



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

**POČÍTAČOVÁ PODPORA ROZPOZNÁVÁNÍ
A KLASIFIKACE RODOVÝCH ERBŮ**

COMPUTER AIDED RECOGNIZATION AND CLASSIFICATION OF COAT OF ARMS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. FRANTIŠEK VÍDEŇSKÝ

VEDOUCÍ PRÁCE

SUPERVISOR

Doc. Ing. FRANTIŠEK ZBOŘIL, Ph.D.

BRNO 2017

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav inteligentních systémů

Akademický rok 2016/2017

Zadání diplomové práce

Řešitel: **Vídeňský František, Bc.**

Obor: Inteligentní systémy

Téma: **Počítačová podpora rozpoznávání a klasifikace rodových erbů**
Computer Aided Recognition and Classification of Coat of Arms

Kategorie: Umělá inteligence

Pokyny:

1. Seznamte se jednak se základy heraldiky, strukturou rodových erbů a dále se současnými metodami softwarové dekompozice a klasifikace obrazů.
2. Zvolte vhodné metody pro počítačové zpracování obrazů obsahujících rodové erby. Tyto metody mají umožnit provádět počítačem rozpoznání erbu jako celku, jeho transformaci a normalizaci a dále dekompozici objektu erbu na jeho základní části - štít, prilbu, točenici, přikrývadla a klenot.
3. Realizujte metody, které po provedení dekompozice provedou klasifikaci jednotlivých částí podle daných heraldických pravidel a zvyklostí.
4. Implementujte tyto metody a ověřte jejich funkčnost pro zvolenou množinu příkladů, a to i fotografií erbů realizovaných jako znaky na budovách či náhrobcích.
5. Vytvořte plakát formátu A1, na kterém budou demonstrovány podstatné výsledky tohoto projektu.

Literatura:

- Opora předmětu Základy umělé inteligence
- Horák, K., Kalová, I., Petyovský, P., Richter, M.: Počítačové vidění, FIT VUT, 2008, Brno

Při obhajobě semestrální části projektu je požadováno:

- První dva body zadání

Podrobné závazné pokyny pro vypracování diplomové práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva diplomové práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap, které byly vyřešeny v rámci dřívějších projektů (30 až 40% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Zbořil František, doc. Ing., Ph.D.,** UITS FIT VUT

Datum zadání: 1. listopadu 2016

Datum odevzdání: 24. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav inteligentních systémů
612 66 Brno, Božetěchova 2

doc. Dr. Ing. Petr Hanáček
vedoucí ústavu

Abstrakt

Tato diplomová práce se zabývá návrhem systému pro rozpoznávání a klasifikaci rodových erbů jako celku i jeho jednotlivých heraldických součástí. V práci jsou představeny metody počítačového vidění pro segmentaci a detekci objektu a vybrány nejvhodnější z nich. Převážná část součástí erbu je segmentovaná pomocí konvolučních neuronových sítí a zbylé pomocí aktivních kontur. Pro detekci erbů v obraze byla vybrána metoda Histogramu orientovaných gradientů. Pro trénování i ověření funkčnosti je využita vlastní datová sada. Výsledný systém je možné použít jako pomocný nástroj v pomocných vědách historických.

Abstract

This master thesis describes the design and development of the system for detection and recognition of whole coat of arms as well as each heraldic parts. In the thesis are presented methods of computer vision for segmentation and detection of an object and selected methods that are the most suitable. Most of the heraldic parts are segmented using a convolution neural networks and the rest using active contours. The Histogram of the gradient method was selected for coats of arms detection in an image. For training and functionality verification is used my own data set. The resulting system can serve as an auxiliary tool used in auxiliary sciences of history.

Klíčová slova

heraldika, počítačové vidění, Histogram orientovaných gradientů, Support vector machine, konvoluční neuronové sítě, sémantická segmentace, aktivní kontury

Keywords

heraldry, computer vision, Histograms of oriented gradients, Support vector machine, convolutional neural networks, semantic segmentation, active contours

Citace

VÍDEŇSKÝ, František. *Počítačová podpora rozpoznávání a klasifikace rodových erbů*. Brno, 2017. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Zbořil František.

Počítačová podpora rozpoznávání a klasifikace rodových erbů

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana doc. Ing. Františka Zbořila, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

František Vídeňský

15. května 2017

Poděkování

Rád bych poděkoval panu doc. Ing. Františkovi Zbořilovi, Ph.D. za odborné vedení, trpělivost a ochotu při vypracování této diplomové práce. Také bych chtěl poděkovat mé rodině za poskytnutou podporu v době studia a při psaní této práce.

Obsah

1	Úvod	3
2	Heraldika	4
2.1	Definice heraldiky	4
2.2	Historie heraldiky	4
2.3	Znak	6
2.3.1	Kovy, barvy a kožešiny	6
2.4	Štít	7
2.4.1	Heroldské figury	7
2.4.2	Obecné figury	8
2.5	Přilba a její výstroj	9
2.5.1	Přikryvadla	10
2.5.2	Helmovní koruna a točenice	10
2.5.3	Klenot	10
2.6	Honosné kusy	11
3	Vybrané oblasti počítačového vidění	13
3.1	Snímání a předzpracování obrazu	13
3.2	Segmentace obrazu	14
3.2.1	Prahování	15
3.2.2	Detekce hran	15
3.2.3	Metoda rozvodí	15
3.2.4	Aktivní kontury	15
3.2.5	Shluková analýza	16
3.2.6	Statistické modely	16
3.3	Rozpoznávání syntakticky popsaných objektů	17
3.4	Detekce objektu v obraze	18
3.4.1	Houghova transformace	19
3.4.2	Kaskádový klasifikátor	20
3.4.3	Histogram orientovaných gradientů	23
3.5	Matematická morfologie	24
4	Konvoluční neuronové sítě	26
4.1	Konvoluce	26
4.2	Architektura	27
4.2.1	Konvoluční vrstva	28
4.2.2	Pooling vrstva	29
4.2.3	Plně propojená vrstva	29

