Sem vložte zadanie Vašej práce.

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra softwarového inženýrství

Bakalárska práca

Extrakce razítek z naskenovaných dokumentů a jejich úprava

Jakub Šiller

Vedúci práce: doc. RNDr. Ing. Marcel Jiřina, Ph.D.

7. mája 2015

Poďakovanie

Týmto by som chcel poďakovať doc. RNDr. Ing. Marcelovi Jiřinovi, Ph.D. za odborné vedenie pri tvorbe tejto bakalárskej práce. Ďalej by som chcel poďakovať za bezproblémovú spoluprácu pri integrácii mojej aplikácie do programu Surmon Jakubovi Novákovi. Nakoniec by som chcel poďakovať mojej rodine a priateľom za podporu počas celého bakalárskeho štúdia.

Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval(a) samostatne a že som uviedol(uviedla) všetky informačné zdroje v súlade s Metodickým pokynom o etickej príprave vysokoškolských záverečných prác.

Beriem na vedomie, že sa na moju prácu vzťahujú práva a povinnosti vyplývajúce zo zákona č. 121/2000 Sb., autorského zákona, v znení neskorších predpisov, a skutočnosť, že České vysoké učení technické v Praze má právo na uzavrenie licenčnej zmluvy o použití tejto práce ako školského diela podľa § 60 odst. 1 autorského zákona.

V Prahe 7. mája 2015 .....

Odkaz na túto prácu

Šiller, Jakub. *Extrakce razítek z naskenovaných dokumentů a jejich úprava.* Bakalárska práca. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.

Abstrakt

Táto bakalárska práca sa zaoberá problematikou rozpoznávania a extrakcie farebných pečiatok z naskenovaných dokumentov. Popisuje proces segmentácie stránky dokumentu na kandidátske oblasti, výpočet ich atribútov, následnu klasifikáciu a úpravy extrahovanej pečiatky, ako je zvýšenie kontrastu, odstránenie šumu, či otočenie do základnej polohy. V práci sa nachádza súhrn doterajších prístupov, návrh vlastného riešenia a jeho implemetácia, ktorá je založená na strojovom učení s využitím algoritmu k najbližších susedov. Práca na záver obsahuje výsledky a vyhodnotenie testov.

Kľúčové slová Dokument, pečiatka, extrakcia pečiatok, spracovanie dokumentu, spracovanie obrazu, segmentácia, výpočet atribútov, klasifikácia, k-nn.

Abstract

This bachelor thesis focuses on identification and extraction of colour stamps from scanned documents. It describes segmentation of a document page into candidates, feature extraction, classification and post processing of extracted stapms like contrast improvement, noise reduction or rotation to basic position. The thesis also includes summarized previous approaches of stamps extraction, proposal of a solution and its implementation based on machine learning that uses k nearest neighbours algorithm. At the end of the thesis, the tests results are listed.

Keywords Document, stamp, extraction of stamps, document processing, image processing, segmentation, feature extraction, clasification, k-nn.

Obsah

Zoznam obrázkov

Zoznam tabuliek

Úvod

Napriek tomu, že v dnešnej dobe sa dokumenty vytvárajú a uchovávajú vo veľkej miere elektronicky, stále sa veľmi často pracuje s dokumentami v papierovej forme. Platí to najmä pre oficiálne dokumenty ako sú napríklad zmluvy, úradné dokumenty, súdne dokumenty a podobne. Takéto dokumenty často obsahujú pečiatky, ktoré k dokumentu pridávajú ďalšie informácie a môžu slúžiť napríklad na overenie platnosti či pravosti dokumentu, alebo na identifikovanie osoby, respektíve organizácie, ktorá tento dokument vydala. Papierové dokumenty sa často digitalizujú a automatizovane spracovávajú a to hlavne v organizáciach s veľkým množstvom takýchto dokumentov.

Spracovanie pečiatok rozširuje možnosti spracovania dát a prácou s nimi. Umožňuje klasifikáciu dokumentov na základe typu pečiatky (napríklad rozlišovať dokumenty podľa pečiatky „schválené“ alebo „zamietnuté“), kontrolu toho či sú dokumenty overené a teda či obsahujú pečiatku, prípadne kým sú overené, hľadanie súvislostí medzi overenými dokumentami a osobami, ktoré ich overili pečiatkou a podobne. Preto som sa rozhodol v rámci svojej bakalárskej práce navrhnúť a implementovať modul do programu na spracovanie dát Surmon, ktorý bude schopný extrahovať farebné pečiatky z naskenovaného dokumentu a upraviť ich tak, aby mohli byť ďalej spracované, napríklad modulom na rozpoznávanie textu.

Práca je rozdelená do niekoľkých kapitol. Ciele mojej bakalárskej práce sú obsahom prvej kapitoly. V druhej kapitole charakterizujem problematiku a popisujem doterajšie prístupy v oblasti extrakcie pečiatok z naskenovaných dokumentov. Ďalšia kapitola obsahuje návrh riešenia založený na prácach z rešeršnej sekcie. Uvádzam v nej pojmy, ktoré sa v práci vyskytujú, farebné priestory, ktoré využívam a samotný návrh riešenia, ktorý najskôr v skratke predstavím a následne podrobnejšie popisujem v sekciách Segmentácia, Výpočet atribútov, Klasifikácia a Úprava extrahovaných pečiatok. V štvrtej kapitole sa venujem implementácií, konkrétne technológiam, ktoré som použil, štruktúre implementácie a integráciou do programu Surmon. Poslednú kapitolu tvoria výsledky testov a návrhy na zlepšenie.

Kapitola 1

Cieľ práce

Cieľom mojej bakalárskej práce je vytvoriť rešerš doterajších prístupov k extrakcií pečiatok z naskenovaných dokumentov, naštudovať potrebné algoritmi týkajúce sa spracovania obrazu a na základe toho navrhnúť a riešenie implementovať. Mojim cieľom je implementovať navrhnuté riešenie aj ako modul do programu na spracovanie dát Surmon, ktorý bude slúžiť na extrakciu pečiatok z naskenovaných dokumentov. Pečiatky, ktoré má byť modul schopný extrahovať môžu byť ľubovolného tvaru a farby a môžu byť prekryté alebo prekrývať iný grafický prvok. Po diskusií s vedúcim práce sa však obmedzím len na farebné pečiatky. Extrakcia čiernych pečiatok je ďalšia veľká výzva a vo svojej bakalárskej práci sa tým nebudem zaoberať. Na dokument sa nekladú žiadne špecialne požiadavky. Cieľom práce je ďalej, aby vo výslednom module po extrakcií prebehla ešte úprava extrahovaných pečiatok za účelom zlepšenia ich vizuálnych vlastností. Posledným cieľom je otestovať výsledok práce na dokumentoch z portálu www.justice.cz, vyhodnotiť výsledky a navrhnúť možné vylepšenia.

Kapitola 2

Súčasný stav v oblasti extrakcie pečiatok

2.1 Charakteristika problematiky

Extrakcia pečiatok zápasí s mnohými problémami, ktoré vyplývajú z vlastností pečiatok. V prvom rade, pečiatka môže mať akýkoľvek tvar. Pri pečiatkach štátnych inštitúcií alebo iných oficiálnych pečiatkach sa ešte dá očakávať jeden zo základných tvarov ako je obdĺžnik, kruh, elipsa, alebo trojuholník. Pre isté druhy dokumentov dokonca existujú predpisy, ktoré určujú ako má pečiatka v danom dokumente vyzerať. Napríklad zákon č. 63/1993 Z. z. Zákon Národnej rady Slovenskej republiky o štátnych symboloch Slovenskej republiky a ich používaní uvádza: „*. . . na ostatných úradných listinách, ktoré obsahujú rozhodnutie alebo uznesenie štátnych orgánov Slovenskej republiky, alebo osvedčenie dôležitých skutočností alebo oprávnení, sa používa odtlačok úradnej pečiatky so štátnym znakom. Úradné pečiatky so štátnym znakom sú okrúhle s priemerom 36 mm, po obvode kruhu okolo štátneho znaku je označenie štátneho orgánu alebo ustanovizne Slovenskej republiky, prípadne aj ich sídlo. Na osobitné účely, najmä na úradné tlačivá, sa používajú okrúhle úradné pečiatky so štátnym znakom s priemerom 17 mm a 22 mm. Orgány územnej samosprávy označujú úradnými pečiatkami so štátnym znakom a nápisom Slovenská republika listiny obsahujúce rozhodnutie alebo osvedčenie dôležitých skutočností alebo oprávnení vo veciach, v ktorých vykonávajú štátnu správu podľa osobitných predpisov*“. V komerčnej sfére však pečiatka môže mať ozdobný tvar, ktorý sa dá len veľmi ťažko odhadnúť. Predpokladať by sa mohlo, že prevažná väčšina pečiatok sú červené, čierne alebo modré, no rozhodne to nie je pravidlom. Z procesu, akým odtlačok pečiatky v dokumente vzniká vyplývajú ďalšie úskalia extrakcie pečiatok. Pečiatka môže byť v dokumente na ľubovolnej pozícií a ľubovolne pootočená. V podstate takmer nie je možné, aby pečiatka bola otočená v základnej polohe. Pečiatka často prekrýva iné časti dokumentu, alebo je nimi prekrytá (napríklad podpisom). Mnohokrát je toto prekrytie dokonca zámerné. Časť pečiatky môže chýbať alebo byť veľmi slabo viditeľná, pretože sa v danom mieste odtlačilo nedostatočné množstvo atramentu, alebo naopak môže byť,

a často býva rozmazaná. V konečnom dôsledku sa každý odtlačok pečiatky líši od ostatných odtlačkov a to aj v prípade, že sa jedná o odtlačky jednej fyzickej pečiatky. Nakoniec je treba ešte dodať, že kvalita dokumentu a proces digitalizácie tiež značne ovplyvňuje jeho spracovanie. Ako je vidno, problematika extrakcie pečiatok z naskenovaného dokumentu je relatívne rozsiahla a preto sa veľa doterajších prístupov zameriava len na časť problému.

2.2 Súčasný stav riešenia a doterajšie prístupy

V minulosti bolo publikovaných niekoľko prác zaoberajúcich sa detekciou a extrakciou pečiatok z dokumentov. Jednou z nich je publikácia s názvom *A robust stamp detection framework on degraded documents* [20]. Jej autori sa zameriavajú na okrúhle, prípadne elipsovité pečiatky. Metóda popísaná v tomto dokumente je založená na tom, že obrysy takýchto pečiatok sú analyticky vyjadriteľné. Na začiatku detekcie pečiatok je dokument preškálovaný na menšie rozlíšenie, čím sa odstránia jemné detaily a tým sa zefektívnia ďalšie fázy procesu. Pomocou Cannyho detektoru [6] sa detekujú hrany v dokumente a vytvoria sa z nich súvislé spojené komponenty. Na detekciu elipsovitých, respektíve kruhových pečiatok, autori využívajú vlastný detektor elíps, ktorého základom je získanie výrazu pre rodinu eliptických kriviek z informácí získaných z párov krajných bodov spojeného komponentu. Keďže sa pri tomto spôsobe riešienia detekcie pečiatok v dokumente využívajú čisto geometrické vlastnosti prvkov dokumentu, je možné ho aplikovať na farebné aj čiernobiele dokumenty.

Iba geometrické vlastnosti prvkov dokumentu využíva aj prístup k extrakcií pečiatok publikovaný s názvom *A Generic Method for Stamp Segmentation Using Part-based Features* [2]. Proces extrakcie v tomto dokumente začína tým, že sa pomocou algoritmu Features from Accelerated Segment Test (FAST) [45] vyhľadajú kľúčove body dokumentu, pre ktoré sa vypočítajú dva binárne deskriptory - Binary Robust Independent Elementary Features (BRIEF) [7] a Oriented FAST and Rotated BRIEF (ORB) [47]. Na detekovanie pečiatok v dokumente vytvorili autori trénovaciu množinu pečiatok rôznych kategórií a iných grafických prvkov, pre ktoré sa tiež predpočítajú deskriptory BRIEF a ORB v kľúčových bodoch. Tieto vypočítané deskriptory sa porovnajú s deskriptormi komponentov dokumentu a ak sa väčšina deskriptorov kandidáta podobá deskriptorom pečiatok z trénovacej množiny, tak je kandidát prehlásený za pečiatku. Nakoniec prebehne ešte filtrácia na základe jednoduchých geometrických vlastností ako je výška a šírka na odstránenie falošne detekovaných pečiatok.

Autori článku *Location algorithm for seal imprints on Chinese bank-checks based on region growing* [23] predstavujú metódu extrakcie pečiatok založenú na algoritme zväčšujúceho sa regiónu [3] [56]. Pomocou tohoto algoritmu vytvoria v dokumente spojené komponenty vopred určenej farby. Pokiaľ sa jeden komponent nachádza v druhom, zlúčia ich. Výsledné komponenty sú prehlásené za pečiatky. Pečiatka teda musí mať rám,

v opačnom prípade by sa každé písmeno považovalo za samostatnú pečiatku. Táto metóda je účinná pokiaľ sa v dokumente nevyskytujú iné grafické prvky predpokladanej farby pečiatky, pretože inak sú aj tie prehlásené za pečiatky. Ďalšou značnou nevýhodou tohoto riešenia je to, že farba pečiatky musí byť dopredu známa.

Extrakcia pečiatky a podpisu spôsobom popísaným v *Extraction of Signature and Seal Imprint from Bankchecks by Using Color Information* [54] je založená na predpoklade, že pečiatka a podpis majú odlišné farby, ktoré sa zároveň líšia od farby zvyšku dokumentu. Predpokladajú teda, že pixele dokumentu vytvoria tri farebné zhluky vzťahujúce sa k pozadiu, podpisu a pečiatke. Z maximálnych hodnôt trojrozmerného (RGB) histogramu jasu určia stredy uvedených zhlukov. Pomocou niekoľkých projekcií pixelov dokumentu a hraničných hodnôt sa extrahuje podpis a pečiatka. Extrahovaná pečiatka môže mať v miestach prekrytia diery. Vyhľadajú sa preto koncové body čiar pečiatky a ak je blízko koncového bodu čiara podpisu, bod sa označí ako kandidát na obnovenie čiary pečiatky. Dvaja kadidáti na spojenie sú spojení, ak vzdialenosť medzi nimi je približne rovnaká ako priemerná šírka čiary podpisu a zároveň uhol medzi priamkami, ktoré vznikli lineárnou regresiou z pixelov okolo daných kandidátov, je tupý.

Za účelom zvýšenia bezpečnosti pri automatizovanom spracovaní dokumentov sa extrakciou pečiatok zaoberá práca *Stamp Detection in Color Document Images* [34], ktorú článok *Stamp Verification for Automated Document Authentication* [35] rozširuje o fázu overenia pravosti pečiatky. Autori prezentujú riešenie, ktoré extrahuje pečiatky rozličných farieb (s výnimkou čiernej) či tvarov, ktoré môžu byť prípadne prekryté podpisom alebo textom odlišnej farby. Metóda, ktorú popisujú, pozostáva z niekoľkých krokov. Najskôr sa odstráni čierny text a pozadie od ostatných farebných prvkov dokumentu. Následne prebehne farebné zhlukovanie, ktorým sa získa niekoľko obrázkov (pre každú farbu jeden obrázok) dokumentu obsahujúcich len prvky rovnakej farby. V ďalšom kroku sú jednotlivé prvky od seba oddelené pomocou algoritmu XY-cut [30] a považujú sa za kandidátov na pečiatku. Nakoniec na základe porovnania vlastností vypočítaných pre jednotlivých kandidátov s referenčými hodnotami daných vlastností získaných z experimentov, alebo z prvkov trénovacej množiny je kandidát klasifikovaný ako pečiatka, alebo ako iný grafický prvok. Autori uvádzajú, že pre klasifikáciu sa im najviac osvedčil algoritmus Support Vector Machines (SVM) [25].

O rozpoznávaní falošnej pečiatky od pravej sa píše aj v článku *A Robust Registration and Detection Method for Color Seal Verification* [5]. Jeho autori očakávajú, že pečiatka bude okrúhla, elipsovitá, alebo obdĺžnik a jej farba bude červená alebo modrá. Farebný priestor RGB rozdelia na osem podpriestorov, ktoré prislúchajú ôsmim farbám: bielej, čiernej, červenej, zelenej, žltej, modrej, purpurovej a azúrovej. Farba každého pixla spadne do jedného z podpriestorov. Autori extrahujú pixely, ktoré spadajú do podpriestoru červenej (respektíve modrej) farby. Použitím thinning algorithm [48] vytvoria kostru kandidáta, ktorá je využitá na zistenie tvaru kandidáta. Pomocou vzdialeností medzi bodmi kostry určia osi kandidáta a uhol otočenia od základnej polohy.

Kandidát sa v ďalších krokoch porovnáva s jeho predlohou a určuje sa, či je odtlačkom pravej pečiatky.

Autori článku *Robust Stamps Detection and Classification by Means of General Shape Analysis* [17] rozdeľujú algoritmus, ktorý popisujú, na 3 časti. V prvej časti detekujú kadidátov na pečiatky. Dokument prevedú do farebného priestoru YC_bC_r a ďalej pracujú už len so zložkami C_b a C_r. Pomocou projekcie matice reprezentujúcej dokument v zložke C_b (respektíve C_r) v horizontálnom a následne vertikálnom smere hľadajú oblasti s vysokou intenzitou týchto zložiek. Tieto oblasti potom považujú za kandidátov na pečiatku. V druhej časti odfiltrujú prvky, ktoré nie sú pečiatky, pomocou jednoduchých geometrických vlastností ako je pomer strán alebo pomer obsahu plochy kandidáta k obsahu plochy dokumentu. V poslednej časti určujú jeden z piatich tvarov pečiatky (kruh, ovál, štvorec, obdĺžnik a trojuholník). Tátó fáza je bližšie popísaná v článkoch *Efficient Stamps Classification by Means of Point Distance Histogram and Discrete Cosine Transform* [15] a *Principal Component Analysis of Point Distance Histogram for Recognition of Stamp Silhouettes* [16]. Využívajú pri tom histogram vzdialeností bodov obrysu vyjadrených v polárnych súradniciach, nazývaný Point Distance Histogram (PDH). PDH je po rotáci, zmene mierky, či posunutí nemenný. Kvôli veľkej dimenzii histogramu prebehne redukcia dimenzií využitím diskrétnej kosínusovej transformácií [1]. Pomocou Euklidovskej vzidalenosti od šablón uvedených tvarov sa určí tvar pečiatky.

Na základe viacerých predchádzajúcich prác v oblasti extrakcie dokumentov sa autori článku *Stamps Detection and Classification Using Simple Features Ensemble* [18] snažili vytvoriť komplexný systém detekcie a klasifikácie farebných i čiernobielych pečiatok rozličných tvarov. V dokumente vyhľadávajú tri rozličné tvary - čiara, kruh a ostatné tvary, pomocou ktorých následne určujú kandidátské oblasti. Kruhy hľadajú využitím Houghovej kruhovej transformácie [10], detektor čiar je taktiež založený na Houghovej transformácií [10] a nepravidelné tvary sa extrahujú ak ich výška a šírka spadá do určeného intervalu. Nasleduje výpočet vlastností kandidátskych oblastí, ktoré sú vstupom pre dvojstupňovú klasifikáciu, ktorá využíva trénovaciu množinu obsahujúcu pečiatky, ale aj iné grafické prvky. V nej sa v prvom stupni rozhodne či kandidát je alebo nie je pečiatkou a v druhej sa určí trieda tvaru pečiatky .

V dokumente, ktorý nesie názov *Seal Object Detection in Document Images using GHT of Local Component Shapes* [46] je popísané riešenie detekcie pečiatok, ktoré využíva rozpoznané znaky dokumentu ako vysoko úrovňové deskriptory. Každý znak vytvorí so svojimi k najbližšími susedmi k párov. Pre všetky tieto páry sa vypočítajú priestorové vlastnosti na základe vzdialenosti a rotácie. Pomocou vypočítaných vlastností sa vytvoria hypotézy o polohe referenčného bodu pečiatky (stred kruhu s minimálnym polomerom ohraničujúci pečiatku). Nahromadením a spracovaním hypotéz sa pomocou zovšeobecnenej Houghovej transformácie [10] detekuje referenčná pečiatka (ak sa v dokumente nachádza).

Metóda popísaná v práci *Automatic Seal Information Reader* [13] začína run leght smoothing algoritmom (RLSA) [19], ktorý odstráni medzery vo vnútri znakov, medzi znakmi, medzi slovami a aj medzi riadkami v odstavci. Tým rozdelí dokument na časti nazývané „stavebné bloky". Stavebné bloky, ktoré prejdú prvotnou filtráciou výšky a šírky sú kandidátmi na pečiatku. Autori zvolili tri konštantné znakové reťazce na vytvorenie šablón. Ak stavebný blok obsahuje všetky tri konštanté znakové reťazce, postupuje ďalej do testu topológie a ak ním prejde, je prehlásený za pečiatku. Nakoniec ešte prebehne pomocou trénovacej množiny pečiatok a prvkov iných ako pečiatky a algoritmu k najbližších susedov [37] test na falošne detekované pečiatky.

