images*. Dizertační práce, University of Bristol, 1987.

Příloha A

Obsah CD

- Zdrojové kódy aplikácie - `/source/`
- Testovacie súbory - `/tests/`
- Elektronická verzia tejto správy `/bc_latex/`
- Plagát `/plagat.png`

Příloha B

Manual

Funkčnosť aplikácie bola testovaná na Mysql Server vo verzii 14.14 distribúcii 5.5.43, ktorý je dostupný https://dev.mysql.com/downloads/mysql/5.5.html. A tiež PHP verzii 5.5.9. K správnemu behu aplikácie je tiež nutnosť povoliť príkaz `exec` a `shell_exec`, ktoré sú na niektorých serveroch zakázané. Je potrebné nastaviť prístupové práva v súbore `/source/db.inc.php`.Ďalej je potrebné spustiť súbor `/source/create_db.php` vytvorí databáza s užívateľom `admin@localhost` a heslo `heslo`, ktorý slúži na prihlásenie do systému.

Příloha C

Plakat

AUTOMATICKÉ ROZPOZNÁVANÍ A ZPRACOVÁNÍ FAKTUR

PREBDENÉ NOCI ZA POČÍTAČOM?

UŽ VIAC NEMUSÍTE !

PRIDAJTE **FAKTÚRU** VO FORMÁTE OBRÁZKU
A DÁTA, KTORÉ POTREBUJETE SA VÁM HNEĎ
ZOBRAZIA, BEZ TOHO
ABY STE ICH MUSELI ZADÁVAŤ MANUÁLNE

PODPORA JPG, PNG, TIF

NEPODPORUJE PDF A OSTATNÉ

RIEŠITEĽ: VLADIMÍR ŠČEŠŇÁK
VEDOUCÍ: ING. MICHAL ŠPANĚL, PH.D.

TECHNOLÓGIE: C++, TESSERACT, PHP, MYSQL
NA ZDOKONALENÍ APLIKÁCIE SA NEUSTÁLE PRACUJE