4.3	Učení konvolučních neuronových sítí	30
4.4	Sémantická segmentace	31
5	Návrh a implementace systému	33
5.1	Rozdělení systému	33
5.2	Výběr metod	34
5.2.1	Metoda pro detekci štítu	34
5.2.2	Metody pro segmentaci a klasifikaci jednotlivých součástí erbu	35
5.3	Detektor štítu	35
5.3.1	Optimalizace nastavení	35
5.4	Segmentace a klasifikace štítu, přilby, helmovní koruny, točenice a honosných kusů	36
5.4.1	Sjednocení štítu a přilby	37
5.5	Segmentace klenotu a přikryvadel	38
5.6	Propojení částí systému	38
6	Ověření funkčnosti a experimenty	40
6.1	Výběr metrik	40
6.2	Výsledky testů pro ověření funkčnosti	41
6.2.1	HOG detektor štítu	41
6.2.2	Segmentace součástí se známým tvarem pomocí FCN-8s sítě	42
6.2.3	Segmentace klenotů a přikryvadel pomocí aktivních kontur	44
6.3	Experimenty	46
7	Závěr	49
	Literatura	50
	Přílohy	53
A	Výsledky testů pro detektory štítu	54
B	Obsah přiloženého DVD	55

Kapitola 1

Úvod

Naše republika se skládá ze tří historických zemí. Království českého, Markrabství moravského a části Slezska (takzvané České Slezsko). Všechny tři historické země mají dlouhou a bohatou historii, ale nejslavnější období prožívaly ve středověku, ve kterém mocenský základ státu tvořila skupina vysoce společensky postavených jedinců, středověká šlechta.

Každý šlechtický rod, od dob první křížové výpravy [30], si dědičně držel vlastní znak (erb), pomocí kterého se šlechta rozeznávala a prezentovala. I když doba šlechty v našich zemích pominula, stále můžeme nalézt její stopy. Nejpatrnější jsou na stovkách hradů a zámků, ve kterých šlechtici žili a většina rodů na nich svůj erb zanechala. Můžeme nalézt erby vyryté nad bránou, na stěnách, nakreslené v erbovních sálech nebo pomocí erbů ozdobené věci ke každodennímu užívání, od talířů přes židle až po dýmky. Šlechta však nespravovala jen svá vlastní sídla, ale vlastnila města nebo obce, ve kterých měla svou rezidenci, na kterých také můžeme nalézt erb. Některé obce dokonce erb bývalého vlastníka převzali jako svůj vlastní znak. Dále šlechta vlastnila nebo podporovala kostely, kláštery, na kterých také můžeme nalézt erby, nechávala stavět různé sochy, památníky nebo rodinné hroby či náhrobní kameny. Pokud se projdeme po historických částí měst, určitě v každém nějaký rodový erb nalezneme. Erby lze však stejně často nalézt v literatuře. V soudobé literatuře několik autorů vydalo kresby domácích nebo zahraničních erbů. Častěji je však nalezneme ve starých spisech, listinách nebo na pečetích.

Poslední desetiletí se velmi vyvíjí a stále více používá *počítačové vidění* „Disciplína, která se snaží technickými prostředky alespoň částečně napodobit lidské vidění.“ [24]), které můžeme nalézt v průmyslu pro kontrolu výrobků, bezpečnostních systémech, dnešní automobily umí detekovat chodce či jiné překážky nebo nám dokáží přečíst a zobrazit dopravní značky. Policie využívá počítačové vidění pro automatickou detekci a zaznamenání dopravních přestupků, nebo pro přečtení státní poznávací značky, pro hlídání, zda auto nestojí na parkovišti déle, než je stanovená doba nebo hledání ukradených vozidel.

Dosud však nevznikla aplikace pro rozpoznávání erbů i přes to, že existuje věda, která se erby přímo zabývá a nazývá se heraldika.

Tato práce se zabývá rozpoznáváním a klasifikací rodových erbů jako celku, jeho dekompozicí a následnou klasifikací, jeho jednotlivých heraldických součástí pomocí nástrojů počítačového vidění. V první kapitole jsou popsány základní prvky heraldiky. Poté jsou popsány vybrané oblasti počítačového vidění, především pro segmentaci a detekci objektu v obraze. V následujících kapitolách jsou podrobně popsány konvoluční neuronové sítě, návrh výsledného systému a jeho implementace. Nakonec následuje popis ověření funkčnosti a provedených experimentů.

Kapitola 2

Heraldika

V této kapitole je nejprve krátce uvedena definice a historie heraldiky. Poté je popsán samotný znak a následující podkapitoly jsou zaměřeny na jeho jednotlivé součásti. Uvádějí, kde se tyto součásti, v erbu nachází, jak se rozeznají a definují jejich další dělení.

Heraldika je velmi obsáhlý vědní obor, tato kapitola je však výhradně zaměřena na heraldiku rodovou. Ostatní odvětví heraldiky popisována nejsou vůbec nebo jsou jen okrajově zmíněny.

2.1 Definice heraldiky

„Heraldika jest nauka, obsahující souhrn základních pravidel a zvyklostí, dle nichž znaky tvořeny býti mají, a učící, kterak je popsovati a určovati“ [37]. Takto definoval heraldiku na začátku minulého století Vojtěch Král z Dobré Vody. O čtyři desetiletí později jí Karel Schwarzenberg popisuje jako, *„heraldika je nauka o znacích: totiž o učení o nich a jejich dějinách, a umění je tvořiti. Proto je heraldika zároveň věda i umění, nebo – skromněji řečeno – stojí na pomezí historických věd a výtvarných umění; dobrý heraldik je však zároveň uměncem a malířem“* [30].