Základom prístupu publikovanom v článku *Color Seal Extraction from Documents: Robustness through Soft Data Fusion* [50] je takzvaný fuzzy integrál [33], ktorý využíva čisto farebné informácie dokumentu. Autor považuje farebný obrázok za multisenzorový signál. Pomocou dvoch rôznych fuzzy integrálov získa dva obrazy dokumentu v odtieňoch šedej. Pomocou ich rozdielu sa vytvorí binárna maska, ktorá je neskôr použítá na vymaskovanie dokumentu, čím sa extrahujú pečiatky.

Ako extrahovať okrúhlu alebo obdĺžnikovú pečiatku z tradičných čínskych malieb popisujú autori v práci *An Effective Method to Detect Seal Images from Traditional Chinese Paintings* [21]. Z pozorovaní vyplýva, že pečiatky na tradičných čínskych maľbách sú červené. Na základe tohoto poznatku extrahujú červené prvky dokumentu, ktoré sa stanú kandidátmi na pečiatku. Každému kandidátovi vypočítajú najmenší obdĺžnik, ktorý ho ohraničuje a jeho vpísanú elipsu. Pre oba tvary vytvoria grafy závislosti uhla a vzdialenosti bodu na obvode od stredu útvaru. Obdobný graf vytvoria pre reálne okrajové body kandidáta a porovnajú ich s predchádzajúcimi dvoma. Pokiaľ je rozdiel s jedným z grafov dostatočne malý, kandidát je pečiatkou tvaru, ktorému odpovedá porovnávaný graf.

Výlučne farebné informácie využíva metóda extrakcie pečiatok popísaná v práci *Seal Imprint Segmentation Based on Color Feature Classifier* [40]. Vo farebnom priestore HSV , ktorý je podľa autorov menej náchylný na zmeny v jase, sa pre každý pixel vypočítajú deskriptory Scale-invariant feature transform (SIFT) [32] pre odtieň (hue) a sýtosť (saturation) a označia sa ako HS-SIFT . Vo fáze učenia sa vypočítajú HS-SIFT deskriptory pre ručne označené pozitívne vzorky (odtlačky pečiatky) a negatívne vzorky (pozadie). Autori používajú klasifikačný algoritmus SVM [25]. Vo fáze segmentácie sa pre daný pixel vypočítajú HS-SIFT deskriptory a na základe ich skóre sa rozhodne či pixel je pixel pečiatky alebo nie.

Autori práce *Colored Rubber Stamp Removal from Document Images* [11] naopak odstraňujú pečiatku z dokumentu. Dokument sa najskôr prevedie do binárneho dokumentu, ktorý sa využije ako maska pôvodného dokumentu, čím sa odstráni pôvodné pozadie a ostane jednotné biele pozadie. Dokument sa následne prevedie do farebného priestoru HSV. Vypočíta sa vlastný vektor kovariančnej matice priestoru HSV a vytvorí sa

histogram skalárnych súčinov daného vlastného vektoru a vektorov farieb pixelov v priestore HSV. Histogram obsahuje dva vrcholy - jeden reprezentuje farbu textu a druhý pečiatky. Pomocou Otsu's thresholding algorithm [39] je histogram, a teda aj dokument, rozdelený na text a pečiatky.

Ďalšie články, ako napríklad *Automatic Seal Imprint Verification System for Bankcheck Processing* [53], sa často zameriavajú na konkrétny problém, pri ktorom majú relatívne jasne definované ako vyzerá spracovávaný dokument a pečiatka v ňom. Častokrát je dopredu známe, že pečiatka je červená a nič iné červené nie je. V týchto článkoch ide potom najmä o rozoznanie falošnej pečiatky od originálnej.

Uvedené práce sa dajú rozdeliť do troch kategórií podľa toho, aké informácie z dokumentu využívajú za účelom extrakcie pečiatok. Časť autorov publikovalo riešenia, ktoré využívajú čisto farebné informácie. Iní autori naopak len geometrické vlastnosti prvkov dokumentu a niektorí autori využívali oboje. Do prvej kategórie spadajú články [11] [23], [40], [50] a [54]. Do druhej kategórie sa radia [2], [13], [18], [20] či [46] a poslednú kategóriu tvoria [5] , [17], [21] i [35]. Na základe spôsobu využitia získaných informácií a spôsobu riešienia sa tieto práce môžu rozdeliť do iných troch kategórií. Medzi práce, ktoré využívajú dopredu vytvorenú množinu pečiatok, prípadne iných grafickych prvkov, patria [2], [18], [35], [40] a [46]. Pri určovaní tvaru pečiatky využiva [17] množinu tvarov, preto som ho tiež zaradíl do tejto kategórie. [5], [11], [21], [23], [50] a [54] sú práce, ktoré jednoducho extrahujú pixele dopredu danej alebo analýzou zistenej konkrétnej farby. Treťou kategóriou sú ostatné prístupy k extrakcií pečiatok z dokumentu, medzi ktoré patrí [13] a [20].

Kapitola 3

Návrh riešenia

3.1 Návrh riešenia v skratke

Na základe poznatkov získaných z vyššie popísaných prác som zvolil k extrakcii pečiatok z naskenovaných dokumentov prístup využívajúci strojové učenie. Moje riešenie som založil najmä na prácach [18] a [34]. Princíp, na ktorom je založené moje riešenie je jednoduchý. Skladá sa z dvoch fáz. Prvá fáza je prípravná, prebieha v nej učenie a je jednorázová. Druhá, pracovná fáza, slúži na samotnú extrakciu pečiatok z neznámeho naskenovaného dokumentu. Obe fázy majú tri základné časti: segmentáciu, výpočet atribútov a klasifikáciu.

Počas segmenácie sa stránka dokumentu rozdelí na oblasti, ktoré môžu ale nemusia byť pečiatkou. Tieto oblasti sa nazývajú kandidáti alebo kandidátske oblasti. Pre každú kandidátsku oblasť sa vypočítajú vlastnosti, ktoré ju charakterizujú. Tieto vlastnosti sa často nazývajú atribúty. Príkladom atribútu môže byť napríklad pomer výšky a šírky kandidáta. Nasleduje klasifikácia, v ktorej sa určí či kandidát je pečiatkou alebo nie. V pripravnej fáze sa manuálne označí, do ktorej skupiny kandidát patrí. V pracovnej fáze sa atribúty kandidáta porovnajú algoritmom k najbližších susedov (popisujem v sekcií 3.6) s atribútmi už klasifikovaných kandidátov z prípavnej fázy a podľa toho sa určí, do ktorej skupiny kandidát patrí. Za účelom zlepšenia vizuálnych vlastností extrahovaných pečiatok nasleduje po klasifikacií v pracovnej fáze ešte ich úprava.

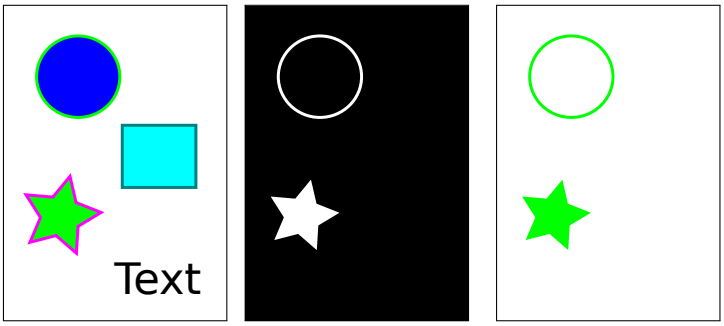
Pre zefektívnenie segmentačnej fázy a pre zjednotenie parametrov stránok dokumentov je stránka dokumentu pred vstupom do celého procesu zmenšená tak, aby dĺžky jej strán boli menšie ako stanovené hranice. Podrobnejšie sú jednolivé časti algoritmu popísané v nasledujúcich sekciach.

3.2 Používané pojmy

Predtým ako začnem popisovať jednotlivé fázy procesu, uvediem pár pojmov, ktoré budem v práci používať.

Pojem *extrakcia* budem používať pre výber prvku, prípadne prvkov z celku ako napríklad pečiatku z dokumentu.

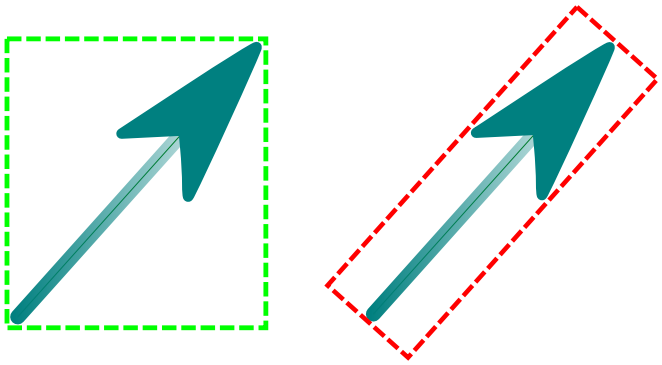
Binárna maska alebo len *maska* je čiernobiely obrázok, ktorý určuje, ktoré pixele pôvodného obrázku sa ďalej použijú alebo zobrazia alebo podobne. Príklad použitia binárnej masky je na obrázku 3.1.



(a) Pôvodný obrázok. (b) Binárna maska. (c) Obrázok po aplikovaní masky.

Obr. 3.1: Ukážka aplikácie binárnej masky na obrázok.

Ďalej budem používať pojmy *ohraničujúci obdĺžnik* a *najmenší ohraničujúci obdĺžnik*. Rozdiel medzi nimi je ten, že *ohraničujúci obdĺžnik* má strany rovnobežné so osami X, Y respektíve so stranami strany dokumentu a *najmenší ohraničujúci obdĺžnik* je najmenší obĺžnik, ktorý ohraničuje prvok. Názorná ukážka je na obrázku 3.2.



(a) Ohraničujúci obdĺžnik. (b) Najmenší ohraničujúci obdĺžnik.

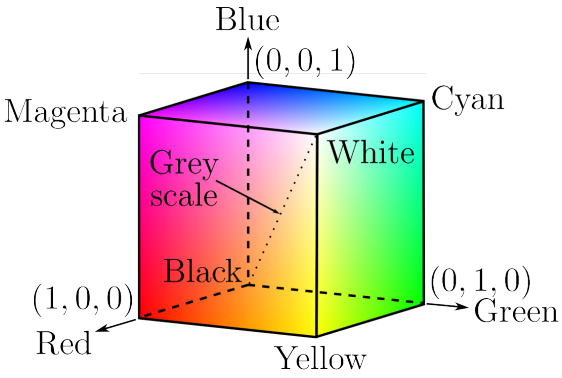
Obr. 3.2: Rozdiel medzi *ohraničujúcim obdĺžnikom* a *najmenším ohraničujúcim obdĺžnikom* šípky.

3.3 Farebné priestory RGB, HSV a YC_bC_r

V kapitole 3 v publikácií [28] pojednávajúcej o farebných priestoroch sa uvádza, že farba je vnem napriek tomu, že viditeľné spektrum vlnových dĺžiek elektromagnetického žiarenia je merateľné a fyzikálne popísatelné. Reprezentácia farby musí byť vhodná pre zaznamenávanie, zobrazovanie a spracovanie obrazu, musí vyhovovať matematickým požiadavkam spracovania obrazu, rešpektovať technické podmienky pre monitory, fotoaparáty, skenery a podobne a zároveň vyhovovať ľudskému vnímaniu. Pretože tieto podmienky sa ťažko dajú zabezpečiť súčasne, využívajú sa za rôznym účelom rôzne reprezentácie.

3.3.1 RGB

Farebný priestor RGB je najpoužívanejší farebný priestor a je najpodobnejší tomu, ako ľudské oko vníma farby. Je založený na aditívnom miešaní troch základnych farieb: červenej R (anglicky red), zelenej G (anglicky green) a modrej B (anglicky blue). Aditívne miešanie znamená, že farby sa k sebe pričítavajú. Tri základné farby vytvárajú bázové vektory ortogonálneho priestoru RGB a nulový vektor reprezentuje čiernu. Každá farba sa dá potom vyjadriť ako lineárna kombinácia bázových vektorov. Každý pixel obrázku je potom vyjadrený ako vektor súradníc v priestore RGB. Farebný priestor RGB je základným farebným priestorom pre digitálne spracovanie obrazu. Autori [34] ale píšu, že tento farebný priestor nie je vhodný na segmentáciu kvôli vysokej korelácií medzi zložkami RGB.

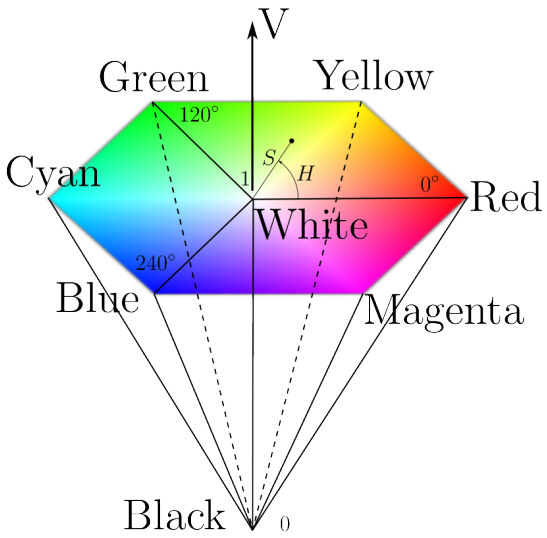


Obr. 3.3: Ukážka priestoru RGB [zdroj: [8]].

3.3.2 HSV

HSV je skratka pre farebný priestor, ktorého bázové vektory reprezentujú odtieň H (anglicky hue), sýtosť S (anglicky saturation) a jas V (anglicky brigthness value). Takáto reprezentácia je založená na ľudskom intuitívnom vnímaní farby. Projekcia priestoru RGB do priestoru HSV (viď obrázok 3.4) vytvorí prevrátenú pyramídu. Uhol okolo vertikálnej osi vyjadruje odtieň H, pričom uhol 0° reprezentuje červenú. Vzdialenosť od

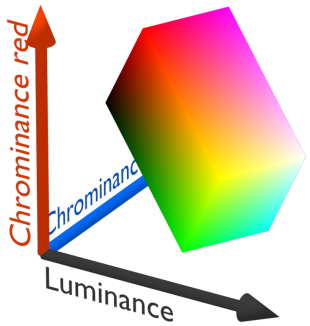
vertikálnej osi je hodnota saturácie S a hodnota jasu V leží medzi 0 pri vrchole pyramídy a 1 pri jej podstave. Tento farebný priestor som zvolil preto, aby sa dal na základe hodnôt zložky H jednoducho odlíšiť odtien pixelov.



Obr. 3.4: Ukážka priestoru HSV [zdroj: [9]].

3.3.3 YCbCr

Autori práce [34], ale aj autori [18] využívajú farebný priestor YC_bC_r. Zložka Y reprezentuje intenzitu svetla a zložky C_b, C_r vyjadrujú intenzitu modrej respektíve červenej zložky relatívne k zelenej zložke [31]. V [18] sa uvádza, že tento farebný priestor sa využíva preto, lebo pečiatky bývajú prevažne červené alebo modré.



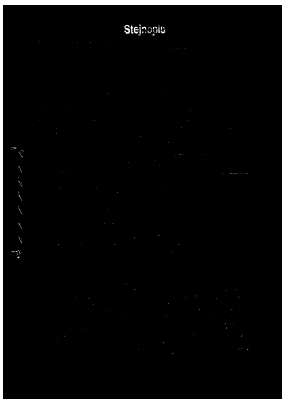
Obr. 3.5: Ukážka priestoru YC_bC_r. [zdroj: [55]]

3.4 Segmentácia

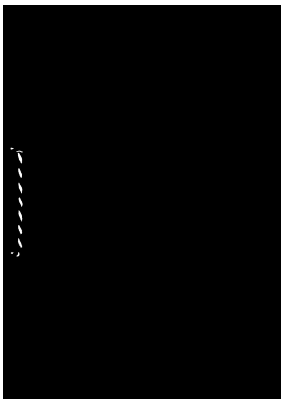
Vstupom pre túto časť algoritmu je stránka dokumentu. Počas segmentácie sa oddelia jednotlivé grafické prvky stránky, o ktorých sa v ďalších častiach rozhodne či sú alebo nie sú pečiatkami.

3.4.1 Rozdelenie dokumentu na farebné vrstvy

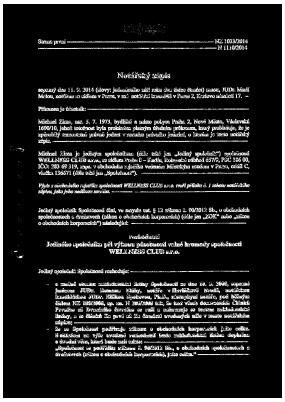
Podobne ako v [34] sa na začiatku segmentácie rozdelí dokument na jednotlivé farebné (chromatické) vrstvy, a nefarebnú (achromatickú) vrstvu, respektíve, vytvoria sa ich binárne masky (viď obrázok 3.6). Keďže som svoje riešenie obmedzil len na farebné pečiatky, s achromatickou vrstvou dokumentu sa ďalej nepracuje.



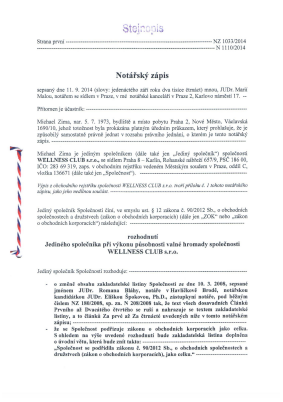
(a) Maska chromatickej vrstvy 1.



(b) Maska chromatickej vrstvy 2.



(c) Maska achromatickej vrstvy.



(d) Originál dokumentu.

Obr. 3.6: Ukážka binárnych masiek dokumentu odpovedajúcich farebným vrstvám a achromatickej vrstve.

Aby sa mohli vytvoriť farebné vrstvy stránky dokumentu, je treba rozdeliť farebné spektrum na skupiny. Na vytváranie takýchto farebných skupín som využil farebný priestor $\mathrm{YC_bC_r}$. Konkrétne, histogram uhlov farieb v rovine $\mathrm{C_b}$ $\mathrm{C_r}$ stránky dokumentu. Pre každý pixel p stránky dokumentu sa vypočíta vzdialenosť r jeho farby od stredu roviny $\mathrm{C_b}$ $\mathrm{C_r}$ a uhol φ v nej (viď obrázok 3.7).

$$r = \sqrt{C_b(p)^2 + C_r(p)^2}, \tag{3.1a}$$

$$\varphi = \mathrm{atan2}(C_b(p), C_r(p)), \tag{3.1b}$$

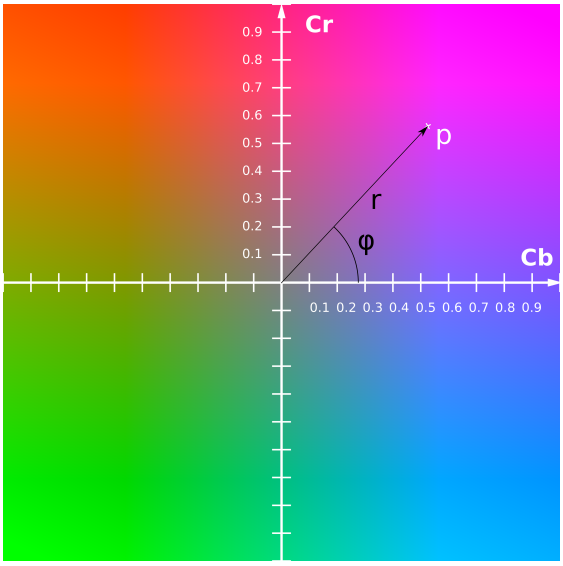
$$\mathrm{atan2} = \begin{cases} \arctan(\frac{y}{x}), & \text{ak } x > 0 \\ \arctan(\frac{y}{x}) + \pi, & \text{ak } y \geq 0,\ x < 0 \\ \arctan(\frac{y}{x}) - \pi, & \text{ak } y < 0,\ x < 0 \\ \frac{\pi}{2}, & y > 0,\ x = 0 \\ -\frac{\pi}{2}, & y < 0,\ x = 0 \\ \text{nedefinované} & x = 0,\ y = 0 \end{cases} \tag{3.1c}$$

$C_b(p)$ a $C_r(p)$ sú h-noty pixelu p v zložkách $\mathrm{C_b}$ respektíve $\mathrm{C_r}$. Pixely s hodnotou r menšou ako stanovená hranica sú označené za achromatické. Farebné skupiny sú potom určené na základe lokálnych miním v histograme. Ako je ale vidno z obrázka 3.8 originálny histogram má príliš veľa lokálnych miním a maxím. Preto pred hľadaním miním je histogram vyhladený, a to tak, že hodnota y' v bode x je vypočítaná ako priemer hodnôt z okolia bodu x. Formálne:

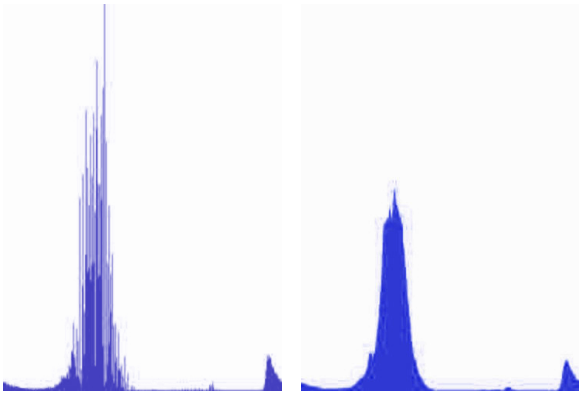
$$y' = \frac{1}{2n} \sum_{i=0}^{n-1} (f(x+n) + f(x-n)), \tag{3.2}$$

kde $f(x)$ je hodnota v bode x v pôvodnom histograme. Vyhladenie prebiaha viackrát, pričom sa parameter n mení. Jedna farebná skupina je potom ohraničená lokálnymi minimami histogramu.