Spojení těchto dvou definic dohromady plně vyjadřuje smysl a poslání heraldiky: první z nich poukazuje na její řád a pravidla, druhá se dotýká jejího vědeckého a uměleckého založení. Heraldika se jako věda opírá o vědecká bádání, ačkoliv má tak charakteristickou vědní činnost, že ji nemůžeme zařadit do žádné jiné historické vědy. I přes pevná, staletí vytvářená pravidla a povinnosti, nechává stále dosti prostoru, aby kreslíř mohl využít své malířské či grafické umění. [37]

2.2 Historie heraldiky

Většina z nás má znaky spojené se vznešenými rytíři jedoucými do bitvy či na turnajové kolbiště. Jejich původ je však o celá staletí starší a pochází ze zemí položených směrem na východ. Například v Knize králů se dočteme o korouhvích perských bohatýrů ozdobených lvy, tygry, levharty, vlky a dalšími zvířaty. Právě z Persie se znaky rozšířili do všech světových stran, dále na východ, kde se dále obohacovala. Poté na sever ke kočovníkům a od nich dále ke Keltům a Germánům.

Antické civilizace již měly znaky pro své bohy (Diův orel, drak Apollona atd.), ale východní vliv k nim pronikl až po římském dobytí Malé Asie, Sýrie a válkami s dalšími východními zeměmi. Z těchto zemí se pak různými znaky zdobily vzácné tkaniny, které si

o několik staletí později kupovali rytíři na křížových výpravách, a nejspíše toutéž cestou se dostaly do Evropy. Znaky měly velký význam také v Byzantské říši, kde se jimi označovaly vysoké státní funkce nebo vojenské hodnosti. Jedním z takových symbolů byl i bílý lev a jedna z teorií, jak se bílý lev dostal do našeho znaku je, že když byl druhý český král Vladislav I. leníkem v Cařihradě, možná dostal za odměnu nějaký dvorský úřad a později si ze symbolu toho úřadu udělal vlastní znak [30].

Jak již bylo dvakrát zmíněno, heraldika se do Evropy dostala při křížových výpravách, kde se začaly obrazovými znaky označovat prapory vojenských oddílů a jednotlivci si je malovali na štíty. Po skončení křížových výprav se heraldika začala využívat v lenním zřízení, kdy svobodné osoby skládaly králi nebo jinému vladaři lenní přísahu a přísahali mu věrnost. Za to dostali půdu a popřípadě dvorní úřad. Lenní závislost nejdříve končila smrtí vazala, až později se stala dědičnou. V nejstarších dobách si vazalové své znaky vytvářeli sami, až později od roku 1340 [30] vznikaly první erbovní listiny (obrázek 2.1) a znaky byly udílány přímo panovníkem nebo jiný nositel práva udílet znak.



Obrázek 2.1: Erbovní listina Auduease Gepleise z Radkersburku z 15. 9. 1624. [2]

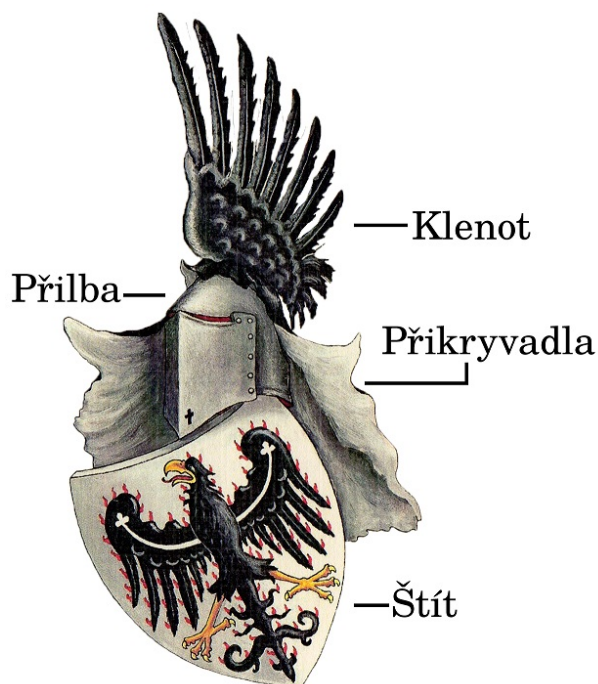
Rozkvět heraldiky dal také příčinu vzniku heroldům, kteří ve své době byli největší znalci heraldiky. Jejich povinností bylo svého pána představovat, svolávat, ale také museli znát velké množství znaků, aby mohli svému pánovi říci, který znak patří, kterému šlechtici.

Časem heraldika pronikla také do městského zřízení, své znaky také dostaly cechy a později dokonce i zámožnější měšťané. Po rozšíření palných zbraní a následném úpadku rytířství a turnajů se erby přesunuly už jen na listiny. Po zrušení šlechtictví se v některých zemích heraldika skoro úplně ustoupilo. V naší republice se do státního znaku dostali neheraldické prvky, které byly odstraněny až po sametové revoluci. Heraldika se využívá dále ve znacích měst a obcí, státní správě a v církevních znacích.

2.3 Znak

„Znak (erb) je obrazec vytvořený podle heraldických pravidel a jako trvalé znamení hodnosti, moci, cti nebo práva osob, rodů a zemí a později i měst a rozličných společenství.“ [37]

Původně znak tvořilo jen znamení namalované na nějaké ploše nebo na štítu. Základ tedy tvoří štít s takzvanou heraldickou figurou. Až s dalším rozvojem heraldiky se začaly přidávat další součásti jako přilba, klenot, přikryvadla, točenice nebo koruny. Tedy takové části, které se využívaly při rytířských turnajích v dobách jejich největšího rozkvětu (velký klenot byl ve válkách nepraktický). Erb s takovými součástmi můžeme vidět na obrázku 2.2.



Obrázek 2.2: Rozdělení erbu Přemyslovců na jednotlivé součásti. Kresba Vladislav Dědek.

V pozdějších dobách se ke znaku začaly přidávat další součásti jako řádové dekorace, hodnostní odznaky, štítonoši, hesla a pokřiky, pláště, stany, klobouky, koruny a čepice. Těmto součástem říkáme honosné kusy a v této práci jsou dále více popsány jen poslední dvě zmíněné, protože s těmi se u rodových erbů můžeme setkat nejčastěji.

2.3.1 Kovy, barvy a kožešiny

V české heraldice se společně nazývají tinkturami a slouží k popisu barevnosti znaku.

Kovy v heraldice známe pouze dva, a to ty nejvzácnější, zlato a stříbro (znázorňovány žlutou a bílou barvou).

Barvy jsou čtyři základní: červená, modrá, zelená a černá. Později se přidala i barva nachová, oranžová, hnědá, ocelová a jiné. Používala se také barva přirozená např. pro části lidských těl, nebo když se řeklo strom přirozené barvy, bylo zřejmé, že je myšleno, že kmen a větve jsou hnědou barvou a lisy barvou zelenou.

Nicméně existují i zbarvení, které nelze popsat barvou ani kovem. Z tohoto důvodu vznikly kožešiny. Jedna je hermelín, který se kreslí jako bílé pole posetými černými ocásky.

Druhá kožešina je popeličina, která znázorňuje k sobě sešité kůže. Kreslí se jako mozaika k sobě obrácených bílých a modrých zvonů.

V heraldice jsou dvě základní pravidla o barvách a to, že nikdy nesmí být kladena barva na barvu a kov na kov. Druhé z nich je však někdy porušováno a po vzniku kožešin přibilo ještě pravidlo, že kožešina nesmí býti kladena na kožešinu.

2.4 Štít

„Hlavní součástí znaku je štít – neboť bez něho nemůže být znaku.“ [30]

Jedná se tedy o jedinou povinnou součást znaku. Znak se sice kreslí také na korouhve nebo na oděv, ale v těchto případech se vždy jedná jen o náhradu štítu.

Tvar štítu nepatří ke znaku, ale na vůli malíře. Ten si mohl volit jak podle slohu, ve kterém pracoval, podle období, do kterého majitel erbu patřil (tvary štítu se také vyvíjely průběžně) nebo může mít jednoduše pár svých oblíbených tvarů a dalšími se nezaobírá.

Základní tvary štítu jsou zobrazeny na obrázku 2.3. Existuje však mnohem více tvarů štítů. Jen [37] uvádí dále anglický, jugoslávský, routový, renesanční a barokový.



Obrázek 2.3: Tvary štítů. Zleva: normanský, gotický, kolčí štít (starším názvem terč), španělský, francouzský, polský a italský. [30]

Štít kreslíme většinou kolmo, ale někdy ho můžeme kreslit i nakloněný (toto naklonění vychází ze způsobů zavěšení štítu na řemeni). Štít však můžeme naklonit jen a pouze tehdy, stojí-li na něm jen jediná přilba. V tom případě štít můžeme naklonit před a na zadní roh postavíme přilbu.

Štít je také možno dělit a tím se jeden znak může skládat z více znaků. Můžeme štít rozdělit rovnými čarami nebo částmi kružnic na několik částí. Zvláštní dělení jsou takzvané štítky (první erb zleva na obrázku 2.7), kdy do štítu vložíme další malý štít.

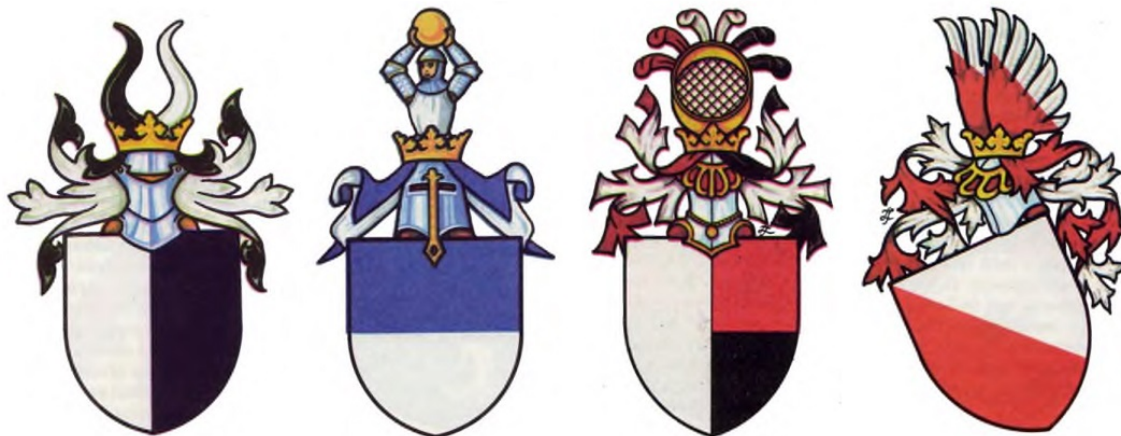
Štít je nositelem různých obrazců, které nazýváme figury a můžeme je dělit na heroldské figury a obecné figury. Přejít mezi nimi tvoří kříže, kterými se však tato práce nezaobírá.

2.4.1 Heroldské figury

Heroldské figury vznikají rozdělením štítu na více částí jednoduchými nebo složitějšími geometrickými členěními. Toto členění nazýváme různě a může být provedeno různými čarami. Každá takto osamocená část štítu je vyplněná jinou tinkturou. Podle [37] můžeme štít dělit na:

- *Polcený* (půlený – svisle, kolmo). Polcení je možno provést čarou rovnou, lomenou, křivou, cimbuřovou, špicemi a podobně.
- *Dělený* (vodorovně). To můžeme provést čarou rovnou, pilovou, vlnitou, plamennou atd.
- *Polcený* a dělený. Kombinace dvou předchozích. Například polcený a od středu dělený nebo dělený a od středu polcený.

- *Kosmo nebo šikmo rozdělený*. Může být opět provedeno různými čarami.
- *Obruba (lem)*. Táhle se podél celého nebo části štítu. Většinou o šířce $1/7$ šířky štítu.



Obrázek 2.4: Heroldské figury. Erby zleva: Chlumčanští, Údrčtí, Hrzánové a Krajířové. Erby převzaty z [17].