Keďže aj vo farebných grafických prvkoch sa môžu nachádzať achromatické pixely, je stránka dokumentu pred vyššie popísaným rozdeľovaním na vrstvy jemne rozmazaná a po rozdelení prebehne redukcia šumu.



Obr. 3.7: Bod p v rovine C_bC_r. [Zdroj obrázka roviny C_bC_r: [59]]



(a) Histogram dokumentu. Na osi x sú hodnoty φ a na osi y je počet pixelov dokumentu s daným φ.

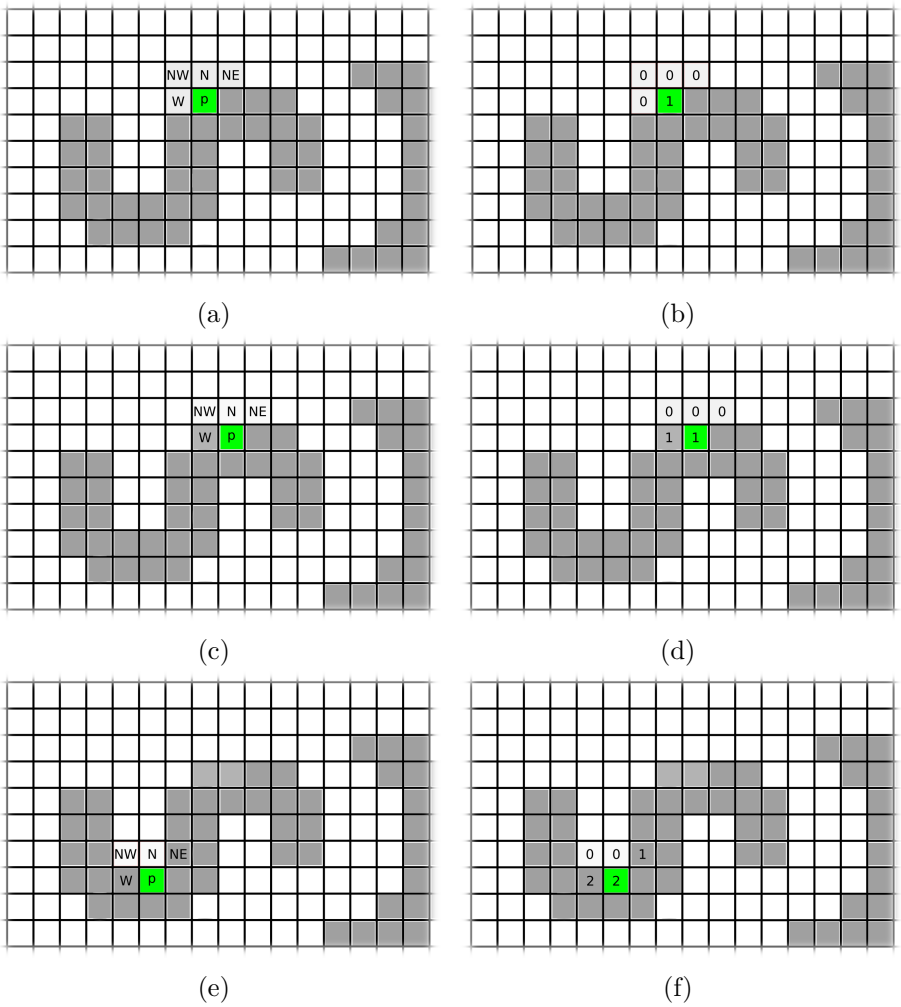
(b) Vyhladený histogram.

Obr. 3.8: Ukážka histogramov pre dokument z obrázka 3.6.

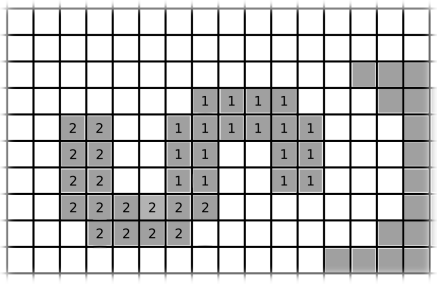
3.4.2 Extrakcia spojených komponentov

Ucelené grafické prvky, v ktorých z každeho pixelu prvku existuje cesta pixelami prvku do všetkých ostatných pixelov prvku, sú teda spojené, budem označovať ako spojené komponenty. Pre extrahovanie spojených komponentov z jednotlivých farebných vrstiev som využil základný algoritmus na označenie spojených komponentov uvedený v [24].

Prvým krokom daného algoritmu je počiatočné označovanie, počas ktorého sa pixelom spojených komponentov priradia štítky. Pre každý pixel binárnej masky farebnej vrstvy stránky sa určí jeho štítok na základe štítkov jeho štyroch susedov. A to tak, že ak má aktuálny pixel p v maske hodnotu 0, nedeje sa nič a pokračuje sa na ďalší pixel. Ak má pixel p v maske hodnotu 1 a všetci z jeho susedov majú hodnotu 0, označí sa pixel p novým štítkom (obrázky 3.9a a 3.9b). Ak práve jeden sused pixelu p má v binárnej maske hodnotu 1, jeho štítkom sa označí aj pixel p (obrázky 3.9c a 3.9d). Ak majú viacerí susedia pixelu p v maske hodnotu 1, pixelu p sa priradí jeden z ich štítkov (obrázky 3.9e a 3.9f). V tomto prípade je vhodné zaznamenať, že dané rôzne štítky susedov pixelu p sú ekvivalentné, čo sa využije v ďalšom kroku agoritmu. Susedia, podľa ktorých sa štítky určujú sú: severozápadný, severný, severovýchodný a západný sused. Smer prechádzania pixelov je zľava doprava a z hora dole, čím je zaistené, že každý zo susedov aktuálne prechádzaného pixelu už má správny štítok. Jednotlivé prípady a výsledok počiatočného označovania pre jeden spojený komponent sú graficky znázornené na obrázkoch 3.9 a 3.10.



Obr. 3.9: Ukážka priradenia štítku aktuálnemu pixelu. NW - severozápadný sused, N - severný sused, NE - severovýchodný sused, W - západný sused pixelu p.



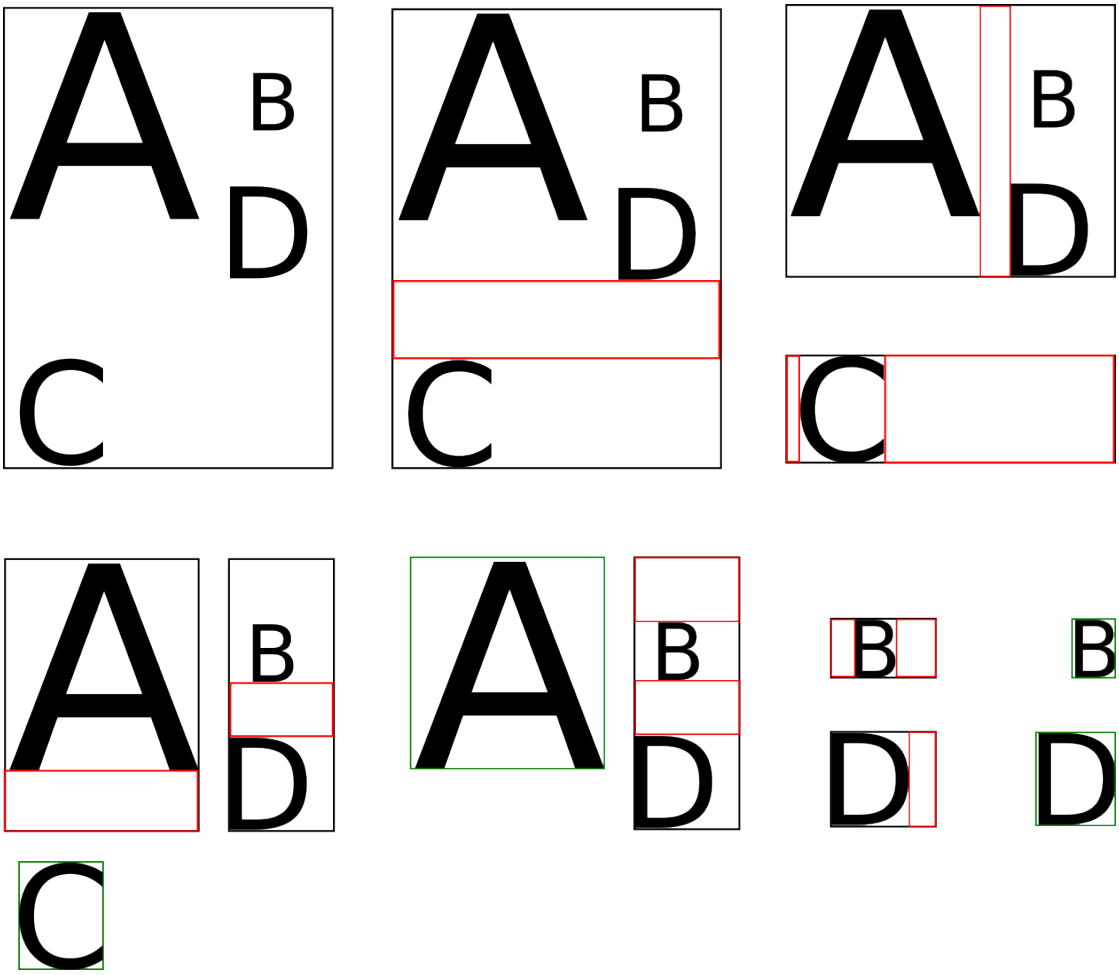
Obr. 3.10: Ukážka priradenia štítkov pixelom jedného spojeného komponentu. Štítky 1 a 2 sú ekvivalentné.

Jeden štítok sa zaručene neobjaví v dvoch rôznych spojených komponentoch, no v jednom spojenom komponente môžu byť pixele označené viacerými štítkami. V nasledujúcom kroku sa preto vytvoria skupiny ekvivalentných štítkov nasledovne. Vytvorí sa binárna matica L veľkosti $N \times N$, kde N je počet štítkov vo všetkých spojených komponentoch. Matica L ma v i-tom riadku a j-tom stĺpci hodnotu 1 ak i-ty štítok je ekvivalentný s j-tym. Relácia ekvivalencie je symetrická, reflexívna a tranzitívna. Matica L je tvorená tak, že symetrická je. Reflexívnosť získa jednoduchým pridanim jednotiek na diagonálu, pričom sa význam matice nezmení, keďže každý štítok je sám so sebou ekvivalentný. Pre doplnenie tranzivity uvádza [24] nasledujúci algoritmus, ktorý je známi ako Floyd – Warshallov algoritmus:

```
for i = 1 to n
  if L[i,j] = 1 then
    for k = 1 to n
      L[i,k] = L[i,k] OR L[j,k];
```

Nakoniec, prechodom oštítkovanými pixelami stránky dokumentu, ktorých ekvivalentné štítky sa nahradia zástupcom skupiny ekvivalentých štítkov sa získajú jednotlivé spojené komponenty obrázku.

Pri veľkom počte štítkov môže matica L nadobúdať príliš veľké rozmery a jej spracovanie može trvať dlho, pretože Floyd – Warshallovho algoritmus má časovú zložitosť $\mathcal{O}(n^3)$. Preto sa pred získavaním spojených komponentov farebné vrstvy stránky dokumentu rozdelia na menšie celky, z ktorých sa následne vyberú spojené komponenty. Na rozdelenie farebných vrstiev som použil rekurzívny algorimtus XY-Cut[30], ktorý hľadá miesta, v ktorých by sa dal obrázok rozdeliť. Ak existuje skupina riadkov alebo stĺpcov, v ktorých sa nevyskytujú pixele popredia, odstráni ich a v danom mieste dokument rozdelí. Algoritmu som pridal jeden parameter, ktorý určuje, koľko minimálne susedných riadkov, respektíve stĺpcov, musí byť bez pixelov popredia, aby sa v danom mieste mohol obrázok rozdeliť. Vďaka tomu sa potom zbytočne neoddeľujú grafické prvky blízko pri sebe, ako sú napríklad písmena v slove. Algoritmus končí, keď už nemá kde rozdeliť obrázok. Ukážka behu XY-Cut algoritmu je na obrázku 3.11



Obr. 3.11: Ukážka behu XY-Cut algoritmu. Čierny rám reprezentuje okraj vstupného obrázku. Červený rám ohraničuje miesto rozdelenia. Zelený rám obkolesuje výsledok rekurzie.

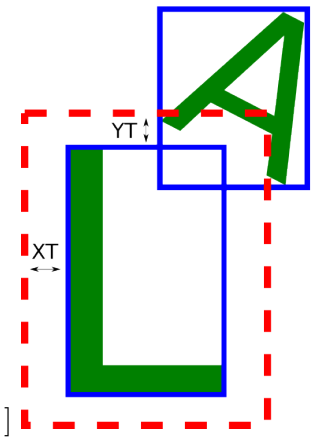
3.4.3 Vytváranie kandidátov

Nakoľko je pečiatka málokedy len jeden spojený komponent, v poslednom kroku segmentácie sa zo spojených komponentov vytvoria kandidátske oblasti zoskupovaním spojených komponentov. Ukažka je na obrázku 3.13. Pre každý spojený komponent platí, že ak nie je súčasťou žiadneho kándidáta, vytvorí sa nový kandidát a daný spojený komponent sa stane jeho súčasťou. Kandidát K_1, ktorého je aktuálny spojený komponent SK_1 súčasťou, k sebe pripojí spojený komponent SK_2, ak spĺňa obe podmienky pripojenia. Ak spojený komponent SK_2 je súčasťou kandidáta K_2, kandidáti K_1 a K_2 sa zlúčia.

Podmienkami na pripojenie sú:

1. dostatočná blízkosť na ploche stránky dokumentu
2. dostatočná blízkosť vo farebnom priestore.

Spojené komponenty K_1, K_2 sú dostatočne blízko na ploche stránky, ak aspoň jeden z rohov ohraničujúceho obdĺžnika spojeného komponentu SK_1 je vnútri ohraničujúceho obdĺžnika spojeného komponentu SK_2, rozšíreného o tolerančnú odchylku, viď obrázok 3.12. Kandidátov K_1, K_2 považujem za dostatočne blízke objekty, ak existuje dvojica spojených komponentov $SK_i \in K_1$, $SK_j \in K_2$ tak, že spojené komponenty SK_i, SK_j sú dostatočne blízko. Tento prístup nie je taký presný ako keby som meral vzdialenosti medzi pixelami v spojených komponentoch, je však rýchlejší pretože spojených komponentov je značne menej ako pixelov, ktoré ich tvoria.



Obr. 3.12: Ukážka spojených komponentov, ktoré sú dostatočne blízko. XT – tolerančná odchýlka v osi X, YT – tolerančná odchýlka v osi Y.

Na určenie toho, či sú objekty dostatočne blízko vo farebnom priestore a sú teda dostatočne farebne podobné využívam histogram farieb, tentokrát ale vo farebnom priestore HSV, pretože mi v ňom experimentálne vyšli v tejto časti lepšie výsledky ako v priestore YC_bC_r. Kandidát K_1 a kandidát K_2 (respektíve spojený komponent SK) sú farebne dostatočne podobné, ak priemerná hodnota zložky H kandidáta K_2 (respektíve spojeného komponentu SK) je v intervale $(\overline{H} - C_1 \times \sigma - C_2, \overline{H} + C_1 \times \sigma + C_2)$, kde $\overline{H}$ je priemerná hodnota hodnôt zložky H kandidáta K_1, σ je štadardná odchýlka hodnôt zložky H kandidáta K_1 a C_1, C_2 sú konštanty, ktoré som po experimentoch zvolil ako $C_1 = 1.9$, $C_2 = 1$. Podotýkam, že nakoľko je zložka H uhol, hodnoty 359 a 0 sú po sebe nasledujúce hodnoty.

(a) Spojené komponenty dokumentu. (b) Kandidáti dokumentu. (c) pôvodný dokument

Obr. 3.13: Ukážka zoskupovania spojených dokumentov do kandidátov.

3.5 Výpočet atribútov

Počas segmentácie stránky dokumentu sa vytvorila množina kandidátov, z ktorej boli odstránení kandídati, ktorí mali extrémnu (príliš veľkú alebo príliš malú) šírku alebo výšku. Na to, aby sa dalo určiť či kandidát je alebo nie je pečiatkou je potrebné poznať jeho vlastnosti, respektíve atribúty.

Vstupom pre výpočet niektorých atribútov je kandidát, ktorý je prevedený do odtieňov šedej a normalizovaný pomocou min-max normalizácie [29] podľa vzťahu 3.17. V navrhnutom riešení sa využíva vyše 70 atribútov. Medzi atribúty, ktoré som navrhol patria:

- *Počet spojených komponentov*, ktoré kandidát obsahuje.
- *Počet vnorených spojených komponentov.* Za vnorený spojený komponent považujem spojený komponent, ktorého ohraničujúci obdĺžnik celý leží v ohraničujúcom obdĺžniku iného spojeného komponentu patriaceho kandidátovi.
- *Pomer vnorených spojených komponentov ku všetkým spojeným komponentom kandidáta.*
- *Súradnice stredu ohraničujúceho obdĺžnika relatívne k šírke a výške dokumentu.*
- *Pomer šírky kandidáta k šírke strany dokumentu.*
- *Pomer výšky kandidáta k výške strany dokumentu.*
- *Pomer obsahu strany dokumentu k obsahu ohraničujúceho obdĺžnika kandidáta.*

- *Obvod spojených komponentov kandidáta.*
- *Pomer šírky strany dokumentu ku obvodu spojených komponentov kandidáta.*
- *Pomer obsahu spojených komponentov kandidáta ku ich obvodu.*
- *Pomer obsahu ohraničujúceho obdĺžnika k obvodu spojených komponentov kandidáta.*
- *Štandardná odchýlka, rozptyl, modus a priemerný uhol otočenia najmenších ohraničujúcich obdĺžnikov spojených komponentov kandidáta.*
- *Uhol otočenia najmenšieho ohraničujúceho obdĺžnika kandidáta.*
- *Počet farebných tried kandidáta.*
- *Rozptyl a priemerná hodnota uhlu φ pixelov v rovine C_bC_r* počítaný podľa 3.1.
- *Rozptyl a priemerná hodnota zložky H* farby pixelov vo farebnom priestore HSV.
- *Pomer kratšej strany najmenšieho ohraničujúceho obdĺžnika kandidáta k šírke strany dokumentu.*
- *Pomer dlhšej strany najmenšieho ohraničujúceho obdĺžnika kandidáta k šírke strany dokumentu.*
- *Pomer obsahu strany dokumentu k obsahu najmenšieho ohraničujúceho obdĺžnika kandidáta.*
- *Pomer strán najmenšieho ohraničujúceho obdĺžnika kandidáta.*
- *Rozptyl, štandardná odchýlka a priemerná hodnota pixelov normalizovaného kandidáta v odtieňoch šedej po aplikovaní Sobelovho operátora [49]* ako ukazatele intenzity hrán.
- *Rozptyl, štandardná odchýlka a priemerná hodnota pixelov kandidáta po aplikovaní Fourierovej transformácie [12].* Fourierová transformácia prevádza obrázok z priestorovej domény do frekvenčnej domény. Nízke frekvencie znamenajú plynulejšie prechody naopak vysoké indikujú ostré prechody a ostrý obrázok.
- Huové momenty [22] počítané z normalizovaného kandidáta v odtieňoch šedej.