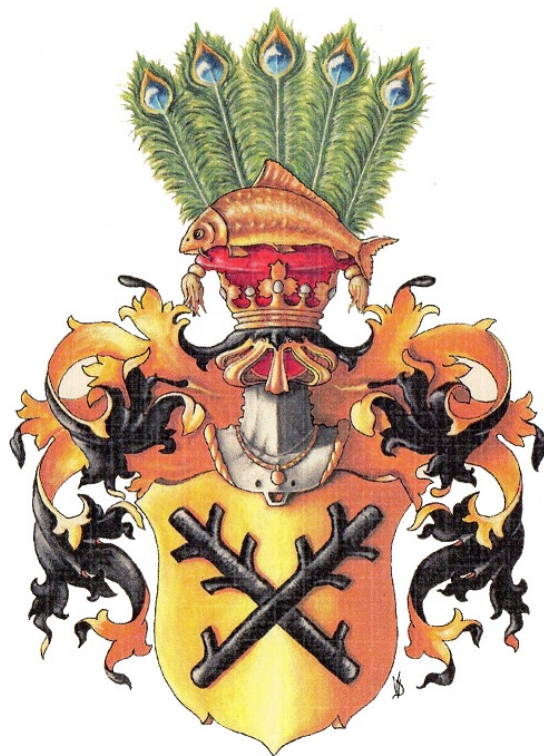
První čtyři druhy dělení můžeme vidět na obrázku 2.4 zleva. Některé speciální dělení mají svůj vlastní název. Dvoje polcení, přičemž mezera mezi nimi má šířku zhruba $2/7$ štítu, se nazývá kůl, při dělení se nazývá břevno. Pokud použijeme kůl i břevno naráz, získáme heroldský kříž. Když štít dělíme ve výšce $2/7$ od horního okraje štítu, nazýváme tuto část hlava, pokud od spodního okraje, tak pata. Zmenšíme-li vzdálenost na $1/7$ získáme poloviční hlavu a patu. Zelená poloviční pata se dále nazývá trávník. Rozdělíme-li polcením i dělením štít na 16 částí, získáme šachovaný štít. Takových speciálních dělení existuje ještě mnohem více v [30, 37] se můžeme dočíst o čtvrti, čelu, šindelovaném štítu, cihlovaném štítu a dalších děleních.

2.4.2 Obecné figury

Jedná se o heraldická znamení převzata z našeho okolí, a právě v jejich vyjádření spočívá největší půvab heraldiky [37]. Obecných figur tedy může být obrovské množství i přes to, že ne vše se dá použít jako erbovní znamení, ale musí se dbát na určitý vkus, „kromě předmětů svou povahou hnusných vše může být erbovním znamením“ [30].

Obecná figura by měla vyplnit celou plochu štítu nebo v případě děleného štítu celou plochu pole a neměla by se alespoň na dvou stranách dotýkat okraje a zobrazuje se většinou s pohledem k pravé straně štítu. Můžeme je dle [30] rozdělit do několika skupin.

První skupina jsou lidé. Není však možné jako figuru použít kohokoliv, protože jiný kreslíř by nemusel vědět, jak tuto postavu nakreslit. Z tohoto důvodu se volí postavy křesťanského náboženství, tedy Bůh Otec, Ježíš Kristus, Duch Svatý, Panna Marie a různé další svatí nebo světice. Další skupinou jsou zvířata, ze kterých vidáme nejčastěji lva, který se kreslí většinou ve skoku nebo orlici (obrázek 2.2 – plamenná orlice). Z šelem se používá dále tygr nebo divoký kanec, z býložravců je to často kůň, býk nebo kozel. Z ptáků se kromě orlice vyskytují i orlové, sokolové, čápi, pávi, krkavci či havrani. Dále mezi zvířata patří různé ryby, plazi nebo mořská zvířata.



Obrázek 2.5: Erb Pánů z Lipé. Kresba Vladislav Dědek.

Svojí vlastní skupinu mají nestvůry. Z této skupiny vídáme draka, který může mít spoustu podob, může chrlit oheň, mít roztažená křídla nebo různý počet hlav či nohou. Dalšími zástupci z této skupiny jsou bazilišek, gryf, jednorožec (obrázek 2.1), siréna, kentaur a další. Velkou samostatnou skupinou, jsou také rostliny, z nich nejčastěji vídáme růži a lilie. Vyskytuje se také srdcový list, trojlist, kopřivový list, granátové jablko nebo ostrev (obrázek 2.5).

Můžeme také kreslit různé druhy nástrojů, ať válečné, mezi něž patří meč, šíp, kotouč nebo předměty selské jako srp, kolo, korbel, žebřík a mlýnský kámen. Jako erbovní znamení můžeme použít i stavby. Z těch se vyskytují nejčastěji ty, které mají, co do činnosti s válkou jako hradba, zeď s cimbuřím, věž, hrad.

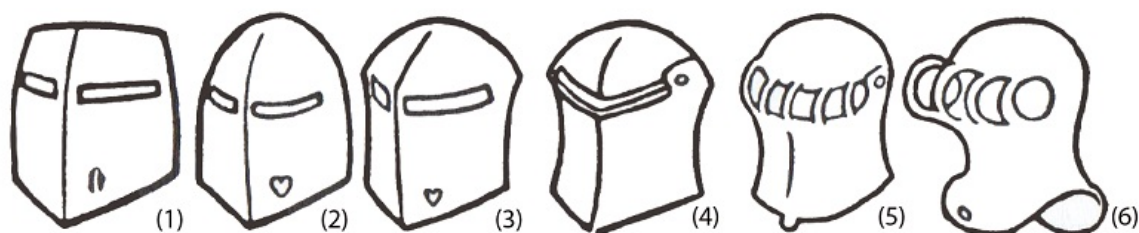
Z přírodních jevů vídáme ve znacích slunce, měsíc, hvězdy, blesk, plamen nebo hory. Poslední a speciální skupina jsou jiné předměty, které se nadají jednoduše zařadit. Jedná se o rukáv, šachový kůň a klévkové kolo (kolo z osmi kopí zakončené liliemi).

2.5 Přilba a její výstroj

Jak již bylo zmíněno, štít může být sám o sobě znakem, ale mohou být k němu přidány další součásti a nejčastější z nich je přilba s klenotem.

Tvary přileb jsou zobrazené na obrázku 2.6. Přilba turnajská (také obrázek 2.5) byla vyhrazená pouze šlechtě a měšťané ji využít nemohli. To však neznamená, že by šlechtic musel použít turnajskou přilbu, ale může si zvolit, kterou chce.

U nás se přilby malují stříbrné nebo ocelové, popř. se zlatými ozdobami, ale pouze přilby panovníků mohou být celé zlaté.



Obrázek 2.6: Tvary přileb. První tvar přilby byl hrncový (1), z něj se vyvinula přilba kbelíková (2, 3) a z té následně přilba kolčí (4). A z té se nakonec vyvinula přilba turnajská (5, 6). [30]

Přilbu můžeme postavit různými způsoby. Již bylo zmíněno, že pokud je štít nakloněn, může na něm být pouze jedna, a to na horním rohu (obrázek 2.2 a čtvrtý erb na obrázku 2.4). Je-li štít vzpřímen a přilba hledí vpřed, umísťuje se většinou doprostřed. Hledí-li však do strany, umístí se na zadní roh. Pokud je přileb více, štít musí být vzpřímen a přilby rozmístěny rovnoměrně hledící dovnitř erbu (u lichého počtu přileb, prostřední hledí vstříc).

2.5.1 Přikryvadla

K přilbě nezbytně patří přikryvadla, která se vyvinula z látkových povlaků, které chránily přilbu před sluncem, deštěm, ale také sekem mečem (z toho důvodu sahaly až na ramena). Nejdříve se kreslila skromně, jak můžeme například vidět na obrázku 2.2, ale později se začala třepit a rozstříhávat a přikryvadla se kreslila čím dát tím bohatší, jak můžeme vidět na obrázku 2.1 a 2.5.

Přikryvadla přebírají barvu od klenotu. Nejprve se kreslila jednobarevně, ale časem se kreslily dvojbarevná, ještě později i více barevná, ale stále jen těmi barvami, které jsou v klenotu. Pouze pár erbů má výjimku a volí se jiná barva přikryvadel.

Pokud je ve znaku přileb více, má vždy každá přilba vlastní přikryvadla.

2.5.2 Helmovní koruna a točence

Každý klenot se upevňuje přímo na přilbě a přikryvadlech. Po čase vznikla spojidla mezi klenotem a přikryvadly, které mělo za účel zakrýt jejich připevnění na přilbu. Kreslíři také usnadňují kreslení klenotu, protože nemusí toto spojení pečlivě kreslit.

První spojidlo vzniklo tak, že se vzala látka přikryvadla, stočila se kolem klenotu jako provaz a vinula se kolem vrcholu přilby. Toto překrytí se nazývá točenice a lze vidět na obrázku 2.1. Většinou se na točenici střídá kov a barva přikryvadla.

Druhý způsob překrytí je použít helmovní korunu. Ta se v počátcích vyskytovala pouze u panovníků. Časem je však začali používat i mocní velmožové a o několik staletí později se udělovaly znaky už jen s helmovní korunou. Tato koruna má tvar obrouče posázeného třemi listy, mezi nimiž mohou být perly jako na obrázku 2.5.

2.5.3 Klenot

Klenot může být stejně jako znak cokoliv vkusného. V našich zemích býval vztah mezi štítem a klenotem. Často se tedy obsah štítu použil jako klenot, většinou umístěný na nějakém dalším předmětu. Jako nejčastější klenoty se volí křídla (obrázek 2.2 a čtvrtý erb na obrázku

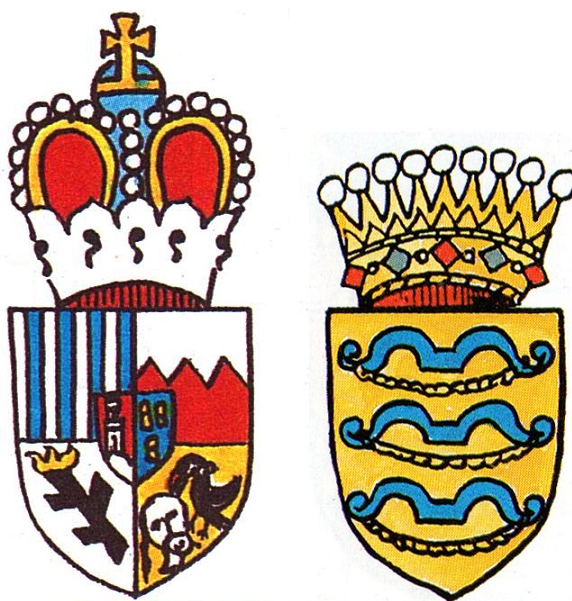
2.4), pera nebo rohy (třetí erb na obrázku 2.4). Některá znamení můžeme použít jako klenot přímo, jiná se musí nejprve upravit. Například velká zvířata musela být rostoucí (nebyla na nohách, ale lze vidět například jen hlavu a část trupu jako na obrázku 2.1), ale menší zvířata, vhodného tvaru jako například sedící kočka, svíjející se had nebo kapr mohou být podloženy jinou věcí, například polštářem (obrázek 2.5).

Při kreslení klenotu je velmi důležité, kam směřuje přilba, „za obecné pravidlo lze prohlásit: hledí-li přilba vstříc, nemůže klenot hledět na stranu“ [30]. Pokud přilba hledí vpřed a máme klenot, který nelze natočit stejným směrem (například lev) je nutné přilbu natočit tak, aby hleděla do strany.

2.6 Honosné kusy

Honosné kusy jsou všechny ostatní heraldické části, které je možné umístit do znaku. Jak již bylo zmíněno, patří sem štítonoši, hesla, pokřiky, pláště, stany, klobouky, koruny. Vlastní honosné kusy mají také znaky rytířských řádů (ty se dále lišily řád od řádu). Mezi ně patří například řádový řetěz, kříž nebo stuha. Znaky duchovních mají vlastní hodnostní odznaky jako berla, čepice, kardinálský klobouk, které označovaly postavení v církevní hierarchii.

Nás z pohledu rodových erbů však nejvíce zajímají koruny a čepice, které se (stejně jako klobouk) umísťují místo nebo zároveň s přilbou.