Nasledujúce atribúty som prebral z práce [18] alebo sú z nich odvodené. Časti atribútov nechávam anglický názov, pretože som nenašiel vhodný preklad.

- *Pomer výšky a šírky kandidáta.*
- *Počet výrazných vrcholov normalizovaného kandidáta v odtieňoch šedej..*
- *Počet výrazných vrcholov binárnej masky kandidáta.*

- *Hustota*, ako pomer obsahu spojených komponentov kandidáta k obsahu ohraničujúceho obdĺžnika kandidáta.

- *Rozptyl, štandardná odchýlka a priemer hodnôt pixelov kandidáta po prevedení do odtieňov šedej*, podľa vzťahu uvedeného v [36]:

$$Y(p) = 0.299 \times R(p) + 0.587 \times G(p) + 0.114 \times B(p), \tag{3.3}$$

  kde $Y(p)$ je hodnota pixelu v odtieňoch šedej a $R(p)$, $G(p)$, $B(p)$ sú súradnice pixelu p vo farebnom priestore RGB.

- *Rozptyl, štandardná odchýlka a priemer intenzity pixelov kandidáta.* Intenzitu I pixelu p definujem ako

$$I(p) = \frac{R(p) + G(p) + B(p)}{3}, \tag{3.4}$$

  kde $R(p)$, $G(p)$ a $B(p)$ sú súradnice pixelu p vo farebnom priestore RGB.

- *Michelsonov kontrast M [41] a RMS kontrast [41] pre maticu odtieňov šedej a pre maticu intenzity pixelov kandidáta*, podľa vzťahov:

$$M = \frac{MAT_{max} - MAT_{min}}{MAT_{max} + MAT_{min}}, \tag{3.5a}$$

$$RMS = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}, \tag{3.5b}$$

  kde MAT_{max} je maximálna hodnota matice, MAT_{min} je minimálna hodnota matice, x_i je i-ta hodnota matice, $\bar{x}$ je priemerná hodnota matice a n je počet pixelov v matici.

- *Pomery priemernej hodnoty pixelov v odtieňoch šedej s Michelsonovým konrastom a RMS kontrastom pre obe matice.*

- *Pomery priemernej hodnoty intenzity pixelov s Michelsonovým konrastom a RMS kontrastom pre obe matice.*

- *Miera podobnosti kandidáta s dokonalým štvorcom S*

$$S = (\frac{min(\sqrt{A_p}, \frac{P}{4})}{max(\sqrt{A_p}, \frac{P}{4})})^2, \tag{3.6}$$

  kde A_p je obsah spojených komponentov kandidáta vyjadrený v ako súčet pixelov, ktoré im prislúchajú a P je obvod kandidáta resp. jeho spojených komponentov.

- *Okrúhlosť R*

$$R = \frac{R_1 + R_2 + R_3}{3} \tag{3.7a}$$

$$R_1 = \left|1 - \frac{M_2 - M_1}{max(M_1, M_2)}\right| \tag{3.7b}$$

$$R_2 = \left|1 - \frac{\left|\frac{\pi\left(\frac{M_1+M_2}{2}\right)^2}{2} - A_p\right|}{A_p}\right| \tag{3.7c}$$

$$R_3 = \left|1 - \frac{|\pi max(M_1, M_2) - P|}{P}\right|, \tag{3.7d}$$

  kde M_2 je dĺžka dlhšej strany a M_1 je dĺžka kratšej strany najmenšieho ohraničujúceho obdĺžnika kandidáta.

- *Miery podobnosti s kruhom S_{cir}, štvorcom S_{sqr}, elipsou S_{eli}, obdĺžnikom S_{rct}, trojuholníkom S_{tri} a kosoštvorcom S_{rho}*

$$S_{cir} = R_s - R_l \tag{3.8a}$$

$$S_{sqr} = \frac{A_p}{4R_s^2} \tag{3.8b}$$

$$S_{eli} = \frac{A_p}{\pi R_s R_l} \tag{3.8c}$$

$$S_{rct} = \frac{A_p}{4R_S\sqrt{R_l^2 - R_s^2}} \tag{3.8d}$$

$$S_{tri} = \frac{A_p\sqrt{3}}{(R_l + R_s)^2} \tag{3.8e}$$

$$S_{rho} = \frac{A_p\sqrt{R_l^2 - R_s^2}}{2R_l^2 R_s}, \tag{3.8f}$$

  kde R_l je dĺžka dlhšej strany a R_s je dĺžka kratšej strany najmenšieho ohraničujúceho obdĺžnika kandidáta.

Ďalšie 4 atribúty sa počítajú z matice, ktorá vznikne tak, že sa pre jednotlivé odtiene šedej vypočíta, koľkokrát má pixel s hodnotou p medzi svojimi blízkymi horizontálnymi susedmi pixel s hodnotou q. Hodnota na p-tom riadku a q-tom stĺpci je potom daný súčet. V [18] sa táto matica nazýva *Gray-Level Coocurrence Matrix* a označuje sa GLCM. Matica GLCM sa teda vytvorí podľa vzťahu:

$$GLCM_{\Delta x,\Delta y}(p,q) = \sum_{x=1}^{M}\sum_{y=1}^{N}\begin{cases} 1, & \text{ak } MAT(x,y) = p \wedge MAT(x+\Delta x, y+\Delta y) = q \\ 0, & \text{inak} \end{cases} \tag{3.9}$$

Matica MAT je matica obrázku kandidáta v odtieňoch šedej.

- *Kontrast matice GLCM* sa vypočíta ako

$$GLCM_C = \sum_p \sum_q (p-1)^2 GLCM(p,q) \tag{3.10}$$

- *Energia matice GLCM* sa počíta pomocou vzťahu

$$GLCM_E = \sum_p \sum_q GLCM(p,q)^2 \tag{3.11}$$

- Pre vypočítanie *Homogenity matice GLCM* sa používa vzťah

$$GLCM_H = \sum_p \sum_q \frac{GLCM(p,q)}{1+|p-q|} \tag{3.12}$$

- *Korelácia matice GLCM* sa spočíta ako

$$GLCM_{Corr} = \sum_p \sum_q \frac{(p-\mu_p)(q-\mu_q)GLCM(p,q)}{\sigma_p \sigma_q}, \tag{3.13}$$

  kde μ_p, σ_p a μ_q, σ_q sú priemer a rozptyl hodnôt v riadku p respektíce stĺpci q.

A nakoniec momentové atribúty počítané z normalizovaného kandidáta v odtieňoch šedej.

- *ellipticity* E_I

$$I_1 = \frac{\mu_{20}\mu_{02} - \mu_{11}^2}{\mu_{00}^4} \tag{3.14a}$$

$$E_I = \begin{cases} 16\pi^2 I_1, & \text{ak } I_1 < \frac{1}{16\pi^2} \\ \frac{1}{16\pi^2 I_1}, & \text{inak} \end{cases} \tag{3.14b}$$

  kde $\mu_{p,q}$ sú centrálne momenty druhého rádu [14]

- *circular variance* C_{var}, ako ďalšia podobnosť s kruhom

$$\mu = \frac{1}{N}\sum_{i=1}^{N} p_i \tag{3.15a}$$

$$\mu_r = \frac{1}{N}\sum_{i=1}^{N} |p_i - \mu| \tag{3.15b}$$

$$C_{var} = \frac{1}{N\mu_r^2}\sum_i (|p_i - \mu| - \mu_r^2), \tag{3.15c}$$

  kde p_i je hodnota i-teho pixelu normalizovaného kandidáta v odtieňoch šedej a N je počet pixelov.

- *triangularity* T_I, ktorá pre dokonalý trojuholník nadobúda hodnotu 1 a počíta sa pomocou vzťahu

$$T_I = \begin{cases} 108 I_1, & \text{ak } I_1 < \frac{1}{16\pi^2} \\ \frac{1}{108 I_1}, & \text{inak} \end{cases} \tag{3.16a}$$

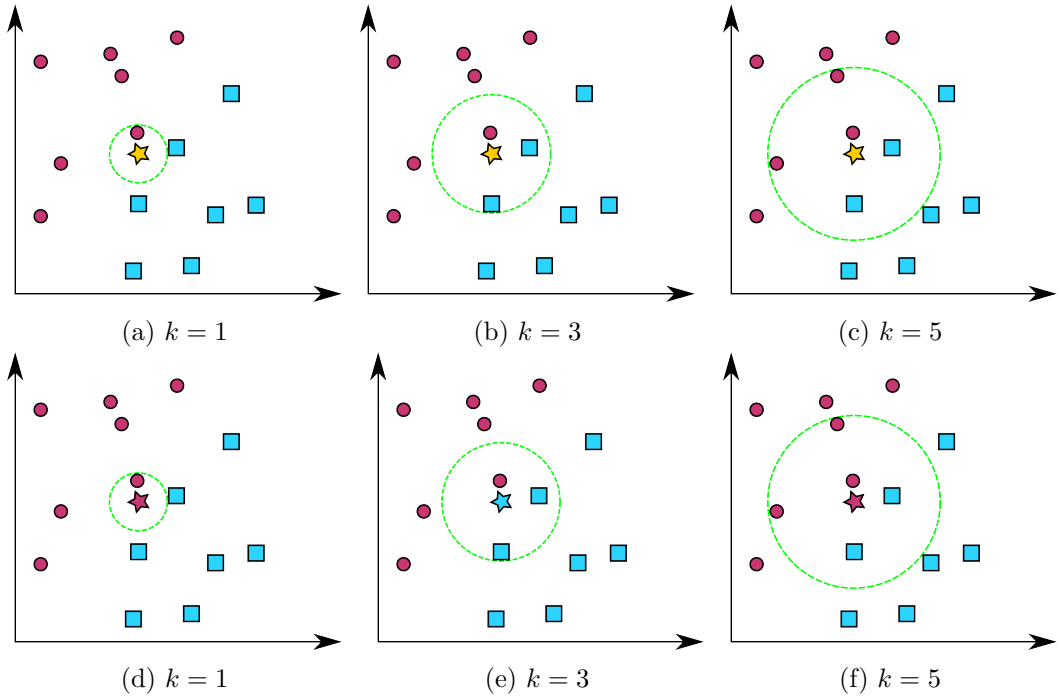
3.6 Klasifikácia

Keď už sú známe vlastnosti prvkov, ktoré algoritmus skúma, prichádza klasifikácia, v ktorej sa rozhodne o tom, či kandidát je pečiatkou alebo nie. Ako klasifikačný algoritmus som po dohode s vedúcim bakalárskej práce zvolil algoritmus k najbližších susedov (v angličtine k nearest neighbours, k-nn). Tento algoritmus zvolili aj autori [18]. Algoritmus k-nn funguje veľmi jednoducho. Počas trénovacej fázy algoritmu sa iba uložia hodnoty atribútov prvkov trénovacej množiny. Trénovacia množina je množina grafických prvkov, pre ktoré sa vypočitali hodnoty atribútov tak, ako je popísane v predchádzajúcej sekcií. Prvky trénovacej množiny sa uložia spolu s označením triedy, do ktorej patria. V tomto prípade to može byť trieda „pečiatka“ alebo trieda „iný grafický prvok“. Trieda je trénovacím prvkom priradená manuálne. Učiaca fáza prebieha predtým ako chceme použiť algoritmus na detekovanie pečiatky v neznámom dokumente a prebieha len raz. Počas pracovnej fázy je kandidát, rovnako ako prvky trénovacej množiny, považovaný za bod v n-rozmernom priestore, kde n je počet atribútov, ktoré sa vypočítali v predchádzajúcom kroku procesu. Súradnicami sú hodnoty daných atribútov. Kandidátovi sa priradí (predikuje) tá trieda, ktorá má najväčšie zastúpenie medzi najbližšími susedmi kandidáta. Počet susedov, ktorí sa berú do úvahy udáva parameter k. Definícia vzdialenosti môže byť ľubovolná, ja som použil Euklidovu vzdialenosť. Obrázok 3.14 znázorňuje, ako by pracoval algoritmus k-nn v dvojrozmernom priestore. Algoritmus má za úlohu priradiť žltej hviezde farbu skupiny, do ktorej patrí. Je vidieť, že pri rôznych hodnotách parametru k sa výsledky líšia, preto som pri testovaní skúmal vplyv tohoto parametru na výsledky predikcie. Výsledky experimentov sú v sekcií 5.1.

Atribúty môžu mať hodnoty v rozličných rozsahoch, čo spôsobuje, že jeden atríbút zaváži pri počítaní vzdialenosti viac ako iný. Preto je treba ešte pred učením normalizovať hodnoty atribútov trénovacej množiny ako aj hodnoty atribútov kandidátov počas pracovnej fázy. Pre normalizovanie som zvolil min-max normalizáciu [29], ktorá prepočítava hodnoty atribútov do nového rozsahu pomocou vzťahu:

$$v' = \frac{v - min_a}{max_a - min_a}(max_n - min_n) + min_n, \tag{3.17}$$

kde v' je nová hodnota atribútu a , v je pôvodná hodnota atribútu a, max_a je najväčšia z hodnôt atribútov a, min_a je najmenšia z hodnôt atribútov a, max_n a min_n sú medze rozsahu, do ktorého sa hodnoty prevádzajú. Trénovacia množina sa počas učenia uložila normalizovaná, preto v pracovnej fáze sa použijú hodnoty max_a a min_a vypočítané z hodnôt trénovacej množiny pred normalizáciou. V dôsledku toho sa môže stať, že atribút a neznámeho kandidáta bude mimo rozsahu, to však v konečnom dôsledku nevadí, pretože váhy atribútov ostanú rovnaké.



(a) $k = 1$ (b) $k = 3$ (c) $k = 5$

(d) $k = 1$ (e) $k = 3$ (f) $k = 5$

Obr. 3.14: Ukážka algoritmu k-nn pre určenie farby hviezdy. Zelený kruh znázorňuje okolie, v ktorom sa nachádza k najbližších susedov hviezdy (neznámeho prvku).

3.7 Úprava extrahovaných pečiatok

Nakoniec sa grafické prvky stránky dokumentu, ktoré boli počas klasifikácie prehlásené za pečiatky upravia, aby sa zvýšila ich obrazová kvalita. Ako prvé sa zvýši kontrast pečiatky. Pečiatka sa prevedie do farebného priestoru $\mathrm{YC_bC_r}$. Hodnoty zložky Y pixelov pečiatky sa normalizujú pomocou vzťahu 3.17 min-max normalizácie [29] do celého rozsahu zložky Y v priestor $\mathrm{YC_bC_r}$. Následne sa pečiatka prevedie späť do priestoru RGB.

Na odstránenie šumu, ktorý by mohol obrázok pečiatky obsahovať som použil algoritmus „Non-Local Means Denoising“ [4], ktorý odstraňuje šum počítaním váženého priemeru farby najpodobnejších pixelov nasledovne. Pre každý pixel p sa vypočítajú nové honoty v zložkách R, G, B pomocou nasledujúceho vzťahu:

$$u_i'(p) = \frac{1}{C(p)} \sum_{q \in B(p,r))} u_i(q) w(p,q), \tag{3.18}$$

kde $u_i(p)'$ je nová hodnota pixelu p v zložke $i \in \{R, G, B\}$, $u_i(q)$ je hodnota pixelu q v zložke i, $C(p)$ je normalizačný faktor a počíta sa pomocou vzťahu 3.19, w je váha, ktorá sa počíta podľa 3.21 a $B(p,r)$ je okolie bodu p o veľkosti $2r+1$ x $2r+1$ pixelov.

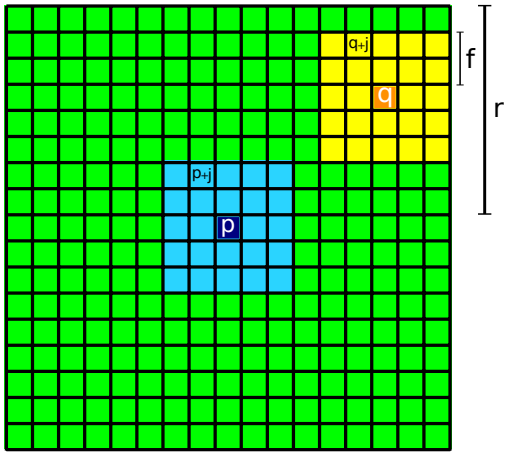
Parameter r teda určuje veľkosť okolia, z ktorého sa priemer počíta (viď obrázok 3.15).

$$C(p) = \sum_{q \in B(p,r)} w(p,q) \tag{3.19}$$

$$d^2 := d^2(B(p,f), B(q,f)) = \frac{1}{3(2f+1)^2} \sum_{i \in \{R,G,B\}} \sum_{j \in B(0,f)} (u_i(p+j) - u_i(q+j))^2 \tag{3.20}$$

$$w(p,q) = e^{-\frac{max(d^2 - 2\sigma^2, 0)}{h^2}} \tag{3.21}$$

Parameter f udáva veľkosť okolia, z ktorého sa počíta vzdialenosť (podobnosť) d^2 medzi pixelami p a q (viď obrázok 3.15). Hodnota parametru h sa určí pomocou tabuľky (tiež uvedenej v [4]), v závislosti od štandardnej odchýlky šumu σ. Tento algoritmus som využil v podobe funkcie implementovanej v knižnici OpenCV [26].

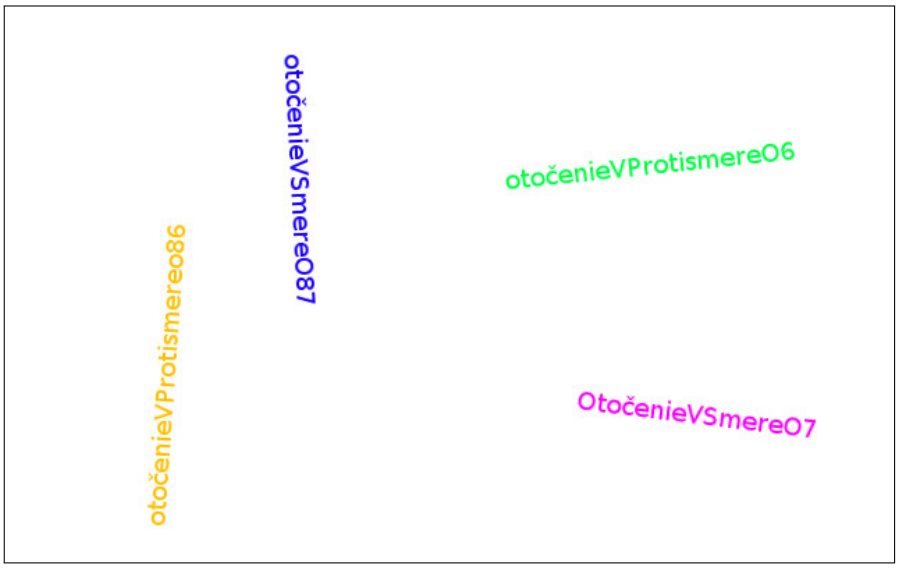


Obr. 3.15: Ukážka okolí pixelov, ktoré sa využívajú na odstránenie šumu. Zelený obdĺžnik reprezentuje $B(p,r)$, modrý $B(p,f)$ a zelený $B(q,f)$.

Nakoniec, sa pečiatky ešte otočia do základnej polohy. Pôvodne som mal snahu využiť na to Houghovú transformáciu [10], ale zistil som že problém rotácie do základnej polohy pečiatok nie je taký triviálny ako sa môže zdať, preto po dohode s vedúcim bakalárskej práce som zvolil menej účinný, no jednoduchší spôsob. Z pozorovaní vyplýva, že keď je pečiatka v základnej polohe, aj jej najmenší ohraničujúci obdĺžnik je v základnej polohe. Toto pozorovanie však platí len na pečiatky, ktoré majú jeden rozmer väčší ako druhý, preto rotácia prebieha len na pečiatkách, ktoré majú dlhšiu stranu aspoň 1,5krát dlhšiu ako druhú stranu. Za účelom rotácie pečiatky do základnej polohy sa najskôr nájde najmenší ohraničujúci obdĺžnik pečiatky. Pečiatka sa potom otočí tak, aby sa jeho najmenší ohraničujúci obdĺžnik dostal do základnej polohy. Z pozorovaní taktiež vyplýva, že pečiatka je zvyčajne širšia ako vyššia, preto je otočená vždy do takej polohy, aby jej dlhšia strana bola rovnobežná s osou X viď obrázok 3.17. Týmto celý proces extrakcie pečiatky a jej následnej úpravy končí.

(a) Extrahovaná pečiatka. (b) Extrahovaná pečiatka po zvyšení kontrastu.

Obr. 3.16: Ukážka zvýšenia kontrastu pečiatky.



(a) Testovací obrázok.

(b) $\alpha = -86°$ (c) $\alpha = -6°$ (d) $\alpha = 87°$ (e) $\alpha = 7°$

Obr. 3.17: Ukážka výsledkov po zvýšení kontrastu a rotácií textu do základnej polohy. Jednotlivé texty boli zo základnej polohy otočené v smere hodinových ručičiek o uhol α.

(a) Extrahovaná pečiatka.

(b) Extrahovaná pečiatka so zvýšeným kontrastom, po redukcií šumu a otočená do základnej polohy.

Obr. 3.18: Ukážka pečiatky pred a po upravení.

Kapitola 4

Implementácia

4.1 Použité technológie

4.1.1 Java

Keďže implementácia mojej bakalárskej práce bola plánovaná ako modul do programu na spracovanie dát Surmon [51], ktorý je písaný v programovacom jazyku Java, implementoval som aj ja v tomto jazyku. Jazyk Java je populárny, objektovo orientovaný, vyšší jazyk. Je to jazyk, ktorý sa interpretuje a vďaka tomu sú aplikácie prenositeľné medzi operačnými systémami. Ďalšie informácie o tomto programovacom jazyku na [60].

4.1.2 OpenCv

V implementácií vo veľkej miere využívam knižnicu na spracovanie obrazu OpenCV vo verzií 3.0. Názov OpenCV je skratka pre *open source computer vision*. Táto knižnica je voľne dostupná pre akademické i komerčné účely. Je použiteľná s operačnými systémami Windows, Linux, Mac Os, iOS a Android a má rozhranie pre jazyky C, C++, Python a Java. Dokumentácia je však v prevažnej väčšine pre jazyky C, C++ alebo Python. Na svojej webovej stránke `www.opencv.org` ďalej uvádzajú, že knižnica OpenCV bola navrhnutá tak, aby pracovala efektívne a mohla byť využitá v aplikáciach v reálnom čase.

4.2 Štruktúra implementácie

Navrhnuté riešenie som implementoval ako samostatnú aplikáciu, ktorá slúži na extrakciu pečiatok z naskenovaných dokumentov, ale aj na testovanie a učenie algoritmu k najbližších susedov. Taktiež som riešenie implentoval ako modul do nástroja na spracovanie dat Surmon [51]. Pri implementácii som využil trojvrstvovú architektúru.

4.2.1 Prezentačná vrstva

Táto vrstva slúži na zobrazovanie a ukladanie výsledkov, medzivýsledkov a pomocných prvkov jednotlivých fáz ako aj celého algoritmu. Obsahuje jedinú triedu Render, ktorej metódy umožnujú ukladať a zobrazovať kandidátov, farebné histogrami, stránku dokumentu s farebne odlíšenými spojenými komponentami či kandidátmi a ďalšie.

4.2.2 Biznis vrstva

V tejto vrstve sú triedy, ktoré tvoria jadro aplikácie. Triedy, ktoré obsahuje by sa dali rozdeliť do dvoch kategórií. Prvou kategóriou sú triedy, ktoré reprezentujú prvky dokumentu, medzi ktoré patrí strana dokumentu, spojený komponent, kandidát a pečiatka. Druhú skupinu tvoria triedy, ktoré majú na starosti jednotlivé fázy algoritmu ako je segmentácia, výpočet atribútov, klasifikácia a úprava pečiatok.

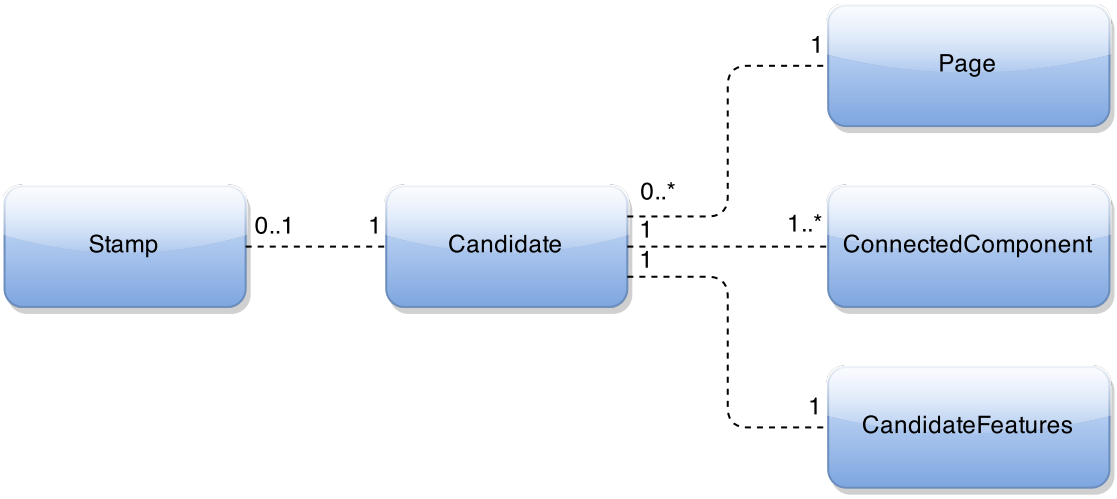
Trieda Page je trieda reprezentujúca stránku dokumentu a slúži iba ako obalovacia trieda, ktorá obsahuje maticu hodnôt pixelov vo farebnom priestore RGB.

Trieda ConnectedComponent reprezentuje spojený komponent. Má metódy na výpočet farebných histogramov a ich štatistických vlastností, ktoré sú využité v segmentačnej časti, rovnako ako metóda na určenie či je jeden spojený komponent dostatočne blízko iného spojeného komponentu na to, aby oba patrili jednému kandidátovi. Má uložené súradnice svojho ohraničujúceho obdĺžnika, pixele, ktoré mu prislúchajú, farebné štatistiky a neskôr kandidáta, do ktorého patrí.

Trieda Candidate reprezentuje kandidáta. Obsahuje metódy na pridanie spojeného komponentu ako aj zlúčenie sa s iným kandidátom. Tiež má metódy na výpočet farebných histogramov a ich štatistických vlastností a metódu na určenie či je spojený komponent prípadne jeho kandidát dostatočne blízko vo farebnom priestore k inému kandidátovi. Má zaznamenané spojené kompontenty, ktoré mu patria, súradnice ohraničujúceho obdĺžnika, farebné štatistiky a vypočítané atribúty.

Trieda CandidateFeatures je obalovacou triedou pre atribúty kandidáta.

Trieda Stamp reprezentuje pečiatku. Slúži ako obalovacia trieda. Má uloženú maticu pixelov obrázku pečiatky a odkaz na kadidáta, z ktorého vznikla.



Obr. 4.1: Ukážka vzťahov medzi triedami reprezentujúcimi prvky dokumentu.

Trieda Segmentation je zodpovedná za segmentáciu stránky dokumentu a vytvorenie kandidátov. Má jedinú verejnú metódu *getCandidates*, ktorej argumentom je strana dokumentu a návratovou hodnotou je zoznam kandidátov, ktoré strana obsahuje. Všetky ostatné metódy sú označené ako *protected*. Tieto metódy slúžia na jednotlivé podúlohy segmentácie.

Trieda CandidateFeaturesExtraction slúži na výpočet všetkých atribútov kandidáta.

Trieda PostProcessing je zodpovedná za úpravu extrahovanej pečiatky. Má metódy na zvýšenie kontrastu, odstránenie šumu a na rotáciu do základnej polohy.

Trieda ImgProcessing má implementované pomocné metódy, ktoré sú často použité vo viacerých fázach procesu ako napríklad vyhladenie histogramu alebo vypočítanie uhla φ v rovine C_bC_r.

Trieda StampExtraction je hlavnou triedou a spája časti procesu extrakcie pečiatok. Ďalej má metódy určené na testovanie ale aj metódy na vytvorenie trénovacej množiny pre algoritmus k-nn.

4.2.3 Dátová vrstva

Táto vrstva umožnuje načítavať a ukladať dáta. Neprebiehajú tu žiadne výpočty.

Trieda Loader obsahuje metódu na načítanie obrázku dokumentu ale aj metódu, pomocou ktorej sa zo súboru načíta trénovacia množina a hodnoty pre normalizáciu.

Trieda Logger umožnuje zapísať trénovaciu množinu do súboru.

Trieda Database zabezpečuje prácu s databázou Apache Derby 10.10 [52] a slúži na uloženie a prácu s trénovacou množinou.

Trieda Settings nemá žiadnu metódu a slúži len na nastavenie parametrov procesu.

Ďalší popis metód a tried sa nachádza v dokumentácií na priloženom CD.

4.3 Integrácia do programu Surmon

Program Surmon [51] je nástroj na spracovanie dát. Dáta sa spracujú procesom, ktorý užívateľ vytvorí spájaním modulov. V prílohe C.1 je príklad procesu, ktorý načíta obrázok, zobrazí, otočí o zadaný uhol a znova zobrazí. Pri implementácii modulu pre Surmon som sa riadil architektúrou Model-View-Controller (MVC) [27], ktorá sa zkladá z 3 základných prvkov:

- *Model* je základom aplikácie. Preberá informácie a dáta, ktoré mu poskytuje *Controller*, prebiehajú v ňom výpočty a výstupy posiela do *View*. *Model* taktiež uchováva informácie potrebné pre aplikáciu. Sem patrí moja dátová a biznis vrstva.
- *View* slúži ako užívateľské rozhranie aplikácie. Sem patria triedy Surmonu, ktoré tvora užívateľské prostredie aplikácie.
- *Controller* zabezpečuje preberanie dát z *View* a predáva ich modelu. Ako *Controller* slúžia triedy Surmonu, ktoré sa starajú o prepájanie užívateľského prostredia aplikácie s aplikáciou a o prepájanie modulov medzi sebou.

Pre vytvorenie modulu do Surmonu mi stačilo vytvoriť dve nové triedy. Trieda StampExtractionComponent reprezentuje grafickú podobu modulu, pomocou ktorej môže užívateľ zaradiť môj modul do procesu. Je súčasťou *View* celého programu a patrí do prezentačnej vrtvy. Trieda StampExtractionBlock definuje vstupné a výstupné rozhranie modulu, metódou *activate()* preberie obrázok dokumentu zo vstupu, predá ho metóde *extractStampsFromDocument* z triedy StampExtraction a zoznam obrázkov, ktorý je návratovou hodnotou tejto funkcie, predá na výstup. Všetko ostatné už zabezpečí Surmon. Túto triedu som zaradil do biznis vrstvy a v rámci MVC patrí do modelu.

Trieda Logger, Database ani Render sa nevyužívajú, preto som ich z modulu odstránil.

Kapitola 5

Vyhodnotenie

Na testovanie a vytvorenie trénovacej množiny som použil 210 stránok rôznych dokumentov z portálu www.justice.cz obsahujúcich spolu 347 farebných pečiatok rôznej kvality a 55 stránok dokumentov, ktoré neobsahovali pečiatky, ale obsahovali iné grafické prvky ako napríklad obrázky. Osobitne som zaznamenal výsledky segmentácie a klasifikacie a nakoniec vyhodnotil celý algoritmus.

5.1 Vyhodnotenie segmentačnej časti

Po segmentácií pečiatok vznikli kandidáti, ktorí reprezentujú celú pečiatku, časť pečiatky, pečiatku spolu s ďalším grafickým prvkom a podobne. Za učelom vyhodnotenia segmentácie som vytvoril nasledujúce kategórie:

- skupinu A tvoria pečiatky, ktorých kandidáti reprezentujú celú pečiatku bez väčších vizuálnych strát a iné grafické prvky, ktoré tiež patria kandidátovi zaberajú minimálnu časť.
- skupinu B tvoria pečiatky, ktoré sú viaceré súčasťou jedného kadidáta.
- skupinu C tvoria pečiatky, ktorých kandidáti reprezentujú pečiatku (alebo viac pečiatok), ale obsahujú aj časť iných grafických prvkov.
- skupinu D tvoria pečiatky, ktorých kandidáti reprezentujú len časť pečiatky.
- skupinu E tvoria pečiatky, ktorých kandidáti obsahujú pečiatku, ale ich veľkú časť zaberajú iné grafické prvky.
- skupinu F tvoria pečiatky, ktorých kandidáti reprezentujú len časť pečiatky, ktorá nenesie žiadne informácie.
- skupinu G tvoria pečiatky, ktorým počas segmentácie nevzikli kadidáti.

Kandidáti obsahujúci pečiatky skupiny B vznikli tak, že pečiatky, ktoré kandidát obsahuje boli odtlačené veľmi blízko seba alebo sa dokonca prekrývajú a preto boli zlúčené do jedneho kandidáta. Rovnako sa mohlo stať, že pečiatky spája iný grafický prvok rovnakej farby ako je napríklad podpis.

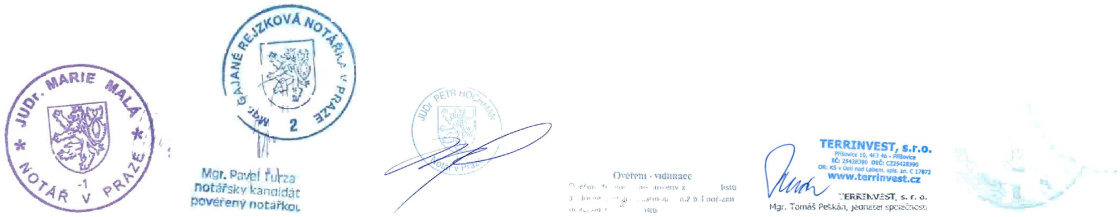
Ak sa pečiatka nachádzala príliš blízko grafického prvku rovnakej farby, algoritmus ich zlúčil do jedného kandidáta a pečiatka patrí do skupiny C.

Pri rozdelovaní na farebné vrsty sa mohlo stať, že časť pečiatky sa zaradila do inej vrstve ako zvyšok pečiatky. Preto sa spojené komponenty následne spájali dohromady na základe farebnej a plošnej informácie. Mohlo sa ale stať, že časť pečiatky, ktorá sa oddelila má natoľko inú farbu, že sa nespojila zo zvyškom pečiatky. Mohlo sa taktiež stať, že časť pečiatky bola označená za achromatickú a do ďalšej časti sa ani nedostala. Dokument niekedy obsahuje len časť pečiatky alebo pečiatku, ktorej sa niektoré časti neodtlačili a kandidát teda nemá ako vytvoriť kompletnú pečiatku. Pečiatky, ktorých kandidáti vznikli týmito spôsobmi patria do skupiny D.

Skupinu E tvoria pečiatky, ktorých kandidáti vznikli tak, že boli zlúčené s veľkým grafickým prvkom rovnakej farby alebo sa zlúčili s malým grafickým prvkom rovnakej farby, ktorý bol však blízko ďalšieho grafického prvku rovnakej farby, ktorý sa pridal kandidátovi a tak ďalej.

Skupina F je podobná skupine D, avšak časti pečiatky, ktoré sa extrahovali nesú príliš málo informácie, aby sa dali ďalej na niečo použiť.

Pečiatky, ktoré neboli extrahované vôbec sa radia do skupiny G. Tieto pečiatky mohli byť označené za achromatické alebo sú priveľmi slabo viditelné.



(a) A (b) B (c) C (d) D (e) E (f) F

Obr. 5.1: Ukážka kandidátov v jednotlivých kvalitatívnych skupinách.

Tabuľka 5.1 vyjadruje koľko percent pečiatok sa po segmentácií zaradilo do jednotlivých skupín.

skupina:	A	B	C	D	E	F	G
podiel v %:	58.8	14.4	5.76	11.53	6.92	1.44	1.15

Tabuľka 5.1: Výsledky segmentácie.

Za úspešné výstupy segmentačej časti považujem kandidátov patriacich do prvých troch prípadne štyroch skupín, čo v súčte dáva úspešnosť medzi 78,96 % a 90,49 %. Uvádzam rozmedzie namiesto presnej hodnoty z dôvodu, že hranice medzi jednotlivými

skupinami nie sú presne definované. Zaraďovanie kandidátov do skupín A – G prebiehalo manuálne. Úspešnosť segmentácie nie je taká vysoká ako by bolo žiadúce, znižujú ju však aj dokumenty, ktoré obsahujú veľmi slabo viditeľné či zle odtlačené pečiatky, pečiatky, ktoré sa prekrývajú a podobne. Segmentácia, tak ako je implenetovaná, má veľa parametrov a ďalšie testovanie vhodných hodnôt môže viesť k zvýšeniu jej úspešnosti.

5.2 Vyhodnotenie kvality atribútov

Atribúty kandidátov som pomocou progamu Rapid Miner [44] analyzoval s využítím algoritmov, ktoré atribútom priradia váhu podľa ich významnosti. Prvým algoritmom, pomocou ktorého som počítal váhy atribútom, je algoritmus relief [42]. Podľa [58] patrí tento algoritmus medzi najúspešnejšie algoritmi pre posudzovanie kvality atribútov. Relief hodnotí kvalitu atribútu tak, že z trénovacej množiny vyberie vzorku prvkov a pre každý jej prvok P, porovnáva hodnotu daného atribútu jeho najbližšieho suseda z rovnakej triedy s hodnotou atribútu jeho najbližšieho suseda z opačnej triedy. V tomto prípade sú to teda triedy „pečiatka“ alebo trieda „iný grafický prvok“.

Druhý algoritmus, ktorý som využil na posudzovanie významnosti atribútov je založený na počítaní korelácie medzi atribútmi [57]. Korelácia je číslo v intervale $< -1, 1 >$, ktoré udáva zavislosť medzi dvomi atribútmi. Ak by sa korelácia medzi atribútmi a_1 a a_2 rovnala 1, znamenalo by to, že nízke hodnototy atributú a_1 majú súvislosť z nízkymi hodnotami atribútu a_2 a rovnako tak aj vysoké hodnoty. Korelácia -1, hovorí o tom že nízke hodnoty atribútu a_1 majú súvislosť s vysokými hodnotami atribútu a_2 a opačne.

Výsledky tejto analýzy pre oba algoritmi sa nachádzajú v prílohe D a sú zoradné od najvýznamnejšieho atribútu po najmenej významný atribút.

5.3 Vyhodnotenie klasifikácie

Na obrázku 3.14 je znázornené ako sa výsledky algoritmu k-nn môžu meniť v závislosti na parametri k. Preto som vplyv tohoto parametru testoval pre hodnoty k od jedna do päť. Do klasifikácie neboli zaradené všetky vypočítané atribúty, pretože niektoré z nich nemusia byť vhodné na rozlíšenie pečiatky od ostatných grafických prvkov, rovnako, ako farba očí človeka nie je vhodným atribútom pre klasifikáciu na mužov a ženy. Preto som pre každú hodnotu parametru k pomocou programu na vyťažovanie znalostí z dát Rapid Miner [44] hľadal podmnožinu vypočítaných atribútov, ktorá dáva pri klasifikácií čo najlepšie výsledky. Tieto podmnožiny som hľadal pomocou operátoru Optimize Selection [38] s dopredným smerom výberu atribútov. Optimálna podmnožina by sa zaručene našla, kebyže sa zmerajú výsledky každej podmnožiny atribútov a najlepšia z nej sa prehlási za optimálnu. Pri 70 atribútoch je však počet podmnožín rovný $2^{70} = 1\ 180\ 591\ 620\ 717\ 411\ 303\ 424$ a keby test jednej podmnožiny priemerne trval jednu sekundu, testovanie by trvalo 37 436 314 710 724 rokov. Preto vybratá podmnožina s veľkou pravdepodobnosťou nie je optímálna no jej výsledky sú najlepšie z pomedzi testovaných podmnožín. Konkrétne názvy členov triedy CandidateFeatures zastupujúce jednotlivé atribúty patriace jednotlivým podmnožinám uvádzam v prílohe v tabuľke E.1.

Ak atribút nebol vybratý do žiadnej z nich, v tabuľke sa nenachádza. Napriek tomu, že podmnožiny atríbútov boli vybraté pre jednotlivé k, otestoval som všetky pre k od jedna do päť. A ukázalo sa, že v niektorých prípadoch množina atribútov vybratá pre $k = i$ mala lepšie výsledky pre hodnotu j ako množina vybratá pre hodnotu j.