Obrázek 2.7: Knížecí a hraběcí erby. Zleva: Schwarzenbergové [22] a Arco z Jaroslavice [23].

Koruny se vyvinuly z antických věnců a nejprve se nosily bez přesného hodnostního významu. Až později vedle korun královských začala své koruny, o pěti listech, nosit vyšší anglická a francouzská šlechta. Nám však byla bližší říšská knížata, kterým byla dána koruna tvaru knížecí čepice nebo koruna o pěti listech a čtyřech perlách, a hrabata, u kterých je rozšířená koruna s devíti perlami.

Knížecí čepice bývají červené, podšité hermelínem se dvěma perlovými oblouky a říšským jablkem (kurfiřti měli perlové oblouky čtyři). Čepice nesmí být na nakloněném štítu a je vždy umístěna uprostřed.

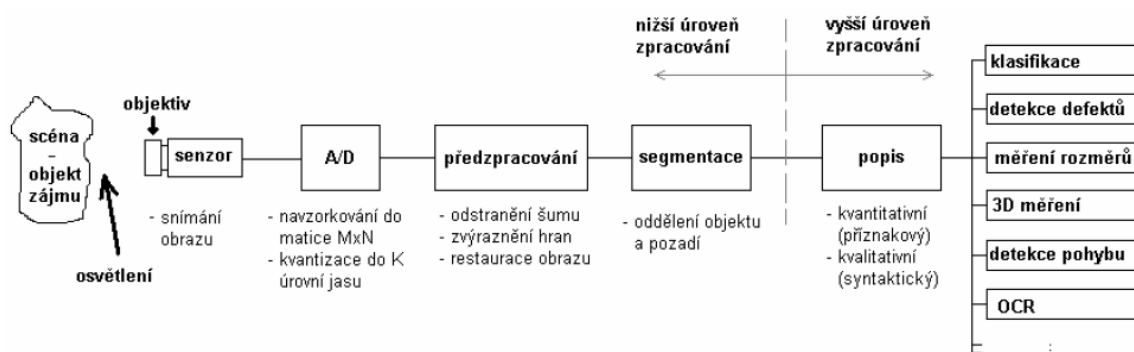
Obruče a listy všech korun (s výjimkou anglické šlechty) jsou obruče ozdobené drahokamy, listy nemají přesně definovaný tvar a mohou být ozdobené perlami. Pokud koruna k erbu přímo nepřiléhá, mezera se vyplňuje červenou barvou (první erb na obrázku 2.7). O položení koruny platí stejná pravidla jako u knížecí čepice.

V pozdějších dobách chtěly mít rody v erbu jako korunu (popř. čepici) a helmu zároveň. V tom případě, pokud vycházel z koruny plášť, umístila se přilba s klenotem na něj. Druhá možnost byla vložit korunu mezi přilby a spojit přikryvadly.

Kapitola 3

Vybrané oblasti počítačového vidění

Jak již bylo zmíněno v první kapitole, počítačové vidění se snaží nahradit naše lidské vidění. Nejedná se však o jedno komplexní řešení, ale spíše o soubor dílčích technik (oblastí), jejichž spojením můžeme lidské vidění nahradit. Základní řetězec zpracování obrazu můžeme vidět na obrázku 3.1.



Obrázek 3.1: Sekvence úkonů zpracování obrazu. [13]

Mezi základní techniky počítačového vidění patří segmentace obrazu a detekce objektů v obraze. V následujících podkapitolách je nejdříve stručně uvedeno snímání a předzpracování obrazu, následně jsou popsány základní oblasti počítačového vidění a pro každou z oblastí jsou uvedeny vybrané metody.

3.1 Snímání a předzpracování obrazu

Předtím než začneme obraz zpracovávat, musíme ho nejprve do počítače nasnímat a *digitalizovat* (převod z analogového signálu do digitálního). Obě tyto operace za nás provedou kamery, fotoaparáty, scannery atd. Takto získaný obraz je uložen do počítače v číselné formě, kde jeho vstupní informací může být jas nebo několik spektrálních složek (nejčastěji červená, zelená a modrá). Vstupní signál je popsán funkcí $f(x,y)$, kde proměnné odpovídají souřadnicím v obraze, který je vzorkován, kvantován a výsledkem je matice čísel popisující *digitální obraz*. Prvek matice nazýváme *obrazový element* nebo *pixel*. [27]

Dalším krokem ve zpracování obrazu je takzvané předzpracování, které má za cíl potlačit šum (vzniká při snímání nebo přenosu obrazu), odstranit zkreslené vlastnosti snímacího zařízení. Dále může zvýraznit nebo naopak potlačit jiné důležité rysy (například zvýraznění hran).

Při předzpracování obrazu se využívají nadbytečné údaje v obraze. Sousední elementy mívají často stejnou nebo velmi blízkou hodnotu jasu. Díky tomu můžeme například odstranit šum z obrazového elementu pomocí průměru hodnot jasu v okolí.

Podle Shannnovy teorie [24] nezískáme pomocí předzpracování žádnou novou informaci, jen a pouze některé zvýrazníme nebo potlačíme. Výsledek předzpracování je dán znalostí charakteru obrazu.

Typy předzpracování obrazu můžeme rozdělit na:

- *Bodové jasové transformace.* Pro úpravu jednoho pixelu se využívá informace, že jas v bodu výstupního obrazu závisí pouze na jasů bodu ve vstupním obraze. Používají se pro změnu jasové stupnice nebo vyrovnání histogramu jasu.
- *Geometrické transformace.* Využívají se při odstranění geometrického zkreslení, například u snímků družicových snímků Země.
- *Restaurace obrazu.* Pokud známe nebo dokážeme odhadnout poškození obrazu, můžeme jej potlačit. Často se zde využívá konvoluce nebo filtrace.
- *Matematická morfologie.* Geometrizuje vyhodnocení obrazu. Používá se pro odstranění šumu, zjednodušení tvaru, zdůraznění struktury objektu, popis objektu (plocha, obvod atd.).
- *Lokální předzpracování.* Slouží pro výpočet jasu bodu ve výstupním obraze pouze z lokálního okolí odpovídajícího bodu ve vstupním obraze. Pomocí metod spadajících do této skupiny můžeme vyhledat obraz, provádět filtraci, stření obrazu, detekce hran atd.