Úspešnosť klasifikácie hodnotím podľa niekoľkých kritérií. Pre obe skupiny hodnotím presnosť P (anglicky precision) a pomer správne klasifikovaných objektov skupiny ku všetkým objektom skupiny R (anglicky recall). Nakoniec hodnotím presnosť celkovej klasifikácie A.

$$P_p = \frac{TP}{TP + FP} \tag{5.1a}$$

$$P_o = \frac{TO}{TO + FO} \tag{5.1b}$$

$$R_p = \frac{TP}{P_P} \tag{5.1c}$$

$$R_o = \frac{TO}{P_o} \tag{5.1d}$$

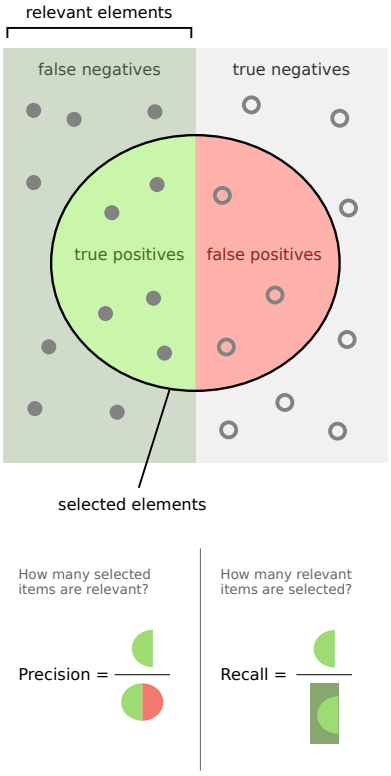
$$A = \frac{TP + TO}{P_p + P_o} \tag{5.1e}$$

$$\tag{5.1f}$$

Kde P_p je presnosť skupiny pečiatok, TP je počet správne klasifikovaných pečiatok, FP je počet nesprávne klasifikovaných pečiatok a teda počet iných grafických prvkov, ktoré boli označené ako pečiatky. Analogicky P_o je presnosť skupiny ostatných grafických prvkov, TO je počet správne klasifikovaných prvkov, ktoré nie sú pečiatky a FO je počet nesprávne klasifikovaných prvkov, ktoré sú pečiatkami, ale boli označené za iné grafické prvky. R_p je „recall“ skupiny pečiatok a R_o ostatných grafických prvkov. P_p je počet všetkých pečiatok a P_o je počet ostatných grafických prvkov. Zjednodušene povedané presnosť pečiatok P_p vyjadruje pravdepodobnosť s akou je objekt klasifikovaný ako pečiatka skutočne pečiatka a „recall“ vyjadruje koľko precent všetkých pečiatok bolo klasifikovaných ako pečiatky. Názorne sú tieto miery zobrazené na obrázku 5.2. Výsledky experimetov zachytáva obrázok 5.3 a tabuľky v prílohe F. Hodnoty osi X reprezentujú podmnožinu atribútov, ktorá bola vybratá ako najlepšia podmnožina spomedzi testovaných podmnožín pri testovaní s daným parametrom k. Na osi Y je úspešnosť klasifikácie v %. Hodnota $x = 0$ reprezentuje celú množinu atribútov.

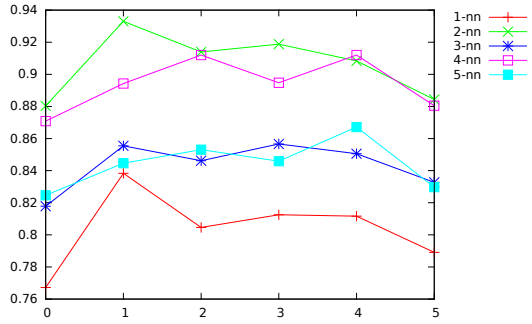
Testovanie klasifikácie prebiehalo pomocou takzvanej krížovej validácie nasledovne. Pre každý testovací dokument prebehol test tak, že sa z trénovacej množiny odstránili kandidáti z daného dokumentu a uložili sa do testovacej množiny. Prvky testovacej množiny sa postupne klasifikovali pomocou prvkov trénovacej množiny. Výsledná úspešnosť klasifikácie je priemerom úspešností testov pre jednotlivé dokumenty.

Zvolenie vhodného parametru k a podmnožiny atribútov záleží od konkrétnej úlohy, na ktorú by sa algoritmus použil. Napríklad ak by bolo cieľom zistiť či každý dokument

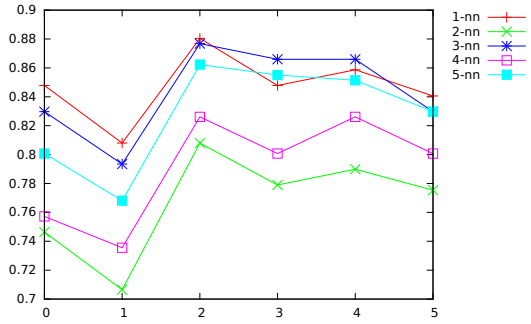


Obr. 5.2: Graficky znázornený význam mier úspešnosti klasifikácie. [zdroj:[43]]

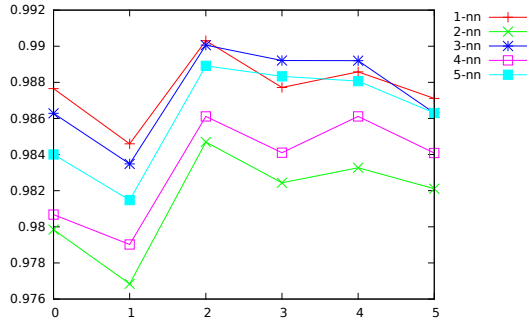
firmy obsahuje pečiatku, vybral by som podmnožinu atribútov, ktorá bola vybratá pre $k = 1$ a parameter k by som zvolil 2, pretože táto kombinácia má najväčšiu presnosť pri pečiatkach. Takže dokumenty, ktoré by boli označené že pečiatky obsahujú, by na 93 % pečiatku naozaj obsahovali. Zvyšné dokumenty by sa potom mohli skontrolovať manuálne. Naopak, keby iná inštitúcia potrebovala zistiť koľko rôznych pečiatok obsahujú jej dokumenty, zvolil by som kombináciu podmnožiny atribútov, ktorá bola vybratá pre $k = 2$ a parameter k by som zvolil 3. Táto kombinácia má takmer najvyššiu hodnotu „Recall“ pečiatok a zároveň nemá takú nízku presnosť ako keby mal parameter k hodnotu 1.



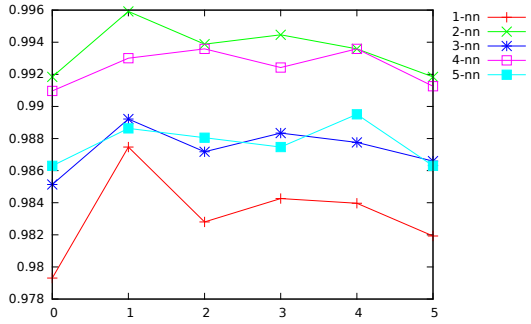
(a) Presnosť pečiatok.



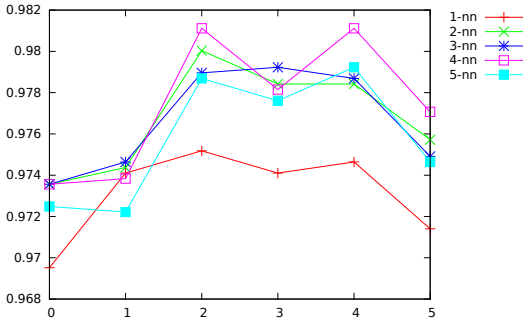
(b) „Recall“ pečiatok.



(c) Presnosť ostatných grafických prvkov.



(d) „Recall“ ostatných grafických prvkov.



(e) Celková presnosť.

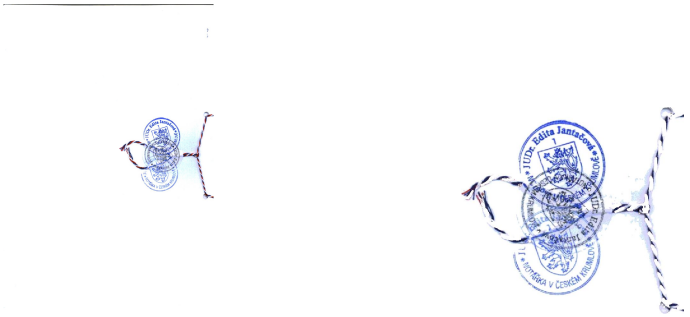
Obr. 5.3: Výsledky experimentov klasifikácie. Hodnoty osi X reprezentujú podmnožinu atribútov, ktorá bola vybratá ako najlepšia podmnožina spomedzi testovaných podmnožín pri testovaní s daným parametrom k. Na osi Y je úspešnosť klasifikácie v %. Hodnota $x = 0$ reprezentuje celú množinu atribútov.

5.4 Vyhodnotenie celého algoritmu

Vyhodnotenie celého algoritmu prebiehalo krížovou validáciou podobne ako hodnotenie klasifikácie. Z celkového množstva farebných pečiatok v dokumentoch sa extrahovalo 76,95 % pečiatok alebo ich častí. Pečiatok, ktoré niesli nejaké pre človeka zmysluplné

informácie sa extrahovalo 75,22 % a pečiatok, ktoré sa extrahovali takmer alebo úplne bez vizuálnej straty sa extrahovalo 68,89 %. Pre vyhodnotenie celého algoritmu bol parameter k algoritmu k-nn nastavený na hodnotu tri a využívala sa podmnožina aribútov číslo dva.

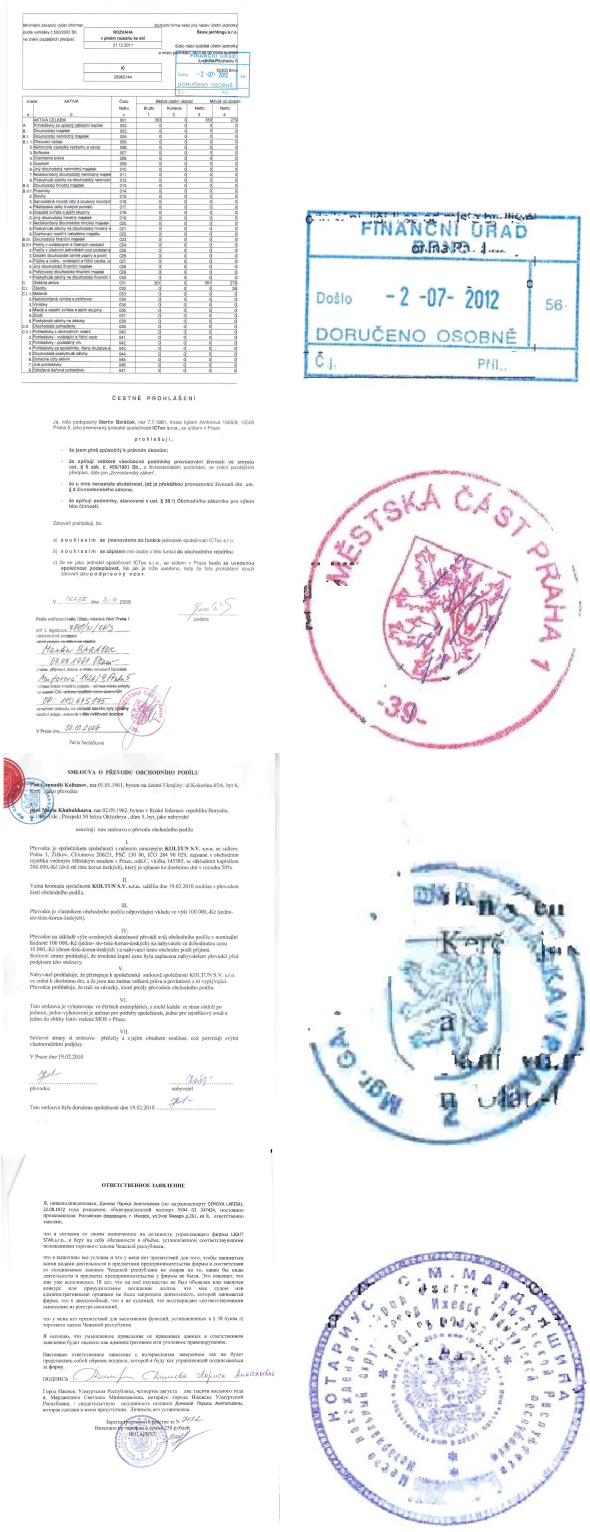
Na testovanie boli použité dokumenty nízkej, priemernej ako aj vysokej kvality, dokumenty s rôznymi typmi a farbami pečiatok, dokumenty, v ktorých boli pečiatky dobre odtlačené, dokumenty, v ktorých boli pečiatky veľmi slabo odtlačené alebo ich časti chýbali, dokumenty, v ktorých boli pečiatky prekryté iným grafickým prvkom alebo ďalšou pečiatkou ale aj dokumenty, ktorých pečiatky boli bez prekrytia. Ukážka dokumentu, v ktorom sú pečiatky, ktoré sa dajú len veľmi ťažko extrahovať pretože sa prekrývajú sú na obrázku 5.4. Ukážka stránky dokumentu s veľmi slabo viditeľnými pečiatkami je na obrázku G.1. Táto strana dokumentu obsahuje štyri pečiatky. Tri z nich zrejme nemali byť súčasťou danej stránky a pretlačili sa z inej. Strana dokumentu ich však obsahuje a preto by mali byť extrahované. Počas testov sa stalo aj to, že napriek tomu, že kadidáti pečiatky po segmentácii patrili do skupiny A, podľa rozdelenia v predošlej sekcii, boli dobre odtlačené a napriek tomu neboli klasifikované ako pečiatky. Na druhej strane na obrázku 5.5 sú ukážky pečiatok, ktoré boli extrahované veľmi dobre napriek tomu, že boli prekryté inými grafickými prvkami a patria tak medzi najväčšie úspechy implementovaného algoritmu. Vyhodnocovanie úspešnosti celého algoritmu prebiehalo manuálne.



(a) Príklad dokumentu s prekrývajúcimi sa pečiatkami.

(b) Extrahovaný kandidát, klasifikovaný ako pečiatka.

Obr. 5.4



Obr. 5.5: Ukážky dobre extrahovaných pečiatok.

5.5 Návrhy na zlepšenie

Možností na zlepšenie navrhnutého a implementovaného algoritmu je niekoľko. V prvom rade z ďalších experimentov by mohli vzísť hodnoty parametrov, ktorých úspešnosť by bola vyššia. V segmentačnej časti sú to najmä parametre, ktoré su použité na vyhladzovanie histogramu farieb dokumentu a aj parametre pri určovaní farebnej podobnosti spojených prvkov. Po prvotnej analýze stránky dokumentu by sa tieto parametre mohli nastavovať pre každý dokument zvlášť v závisloti na jeho vlastnostiach. V mojom riešení segmentácie sa využívajú najmä farebné informácie dokumentu. Za zváženie by stálo využitie geometrických vlastností na výhľadávanie kruhov, úsečiek a podobne. Za účelom zrýchlenia procesu by sa mohlo využiť viac vlákien a atribúty by sa kandidátom mohli vypočítavať súčasne. Hľadanie vhodnej množiny atribútov by mohlo zlepšiť výsledky klasifikácie rovnako, ako experimenty zaoberajúce sa vhodným výberom prvkov do trénovacej množiny. Rovnako by výsledky klasifikácie mohli zlepšiť nové atribúty. Experimenty s ďalšími klasifikačnými algoritmami, by tiež mohli viesť k zlepšeniu výsledkov.

Záver

V rámci svojej bakalárskej práce som vytvoril rešerš pozostávajúcu z prác za posledných 10 rokov. Naštudoval som potrebné algoritmi z oblasti spracovania obrazu, ktoré som neskôr v práci využil. Na základe rešerše som navrhol riešenie založené na strojovom učení, pre ktoré som použil vyše 70 atribútov a klasifikačný algoritmus k najbližších susedov. Skúmal som výber vhodnej podmnožiny atribútov za účelom zvýšenia úspešnosti klasifikácie ako aj vplyv parametru k algoritmu k najbližších susedov.

Navrhnuté riešenie som implementoval v programovacom jazyku Java, s využitím knižnice na spracovanie obrazu OpenCV ako samostatnú aplikáciu, ale aj ako modul do nástroja na spracovanie dát Surmon. Tento modul (respektíve aplikácia) umožnuje extrakciu pečiatok z naskenovaných dokumentov, ktorým následne zlepší vizuálne vlastnosti zvýšením kontrastu, odstránením šumu a otočením to základnej polohy.

Implementáciu som otestoval na dokumentoch z portálu www.justice.cz a výsledky som vyhodnotil. Nízke hodnoty úspešnosti mohli byť spôsobené tým, že testovacie dokumenty obsahovali aj dokumenty nízkej kvality, pečiatky sa v niekoľkých dokumentoch prekrývali a podobne. Sú to však reálne dáta, preto pri nasadení modulu do používania by sa výsledky nemali zhoršiť. Nakoniec som navrhol možné vylepšenia, ktoré by mohli viesť k zvýšeniu úspešnosti, na čo môže nadviazať ďalší študent alebo ja v diplomovej práci.

Literatúra

[1] AHMED, M. Discrete Cosine Transform. *Computers, IEEE Transactions on* [online]. Vol.: C-23, Issue: 1. pp. 90-93. 2006. ISSN: 0018-9340. DOI: 10.1109/T-C.1974.223784. [cit. 26.1.2015]. Dostupné z `http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1672377`.

[2] AHMED, S., SHAFAIT, F., LIWICKI, F., DENGEL, A. A Generic Method for Stamp Segmentation Using Part-Based Features. *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on* [online]. pp.708, 712, 25-28 Aug. 2013, doi: 10.1109/ICDAR.2013.145 [cit. 22.11.2014]. ISSN 1520-5363. Dostupné z: `http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6628710&isnumber=6628563`.

[3] BALLARD, D. H. and C. M. BROWN. *Computer Vision.* Prentice-Hall, INC., Englewood Cliffs, New Jersey, 1982. ISBN 978-0131653160.

[4] BUADES, A., COLL, B., MOREL, J.-M.: Non-Local Means Denoising. *Image Processing On Line*, ročník 1, 2011, doi:10.5201/ipol.2011.bcm_nlm. [cit. 6.5.2015] Dostupné z: `http://www.ipol.im/pub/art/2011/bcm_nlm/`.

[5] CAI, Liang, Li MEI. A Robust Registration and Detection Method for Color Seal Verification. *Advances in Intelligent Computing - Lecture Notes in Computer Science* [online]. Vol. 3644, 2005, pp 97-106. International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings, Part I. ISBN 978-3-540-31902-3. [cit. 26.1.2015] Dostupné z: `http://link.springer.com/chapter/10.1007%2F11538059_11`.

[6] CANNY, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Machine Intell.* Vol. 8, no. 6, pp. 679-697, 1986. ISSN 0162-8828.

[7] CALONDER, M., V. LEPETIT, M. OZUYSAL, T. TRZCINSKI, C. STRECHA and P. FUA. BRIEF: Computing a Local Binary Descriptor Very Fast. *IEEE Transac-*

tions on Pattern Analysis and Machine Intelligence. Vol. 34, no. 7, pp. 1281–1298, 2012. ISSN 0162-8828.

[8] CATTIN P. Digital Image Fundamentals. *University of Basel, Medical Image Analysis Center (MIAC)* [online]. [cit. 6.5.2015] Dostupné z: https://miac.unibas.ch/SIP/02-Fundamentals.html#%2830%29

[9] CATTIN P. Digital Image Fundamentals. *University of Basel, Medical Image Analysis Center (MIAC)* [online]. [cit. 6.5.2015] Dostupné z: https://miac.unibas.ch/SIP/02-Fundamentals.html#%2833%29

[10] DEMBELE, F.: *Object Detection using Circular Hough Transform: Introduction to the Hough Transform.* College of engineering, Michigan State University. [cit. 26.1.2015] http://www.egr.msu.edu/classes/ece480/capstone/fall10/group03/doc.html.

[11] DEY, S., MUKHERJEE, J., SURAL, S., BHOWMICK, P. Colored Rubber Stamp Removal from Document Images. *Pattern Recognition and Machine Intelligence - Lecture Notes in Computer Science*[online]. Vol. 8251, 2013, pp. 545-550. 5th International Conference, PReMI 2013, Kolkata, India, December 10-14, 2013. ISBN 978-3-642-45062-4. [cit. 26.1.2015] Dostupné z: http://link.springer.com/chapter/10.1007%2F978-3-642-45062-4_75.

[12] Discrete Fourier Transform. In: *OpenCV documentation* [online] Last updated on Feb 25, 2015. [cit. 6.5.2015] Dostupné z: http://docs.opencv.org/doc/tutorials/core/discrete_fourier_transform/discrete_fourier_transform.html

[13] FARSHAD, N., BHAVNA, F., PATI, P. B. and RAMAKRISHNAN, A. G. Automatic Seal Information Reader. *Proc. Intern. Conf. Comput. Theory Applns. (ICCTA-2007)* [online]. Kolkata, pp. 502-505. [cit. 26.1.2015]. Dostupné z: http://mile.ee.iisc.ernet.in/mile/publications/DocumentAnalysisRecognition.html.

[14] FLUSSER, J., SUK, T.: Pattern recognition by affine moment invariants. *Pattern Recognition*, ročník 26, č. 1, 1993: s. 167 – 174, ISSN 0031-3203, doi:http://dx.doi.org/10.1016/0031-3203(93)90098-H. [cit. 6.5.2015] Dostupné z: http://www.sciencedirect.com/science/article/pii/003132039390098H

[15] FORCZMAŃSKI, P., FREJLICHOWSKI, D. Efficient Stamps Classification by Means of Point Distance Histogram and Discrete Cosine Transform. *Computer Recognition Systems 4, AISC 95* [online]. Vol. 95, pp. 327-336, doi: 10.1007/978-3-642-20320-6_34 [cit. 22.11.2014]. Dostupné z: http://www.researchgate.net/publication/227270932_Efficient_Stamps_Classification_by_Means_of_Point_Distance_Histogram_and_Discrete_Cosine_Transform

[16] FORCZMAŃSKI, P., FREJLICHOWSKI, D. Principal Component Analysis of Point Distance Histogram for Recognition of Stamp Silhouettes.

Image Processing and Communications Challenges 3 Advances in Intelligent and Soft Computing, 2011 [online]. Volume 102/2011, pp. 219-226, doi: 10.1007/978-3-642-23154-4_25 [cit. 22.11.2014]. Dostupné z: http://www.researchgate.net/publication/226209557_Principal_Component_Analysis_of_Point_Distance_Histogram_for_Recognition_of_Stamp_Silhouettes.

[17] FORCZMAŃSKI, P., FREJLICHOWSKI, D. Robust Stamps Detection and Classification by Means of General Shape Analysis. *Computer Vision and Graphics — Lecture Notes in Computer Science* [online]. Vol. 6374, pp. 360–367. 2010. doi: 10.1007/978-3-642-15910-7_41 [cit. 22.11.2014]. Dostupné z: http://www.researchgate.net/publication/226318209_Robust_Stamps_Detection_and_Classification_by_Means_of_General_Shape_Analysis

[18] FORCZMAŃSKI P., MARKIEWICZ, A. Stamps Detection and Classification Using Simple Features Ensemble.*Mathematical Problems in Engineering* [online]. Article ID 367879, 2014 [cit. 22.11.2014] dostupné z https://www.google.cz/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CCYQFjAA&url=http%3A%2F%2Fdownloads.hindawi.com%2Fjournals%2Fmpe%2Faip%2F367879.pdf&ei=t61wVMGeD4SCzAOBiYHICQ&usg=AFQjCNF9sDKY0aS67sE0BBUU9ND_pe1ZRw&sig2=OZfUeFDMzgfI78EiP5n2ZA&bvm=bv.80185997,d.bGQ&cad=rja.

[19] GONZALEZ, R. C. and R. E. WOODS. D*igital Image Processing*. AddisonWesley Publishing Co., New York, 1993. ISBN:0201180758.

[20] GUANGYU, Z., JAEGER, S., DOERMANN, D. A robust stamp detection framework on degraded documents. *International Conference on Document Recognition and Retrieval XIII* [online]. Pp. 1-9, San Jose, CA, 2006 [cit. 22.11.2014]. Dostupné z: http://lampsrv02.umiacs.umd.edu/pubs/Papers/guangyuzhu-spie06/guangyuzhu-spie06.pdf.

[21] HONG, Bao, De XU, Songhe FENG. An Effective Method to Detect Seal Images from Traditional Chinese Paintings. *Wireless Communications & Signal Processing, 2009. WCSP 2009. International Conference on* [online]. pp. 1-4, 13-15 Nov. 2009. ISBN 978-1-4244-5668-0. [cit. 26.1.2015] Dostupné z: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5371700.

[22] HUANG, Z., Leng, J.: Analysis of Hu's moment invariants on image scaling and rotation. *In Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*, ročník 7, April 2010, s. V7–476–V7–480, doi:10.1109/ICCET.2010.5485542. [cit. 6.5.2015] Dostupné z:http://ro.ecu.edu.au/cgi/viewcontent.cgi?article=7351&context=ecuworks

[23] CHEN L., LIU T., CHEN J., ZHU J., DENG J., MA S. Location algorithm for seal imprints on Chinese bank-checks based on region growing. *Optoelectronics Letters*[online]. Vol. 2, issue 2, pp. 155-157. 2006, doi: 10.1007/BF03034039

[cit. 22.11.2014]. Dostupné z: http://link.springer.com/article/10.1007%2FBF03034039.

[24] CHEN H.,LOONEY C.,PARK J.: Fast Connected Component Labeling Algorithm Using A Divide and Conquer Technique [online]. [cit 6.5.2015.] Dostupné z :http://www.chasanc.com/files/fast_connected_component.pdf.

[25] CHRISTMANN A., STEINWART, I. Support Vector Machines. Information Science and Statistics, 2008 [online]. ISBN 978-0-387-77242-4. [cit. 22.11.2014]. Dostupné z: http://link.springer.com/book/10.1007%2F978-0-387-77242-4.

[26] ITSEEZ. *OpenCV* [software]. [Prístup 6.5.2015]. Dostupné z: www.opencv.org.

[27] KADLEC, T. *Webové technologie 1.* (přednáška) Praha: FIT ČVUT, 2013. [cit. 6.5.2015] Dostupné z: https://edux.fit.cvut.cz/courses/BI-WT1/lectures/architektura/architektura_mvc

[28] KOSCHAN, A.; Abidi, M. A.: *Color Spaces and Color Distances.* John Wiley & Sons, Inc., 2007, ISBN 9780470230367, 37–70 s., doi: 10.1002/9780470230367.ch3. [cit. 25.4.2015] Dostupné z: http://dx.doi.org/10.1002/9780470230367.ch3.

[29] KORDÍK, P. *Vytěžování znalostí z dat.* (přednáška) Praha: FIT ČVUT, 2015. [cit. 6.5.2015] Dostupné z: https://edux.fit.cvut.cz/courses/BI-VZD/_media/lectures/03/p3-preprocessing.pdf

[30] KRISHNAMOORTHY, M., G. NAGY, S. SETH and M. VISWANATHAN. Syntactic segmentation and labeling of digitized pages from technical journals. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* Vol. 15, pp. 737–747, 1993. ISSN 0162-8828.

[31] LEI, L., PENG, J.; YANG, B.: Image retrieval based on YCbCr color histogram. In *Cognitive Informatics Cognitive Computing (ICCI*CC), 2013 12th IEEE International Conference on*, July 2013, s. 483–488, doi: 10.1109/ICCI-CC.2013.6622287.

[32] LOWE, D.G., Distinctive Image Features from Scale Invariant Keypoints. Int'l Journal of Computer Vision (IJCV), Vol. 60, Issue 2,pp 91-110, 2004. ISSN 1573-1405.

[33] MESIAR, R., MESIAROVÁ, A., Fuzzy Integrals, Lecture Notes in Computer Science, Vol. 3131„ pp 7-14, 2004. ISBN 978-3-540-27774-3. [cit 26.1.2015] Dostupné z: http://link.springer.com/chapter/10.1007%2F978-3-540-27774-3_2.

[34] MICENKOVA, B., J. VAN BEUSEKOM. Stamp Detection in Color Document Images. *Document Analysis and Recognition (ICDAR), 2011 International Conference on* [online]. pp. 1125-1129, 18-21 Sept. 2011, doi: 10.1109/ICDAR.2011.227 [cit. 22.11.2014]. Dostupné z: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6065485&isnumber=6065247.

[35] MICENKOVÁ, B., J. VAN BEUSEKOM, SHAFAIT, F. Stamp Verification for Automated Document Authentication. *5th Int. Workshop on Computational Forensics, 2012* [online]. [cit. 22.11.2014]. Dostupné z: http://scholar.google.com/citations?view_op=view_citation&hl=en&user=6Z9AaHAAAAAJ&citation_for_view=6Z9AaHAAAAAJ:u-x6o8ySG0sC.

[36] Miscellaneous Image Transformations. In: *OpenCV documentation* [online]. Last updated on Feb 25, 2015. [cit. 6.5.2015] Dostupné z: http://docs.opencv.org/modules/imgproc/doc/miscellaneous_transformations.html#cvtcolor

[37] NEELAMEGAM, S. and DR.E.RAMARAJ, Classification algorithm in Data mining: An Overview. International Journal of P2P Network Trends and Technology (IJPTT.), Vol. 4, sep. 2013, Issue 8,pp. 369-374. ISSN: 2249-2615. [cit. 26.1.2015]. Dostupné z: http://www.ijpttjournal.org/volume-3/issue-8/IJPTT-V3I8P101.pdf.

[38] Optimize Selection. In: *RapidMiner Documentation.* [cit. 6.5.2015] Dostupné z: http://docs.rapidminer.com/studio/operators/data_transformation/attribute_space_transformation/selection/optimization/optimize_selection.html

[39] OTSU, N.: A threshold selection method from gray-level histograms. *IEEE TRANSACTIONS ON SYSTREMS, MAN, AND CYBERNETIC.* Vol. SMC-9, Issue 1, pp. 62–66, jan.1979, ISSN 0018-9472.

[40] PAN, Weiqing. Seal Imprint Segmentation Based on Color Feature Classifier. *Audio, Language and Image Processing (ICALIP), 2012 International Conference on* [online]. Pp. 837 – 840, 16-18 July 2012. ISBN 978-1-4673-0173-2. [cit. 26.1.2015] Dostupné z: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6376730.

[41] PELI, E. Contrast in Complex Images. *Journal of the Optical Society of America A 7 (10)*. Pp. 2032–2040. Oct 1990. doi:10.1364/JOSAA.7.002032.

[42] PILNÝ A. Feature Ranking/Selection, Sensitivity Analysis. (prednáška) [online]. [cit. 6.5.2015] Dostupné z: https://edux.fit.cvut.cz/courses/MI-PDD/_media/lectures/03/fs-ales.pdf

[43] Precision and recall. In: *Wikipedia: the free enclycopedia*[online]. St. Petersburg, Florida. [cit. 6.5.2015] Dostupné z: http://en.wikipedia.org/wiki/Precision_and_recall#/media/File:Precisionrecall.svg

[44] RapidMiner, Inc. *RapidMiner* [software]. [Prístup 6.5.2015]. Dostupné z: www.rapidminer.com. Požiadavky na systém: Java SE Runtime Environment.

[45] ROSTEN, E., DRUMMOND, T. Fusing points and lines for high performance tracking. *5. IEEE International Conference on Computer Vision, 2005. ICCV 2005.*, vol. 2, oct. 2005, pp. 1508 –1515 Vol. 2. ISBN 0-7695-2334-X.

[46] ROY, P. P., PAL, U., LLADÓS, J. Seal Object D1etection in Document Images using GHT of Local Component Shapes. *Proceedings of the 2010 ACM Symposium on Applied Computing* [online]. pp. 23-27. New York, NY, USA, 2010. ISBN: 978-1-60558-639-7. [cit. 26.1.2015] Dostupné z: `http://dl.acm.org/citation.cfm?doid=1774088.1774094`.

[47] RUBLEE, E., V. RABAUD, K. KONOLIGE and G. BRADSKI. Orb: an efficient alternative to sift or surf. *Computer Vision (ICCV), 2011 IEEE International Conference on.* IEEE, 2011, pp. 2564–2571. ISBN 978-1-4577-1101-5.

[48] RUWEI, D.: Chinese Character Recognition System and Integrated Approach. *Zhejiang Science and Technology Publishing House, Zhejiang, China.* 1998. Vol. 1, Issue 2, pp 126-136, may 2007, ISSN 1673-7466.

[49] Sobel Derivatives. In: *OpenCV documentation* [online]. Last updated on Feb 25, 2015. [cit. 6.5.2015] Dostupné z: `http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/sobel_derivatives/sobel_derivatives.html`

[50] SORIA-FRISCH, A. Color Seal Extraction from Documents: Robustness through Soft Data Fusion. *EURASIP Journal on Advances in Signal Processing 2005* [online]. Vol. 2005:812769, doi: 10.1155/ASP.2005.2146. [cit. 26.1.2015] Dostupné z: `http://asp.eurasipjournals.com/content/2005/13/812769`.

[51] *SURMON* [software]. [Prístup 6.5.2015]. `www.surmon.org`.

[52] THE APACHE SOFTWARE FOUNDATION. *Apache Derby 10.10* [software]. [cit. 6.5.2015] Dostupné z: `https://db.apache.org/derby/derby_downloads.html`

[53] UEDA, K. Automatic Seal Imprint Verification System for Bankcheck Processing. *Information Technology and Applications, 2005. ICITA 2005. Third International Conference on* [online]. Vol. 1, pp. 768 – 771, 4-7 July 2005. ISBN 0-7695-2316-1. Dostupné z: `http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=1488904`.

[54] UEDA, K. Extraction of Signature and Seal Imprint from Bankchecks by Using Color Information. *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on* [online]. Vol. 2, pp. 665 – 668. Aug. 1995. ISBN 0-8186-7128-9. Dostupné z: `http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=601983`.

[55] YCbCrColorSpace Perspective. In: *Wikipedia: the free enclycopedia*[online]. St. Petersburg, Florida. [cit. 6.5.2015] Dostupné z: `http://en.wikipedia.org/wiki/Talk%3AYCbCr#/media/File:YCbCrColorSpace_Perspective.png`

[56] WANG, Liqiang, Xuxiang NI and Zukang LU. Robust protein microarray image segmentation using improved seeded region growing algorithm. *Chinese Optics Letters.* Vol. 1, 2003, Issue 9, p. 520.

[57] Weight by Correlation. In: *RapidMiner Documentation*. [cit. 6.5.2015] Dostupné z: http://docs.rapidminer.com/studio/operators/modeling/weighting/weight_by_correlation.html

[58] Weight by Relief. In: *RapidMiner Documentation*. [cit. 6.5.2015] Dostupné z: http://docs.rapidminer.com/studio/operators/modeling/weighting/weight_by_relief.html

[59] *Wikipedia: the free enclycopedia*[online]. St. Petersburg, Florida. [cit. 6.5.2015] Dostupné z: http://upload.wikimedia.org/wikipedia/commons/3/34/YCbCr-CbCr_Scaled_Y50.png

[60] [cit. 6.5.2015] www.oracle.com/java/

Dodatok **A**

Zoznam použitých skratiek

C_bC_r Rovina v priestore YC_bC_r, ktorý popisujem v sekcii 3.3.3

GLCM Matica výskytu hodnoty odtieňa šedej (anglicky Gray-Level Coocurrence Matrix), ktorú popisujem v sekcii 3.5 a vznikne podľa vzťahu 3.9

HSV Farebný priestor, ktorý popisujem v sekcii 3.3.2

k-nn Algoritmus k najbližších susedov (anglicky k nearest neighbours), ktorý popisujem v sekcii 3.6

MVC Architektúra Model-View-Controller, ktorú popisujem v 4.3

RGB Farebný priestor, ktorý popisujem v sekcii 3.3.1

YC_bC_r Farebný priestor, ktorý popisujem v sekcii 3.3.3

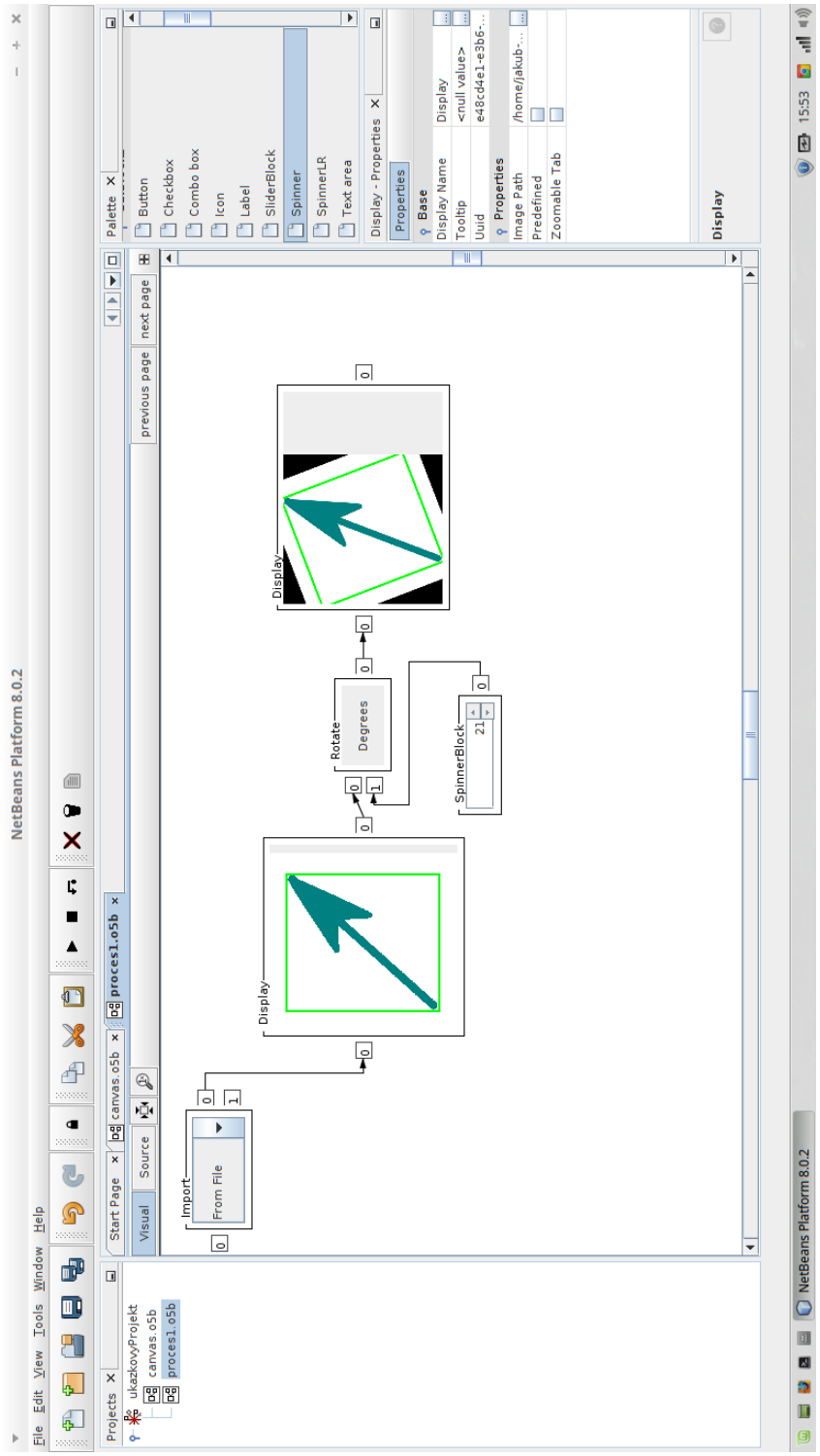
Dodatok **B**

Obsah priloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `impl` ..... implementácia
  - `app` ..... súbory týkajúce sa samostatnej aplikácie
    - `doc` ..... dokumentácia
    - `src` ..... zdrojové kódy
    - `StampExtraction.jar` ..... spustitelná verzia aplikácie
    - `readme.txt` ..... návod na spustenie aplikácie
    - `data` ..... databáza a súbor s trénovacou množinou
  - `modul` ..... súbory týkajúce sa modulu do programu Surmon
    - `readme.txt inštrukcie k používaniu`
    - `src` ..... zdrojové súbory
    - `org-obbb-stampextractionmodule.nbm` ..... plugin do programu Surmon
- `thesis` ..... text práce
  - `thesis.pdf` ..... text práce vo formáte PDF
  - `thesis-tex` ..... zdrojová forma práce vo formáte LaTeX

Dodatok C

Príklad procesu v programe Surmon



Obr. C.1: Príklad procesu v programe Surmon, ktorý načíta obrázok, zobrazí otočí o zadaný uhol a znovu zobrazí.

Dodatok **D**

Tabuľky významnosti atribútov

Atribút	**Významnosť**
nestedConnectedComponentsCountToComponentCountRatio	1.0
numberOfGoodFeaturesToTrackFromGrayScaleWithHarris	0.913039206410343
numberOfGoodFeaturesToTrackFromGrayScale	0.8711258405597944
rectangleS	0.7878295658809078
numberOfGoodFeaturesToTrackFromBinaryMask	0.6479942212560624
triangleS	0.6279855318297483
numberOfGoodFeaturesToTrackFromBinaryMaskWithHarris	0.6033115308162029
minAreaRectShortSideToPageWidth	0.5724184288737695
sobelMatrixAverage	0.5376276757564298
ellipseS	0.5127160440853749
candidateHeightToPageHeight	0.5110667894791638
candidateWidthToPageWidth	0.5054364038541737
squareness	0.49992928513846013
pageAreaToCandidateAreaRatio	0.430268758299508
standardDeviationInConnectedComponentMinimalRectangleAngle	0.40361625651903377
componentsPerimeter	0.40027826423205587
correlationOfGLCM	0.38836174815590413
nestedConnectedComponentsCount	0.3562255886285377
density	0.2971444870758472
fastFourierMagnitudeMatrixAverage	0.2967182539698414
roundness	0.28321186846795215
circularVariance	0.2733534869633867
fastFourierMagnitudeMatrixVariance	0.26239443161957654
standardDeviationOfPixelIntensitiesFromGrayScale	0.2619765609083875
fastFourierMagnitudeMatrixStandardDeviation	0.2519459633795267
averagePixelIntensityFromGrayScale	0.2432943114729675
averageConnectedComponentMinimalRectangleAngle	0.23630616395805082
standardDeviationOfPixelIntensitiesFromMyIntensitieMatrix	0.21485869521317685
averagePixelIntensityFromMyIntensitieMatrix	0.214476070315913
rhombusS	0.19498454880241264
modusOfConnectedComponentMinimalRectangleAngle	0.19152432898543933
fiAverage	0.1870754815293433
numberOfColorClassesInBoundingBox	0.184346739023054
minAreaRectLongSideToPageWidth	0.16902909851093642
contrastOfGLCM	0.16719941002342162
pageWidthToCandidateComponentsPerimeter	0.16327956769397484
averagePixelIntensityFromGrayScaleToMichelsonContrasFromMyIntensityMatrix	0.16319622445502263
averagePixelIntensityFromMyIntensitiesMatrixToMichelsonContrasFromMyIntensityMatrix	0.162876505997783
averagePixelIntensityFromGrayScaleToMichelsonContrasGrayScaleMatrix	0.15117842343533403
averagePixelIntensityFromMyIntensitiesMatrixToMichelsonContrasGrayScaleMatrix	0.1509914607589674
varianceInConnectedComponentMinimalRectangleAngle	0.14876501546440973
varianceInPixelIntensitiesFromGrayScale	0.1471703831966982
varianceInH	0.13001638435616744
relativeXPosition	0.12769286703792954
candidateAreaToComponentsPerimeter	0.12329513973356798
averagePixelIntensityFromMyIntensitiesMatrixToRMSContrastFromMyIntensityMatrix	0.11853610578142365
averagePixelIntensityFromGrayScaleToRMSContrastFromMyIntensityMatrix	0.11726682076453401
minAreaRectangleAspectRatio	0.1169837145317398
varianceInFi	0.1159985857169781
minAreaRectangleAreaToPageArea	0.10268742682329324
varianceInPixelIntensitiesFromMyIntensitieMatrix	0.09674922213766483
huMoment0	0.09563039074253311
hAverage	0.08779290912471684
averagePixelIntensityFromMyIntensitiesMatrixToRMSContrastFromGrayScaleMatrix	0.08593846642842952
averagePixelIntensityFromGrayScaleToRMSContrastFromGrayScaleMatrix	0.08488876039843089
circularS	0.07956722084788345
RMSContrastFromGrayScaleMatrix	0.0724545195690348
RMSContrastFromMyIntensityMatrix	0.0721878111714331
widthToHeightRatio	0.06288074586920012
huMoment1	0.053078519911973834
angleOfMinimalRectangleArea	0.04823054680329567
componentsAreaToComponentsPerimeter	0.04418287823290984
energyOfGLCM	0.03928196150351749
honogenityOfGLCM	0.039194975882833986
triangularity	0.031163375884762177
ellipticity	0.03116337588379224
michelsonContrasGrayScaleMatrix	0.02594644777894006
relativeYPosition	0.016489681931382238
huMoment2	0.010654458183624248
huMoment3	0.010641148719033353
huMoment5	0.009561882244176555
squareS	7.897320217166446E-4
michelsonContrasFromMyIntensityMatrix	0.0

Tabuľka D.1: Tabuľka významnosti počítaná pomocou korelácie.

Atribút	**Významnosť**
nestedConnectedComponentsCountToComponentCountRatio	1.0
candidateHeightToPageHeight	0.6088924832319682
fiAverage	0.49083582957324623
rhombusS	0.45693391972865705
relativeXPosition	0.45215633471327094
circularS	0.4486295091041782
honogenityOfGLCM	0.43928006642705963
energyOfGLCM	0.43560821666286964
ellipseS	0.42970519862535983
sobelMatrixAverage	0.42651563902962436
minAreaRectLongSideToPageWidth	0.4236285829410906
hAverage	0.41151664581317665
angleOfMinimalRectangleArea	0.36738333542470486
modusOfConnectedComponentMinimalRectangleAngle	0.27060196240024215
fastFourierMagnitudeMatrixStandardDeviation	0.26233757274990405
triangleS	0.2599452138186974
candidateWidthToPageWidth	0.24546701320882702
numberOfColorClassesInBoundingBox	0.2029800031259197
fastFourierMagnitudeMatrixVariance	0.20133293314786402
numberOfGoodFeaturesToTrackFromGrayScaleWithHarris	0.1916301602282397
averagePixelIntensityFromGrayScale	0.18620724841428252
numberOfGoodFeaturesToTrackFromGrayScale	0.1858926736579562 3
relativeYPosition	0.1749356030309962
huMoment0	0.16868622299294805
averagePixelIntensityFromMyIntensitieMatrix	0.1481283356987159
averageConnectedComponentMinimalRectangleAngle	0.14086712537393709
squareness	0.1354326337545367
michelsonContrasFromMyIntensityMatrix	0.12125439389534676
standardDeviationOfPixelIntensitiesFromGrayScale	0.11938387423037156
standardDeviationInConnectedComponentMinimalRectangleAngle	0.1074858161843357
contrastOfGLCM	0.10123058356399701
standardDeviationOfPixelIntensitiesFromMyIntensitieMatrix	0.07418434209185006
michelsonContrasGrayScaleMatrix	0.07393618651204285
correlationOfGLCM	0.07364430938465744
pageAreaToCandidateAreaRatio	0.07065575357167785
varianceInPixelIntensitiesFromGrayScale	0.059477839127705566
roundness	0.04779784520528921
nestedConnectedComponentsCount	0.04644644610431127
squareS	0.04575350020376875
widthToHeightRatio	0.04228013597796803
varianceInConnectedComponentMinimalRectangleAngle	0.0374533051510516
varianceInPixelIntensitiesFromMyIntensitieMatrix	0.03317841331681991
rectangleS	0.030892749864342862
huMoment1	0.028355789425509093
RMSContrastFromMyIntensityMatrix	0.026473336187808207
RMSContrastFromGrayScaleMatrix	0.02625696312815998
density	0.02602101603402817
circularVariance	0.025530800940713208
minAreaRectangleAreaToPageArea	0.01952901153959007
ellipticity	0.018622492113042583
triangularity	0.018622492108617678
numberOfGoodFeaturesToTrackFromBinaryMask	0.011088455194860132
minAreaRectShortSideToPageWidth	0.006585084065385199
averagePixelIntensityFromMyIntensitiesMatrixToRMSContrastFromMyIntensityMatrix	0.006156454165471674
averagePixelIntensityFromGrayScaleToRMSContrastFromMyIntensityMatrix	0.006143732774141256
candidateAreaToComponentsPerimeter	0.005598368528629776
minAreaRectangleAspectRatio	0.005430784349483658
averagePixelIntensityFromMyIntensitiesMatrixToRMSContrastFromGrayScaleMatrix	0.004996317208771304
averagePixelIntensityFromGrayScaleToRMSContrastFromGrayScaleMatrix	0.004984052786567599
averagePixelIntensityFromMyIntensitiesMatrixToMichelsonContrasFromMyIntensityMatrix	0.003379783611665663
averagePixelIntensityFromGrayScaleToMichelsonContrasFromMyIntensityMatrix	0.0032375532435235626
componentsAreaToComponentsPerimeter	0.002784677316976962 6
numberOfGoodFeaturesToTrackFromBinaryMaskWithHarris	0.002366388156924655
componentsPerimeter	0.0019013258595602727
varianceInH	0.0010926973727415678
averagePixelIntensityFromMyIntensitiesMatrixToMichelsonContrasGrayScaleMatrix	9.9643512503252E-4
averagePixelIntensityFromGrayScaleToMichelsonContrasGrayScaleMatrix	9.405488086704389E-4
varianceInFi	2.747683353256253E-4
pageWidthToCandidateComponentsPerimeter	1.783496288769899E-4
fastFourierMagnitudeMatrixAverage	7.86682536826951E-5
huMoment3	3.282876889650116E-6
huMoment2	2.5109541277723046E-6
huMoment5	1.3837728241954954E-7

Tabuľka D.2: Tabuľka významnosti počítaná pomocou algoritmu relief.

Dodatok E

Výsledné podmnožiny atribútov

Atribút	**Podmnožina (resp. parameter k)**
averageConnectedComponentMinimalRectangleAngle	2
averagePixelIntensityFromGrayScale	1, 3, 4
averagePixelIntensityFromGrayScaleToMichelsonContrasFromMyIntensityMatrix	2, 3, 4, 5
averagePixelIntensityFromGrayScaleToMichelsonContrasGrayScaleMatrix	3, 4, 5
averagePixelIntensityFromGrayScaleToRMSContrastFromGrayScaleMatrix	2, 3, 4, 5
averagePixelIntensityFromGrayScaleToRMSContrastFromMyIntensityMatrix	1, 2, 3, 4, 5
averagePixelIntensityFromMyIntensitieMatrix	1, 3
averagePixelIntensityFromMyIntensitiesMatrixToMichelsonContrasFromMyIntensityMatrix	2, 3, 4, 5
averagePixelIntensityFromMyIntensitiesMatrixToMichelsonContrasGrayScaleMatrix	1, 3, 4, 5
averagePixelIntensityFromMyIntensitiesMatrixToRMSContrastFromGrayScaleMatrix	1, 3, 4, 5
averagePixelIntensityFromMyIntensitiesMatrixToRMSContrastFromMyIntensityMatrix	2, 3, 4
candidateAreaToComponentsPerimeter	2, 3, 4, 5
candidateWidthToPageWidth	2, 3, 4
circularS	3, 4
circularVariance	3, 4, 5
componentsAreaToComponentsPerimeter	1, 3, 4, 5
componentsPerimeter	1, 2, 3, 4, 5
connectedComponentsCount	1, 2, 3, 4, 5
contrastOfGLCM	3
correlationOfGLCM	3, 4
density	2, 4, 5
ellipseS	2, 3, 4
ellipticity	1, 3, 4
energyOfGLCM	1
fastFourierMagnitudeMatrixStandardDeviation	2, 3
fastFourierMagnitudeMatrixVariance	3
fiAverage	3
hAverage	1, 2, 3, 4, 5
honogenityOfGLCM	1, 3
huMoment0	2, 3, 4, 5
huMoment1	3, 4, 5
huMoment2	3, 4, 5
huMoment3	2, 3, 4, 5
huMoment4	2, 3, 4, 5
huMoment5	4, 5
michelsonContrasFromMyIntensityMatrix	3
michelsonContrasGrayScaleMatrix	1
minAreaRectangleAreaToPageArea	2, 3, 4, 5
minAreaRectangleAspectRatio	1, 3, 4
minAreaRectLongSideToPageWidth	1, 3
minAreaRectShortSideToPageWidth	3, 4, 5
modusOfConnectedComponentMinimalRectangleAngle	3
nestedConnectedComponentsCount	1, 2, 3, 4, 5
nestedConnectedComponentsCountToComponentCountRatio	3
numberOfColorClassesInBoundingBox	3, 4
numberOfGoodFeaturesToTrackFromBinaryMask	1, 5
numberOfGoodFeaturesToTrackFromGrayScaleWithHarris	1, 3
pageAreaToCandidateAreaRatio	2, 3, 4
pageWidthToCandidateComponentsPerimeter	2, 3, 4, 5
rectangleS	4, 5
relativeXPosition	2, 4, 5
relativeYPosition	2, 3, 4
RMSContrastFromGrayScaleMatrix	3, 4, 5
RMSContrastFromMyIntensityMatrix	4, 5
roundness	1, 2, 3, 4, 5
sobelMatrixAverage	2, 3, 4
squareness	3, 4
squareS	4, 5
standardDeviationInConnectedComponentMinimalRectangleAngle	3, 4
standardDeviationOfPixelIntensitiesFromGrayScale	3
standardDeviationOfPixelIntensitiesFromMyIntensitieMatrix	1, 2, 3, 4
triangleS	1, 2, 3
triangularity	2, 3, 4, 5
varianceInConnectedComponentMinimalRectangleAngle	3, 4, 5
varianceInFi	3,4
varianceInH	2, 3, 4, 5
varianceInPixelIntensitiesFromGrayScale	1, 3
varianceInPixelIntensitiesFromMyIntensitieMatrix	3
widthToHeightRatio	3, 4, 5

Tabuľka E.1: Výsledné podmnožiny atribútov.

Dodatok F

Výsledky klasifikácie v tabuľkách

Podmnožina atribútov	1-nn	2-nn	3-nn	4-nn	5-nn
0	0.7672131147540984	0.8803418803418803	0.8178571428571428	0.8708333333333333	0.8246268656716418
1	0.8383458646616542	0.9330143540669856	0.85546875	0.8942731277533039	0.8446215139442231
2	0.804635761589404	0.9139344262295082	0.8461538461538461	0.912	0.8530465949820788
3	0.8125	0.9188034188034188	0.8566308243727598	0.8947368421052632	0.8458781362007168
4	0.8116438356164384	0.9083333333333333	0.8505338078291815	0.912	0.8671586715867159
5	0.7891156462585034	0.8842975206611571	0.8327272727272728	0.8804780876494024	0.8297101449275363

Tabuľka F.1: Presnosť triedy pečiatka.

Podmnožina atribútov	1-nn	2-nn	3-nn	4-nn	5-nn
0	0.8478260869565217	0.7463768115942029	0.8297101449275363	0.7572463768115942	0.8007246376811594
1	0.8079710144927537	0.7065217391304348	0.7934782608695652	0.7355072463768116	0.7681159420289855
2	0.8804347826086957	0.8079710144927537	0.8768115942028986	0.8260869565217391	0.8623188405797102
3	0.8478260869565217	0.7789855072463768	0.8659420289855072	0.8007246376811594	0.855072463768116
4	0.8586956521739131	0.7898550724637681	0.8659420289855072	0.8260869565217391	0.8514492753623188
5	0.8405797101449275	0.7753623188405797	0.8297101449275363	0.8007246376811594	0.8297101449275363

Tabuľka F.2: „Recall“ triedy pečiatka.

Podmnožina atribútov	1-nn	2-nn	3-nn	4-nn	5-nn
0	0.9876543209876543	0.9798445148286784	0.9862853807995331	0.980674935102394	0.9840069787728991
1	0.984597500726533	0.9768439108061749	0.9834830483917705	0.9790229885057471	0.9814814814814815
2	0.9903083700440528	0.9846953508518625	0.9900613855597779	0.9861151287243275	0.9889148191365228
3	0.9877157063468851	0.9824359343507054	0.9892065344224037	0.9841040462427746	0.9883313885647608
4	0.9885797950219619	0.9832708393423709	0.9892002335084646	0.9861151287243275	0.9880675203725262
5	0.9871081160269558	0.9821067821067822	0.9863053613053613	0.9840856481481481	0.9863013698630136

Tabuľka F.3: Presnosť triedy iný grafický prvok.

Podmnožina atribútov	1-nn	2-nn	3-nn	4-nn	5-nn
0	0.9793063246866802	0.9918391139609444	0.9851355290002914	0.9909647333139027	0.9863013698630136
1	0.987467210725736	0.9959195569804722	0.9892159720198193	0.9930049548236666	0.9886330515884582
2	0.9828038472748469	0.9938793354707083	0.9871757505100553	0.9935878752550277	0.9880501311570971
3	0.9842611483532497	0.9944622559020694	0.9883415913727777	0.9924220343923055	0.987467210725736
4	0.9839696881375692	0.9935878752550277	0.9877586709414165	0.9935878752550277	0.9895074322354999
5	0.9819294666278053	0.9918391139609444	0.9865928300786942	0.9912561935295832	0.9863013698630136

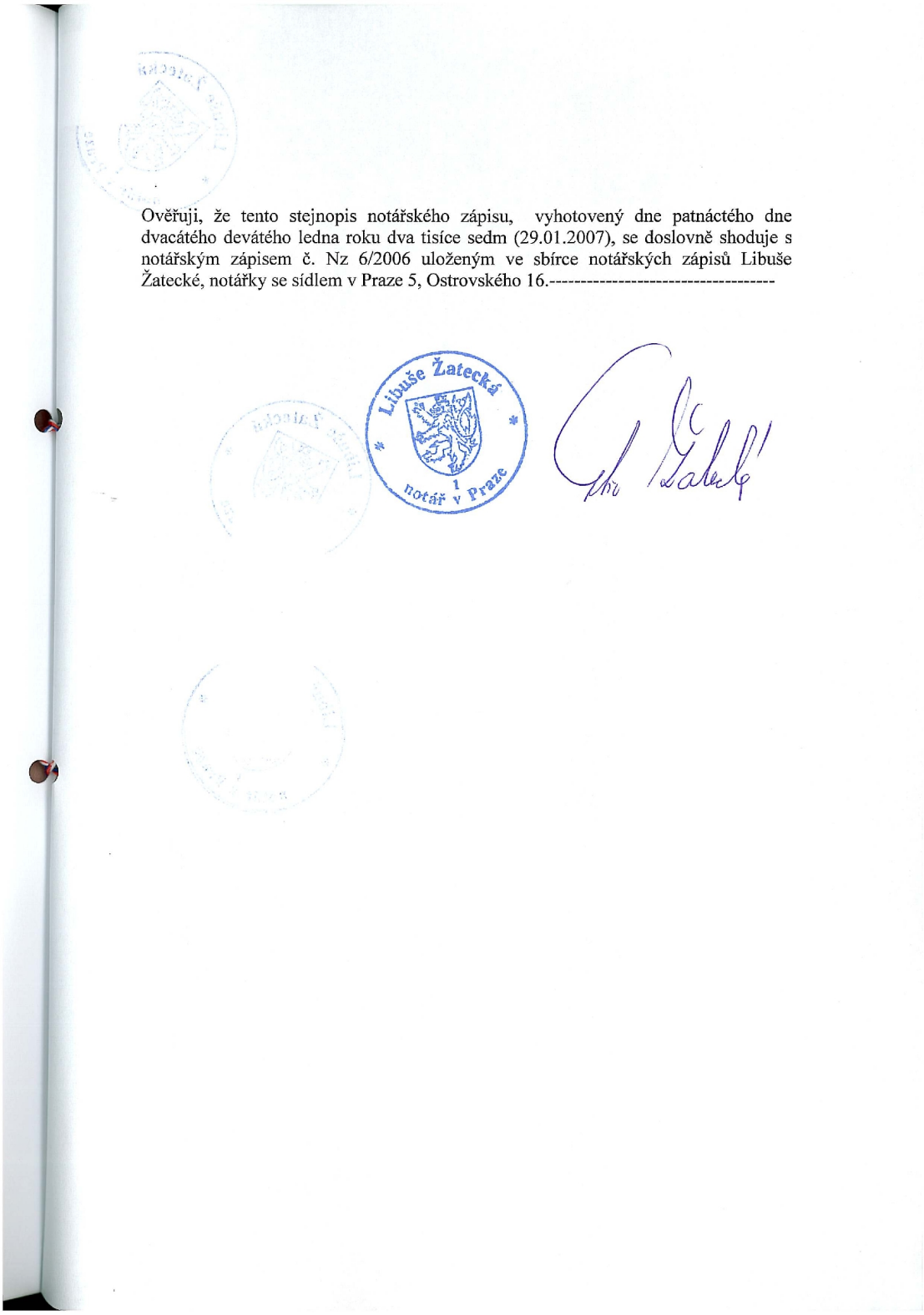
Tabuľka F.4: „Recall“ triedy iný grafický prvok.

Podmnožina atribútov	1-nn	2-nn	3-nn	4-nn	5-nn
0	0.9695171298	0.9735635285	0.9735635285	0.9735635285	0.9724844888
1	0.9741030483	0.9743728082	0.9746425681	0.9738332884	0.9722147289
2	0.9751820879	0.9800377664	0.9789587267	0.981116806	0.9786889668
3	0.9741030483	0.9784192069	0.9792284866	0.978149447	0.9776099272
4	0.9746425681	0.9784192069	0.9786889668	0.981116806	0.9792284866
5	0.9714054492	0.9757216078	0.974912328	0.9770704073	0.9746425681

Tabuľka F.5: Celková presnosť klasifikácie.

Dodatok G

Ukážka dokumentu s veľmi slabo viditeľnými pečiatkami


Ověřuji, že tento stejnopis notářského zápisu, vyhotovený dne patnáctého dne dvacátého devátého ledna roku dva tisíce sedm (29.01.2007), se doslovně shoduje s notářským zápisem č. Nz 6/2006 uloženým ve sbírce notářských zápisů Libuše Žatecké, notářky se sídlem v Praze 5, Ostrovského 16.----------
Libuše Žatecká
1
notář v Praze

Obr. G.1