Výše uvedené typy předzpracování obrazu jsou detailně popsány v [24, 27].

3.2 Segmentace obrazu

„Segmentace obrazu je jedním z nejdůležitějších kroků vedoucích k analýze obsahu zpracovávaných obrazových dat. Snahou je rozdělit obraz do částí, které mají úzkou spojitost s předměty či oblastmi reálného světa zachyceného na obraze.“ [24]

Jak je zmíněno v úvodu podkapitoly, segmentace dělí obraz $f(x,y)$ na jednotlivé podobrazy $R_1, R_2, \dots, R_n$, tak že podobrazy dle [27] splňují následující kritéria:

1. $\bigcup_{i=1}^n R_i = f(x,y)$
2. $R_i \cap R_j = \emptyset, i \neq j$
3. Každý podobraz splňuje nějaké tvrzení nebo množinu tvrzení, např:
 - (a) všechny pixely v podobraze R_i mají stejnou úroveň šedi
 - (b) všechny pixely v podobraze R_i se neliší v úrovni šedi více než o danou hodnotu
 - (c) standardní odchylka úrovně šedi všech pixelů v podobraze R_i je dostatečně malá

Následující podkapitoly pojednávají o základních segmentačních metodách.

3.2.1 Prahování

Prahování je nejjednodušší segmentační metoda, která vychází ze skutečnosti, že mnoho objektů či oblastí obrazu je charakterizováno konstantní odrazivostí nebo pohlcení svého povrchu. Pak je možné využít určené jasové konstanty (prahu) pro oddělení objektů od pozadí. [24] Prahování je tedy transformace vstupního obrazu f na výstupní segmentovaný binární obraz g podle vztahu:

$$g(i, j) = \begin{cases} 1 & \text{pro } f(i, j) \geq T \\ 0 & \text{pro } f(i, j) < T \end{cases} \quad (3.1)$$

kde T odznauje zvolený práh a $g(i, j)$ nabývá hodnoty 1 pro obrazové elementy a 0 pro pozadí. Pro úspěšnou segmentaci je zásadní správná volba prahu. Jeho hodnotu můžeme určit interaktivně nebo pomocí metod pro automatické určování prahu. Globální práh je vhodný pouze pro případy, kde jsou objekty výrazně jasově odlišené od pozadí, z tohoto důvodu se častěji používá lokální práh. Více o metodách pro určení prahu se lze dočíst v [24].

3.2.2 Detekce hran

Segmentace na základě detekce hran jsou založeny na faktu, že hranice oblastí obrazu sestávají z hran, které jsou v obraze detekovány pomocí některého z hranových operátorů. Velkou škálu těchto operátorů můžeme nalézt v [24]. Velmi často se pro detekci hran využívá *Cannyho hranový detektor* [6].

Samotný binární obraz s detekovanými hranami je však pro segmentování objektů nepoužitelný a proto se detekované hrany spojují do řetězců nebo grafů, které odpovídají průběhu hranic jednotlivých objektů. Poté můžeme sledovat hranice, určovat vnitřní a vnější hranice oblastí nebo pokud známe tvar objektu, můžeme použít heuristické prohledávání grafu. [24]

3.2.3 Metoda rozvodí

U této metody je obraz chápán jako terén nebo topografický reliéf, který je postupně zaplaven vodou. Nejprve jsou v gradientním obraze nalezeny lokální minima, které se postupně začnou naplňovat vodou a okolní oblasti jsou postupně zatopeny. Pokud by v nějakém místě hrozilo, že se dvě různé povodí mohly slít, je vytvořena hráz. Zaplavení je zastaveno až v okamžiku, kdy hladina vody dosáhne nejvyšší hodnoty terénu (maxima v obraze). [27]

Výsledkem metody je obraz rozdělený do regionů podle povodí, oddělenými hrázemi. Nevýhodou metody je, že vytváří spoustu oddělených oblastí (hlavně v případech, kdy je v obraze přítomen šum) a regiony je potřeba následně spojit.

3.2.4 Aktivní kontury

Aktivní kontury (jinak snakes či active contours) je metoda založená na postupném tvarování kontury okolo hranic objektu v obraze. Kolem objektu se vytvoří kontura a ta je postupně tvarována okolo hranic objektů působením vnitřních, obrazových a vnějších sil. Vnitřní síly ovlivňují hladkost křivky, obrazové síly směřují tvarování kontury ke hraně objektů a vnější síly jsou výsledkem počátečního umístění kontury. [27]

Konturu můžeme reprezentovat jako sadu bodů $pn = [x_n, y_n]$ pro $n = 0, 1, \dots, N$.

Výsledná pozice kontura odpovídá její minimální energii dle vzorce:

$$E_s = \sum_{n=1}^N E_N\{p_n\} + \sum_{n=1}^N E_I\{p_n\} + \sum_{n=1}^N E_T\{p_n\} \quad (3.2)$$

Kde E_n reprezentuje vnitřní energii kontury (například ohyb nebo zlom), E_I je energie obrazu a E_T představuje energii omezení. Pro měření výše uvedených energií existuje mnoho druhů měření.



Obrázek 3.2: Tvarování aktivní kontury - zleva počáteční stav, standardní postup a modifikovaná verze algoritmu. [27]

Základní verze aktivních kontur při složitějších tvarech objektů nejsou schopny tvarovat konturu podél prudkým zlomům hranic objektů a může vytvářet smyčky, jak můžeme vidět uprostřed na obrázku 3.2. Z toho důvodu byly vytvořeny modifikované verze, které tyto nedostatky odstraňují.

3.2.5 Shluková analýza

Shluková analýza bývá většinou využívána při klasifikaci objektů na základě jejich podobnosti. Na stejném principu funguje i při segmentaci, kdy jsou pixely přiřazovány do shluků na základě jejich podobnosti.

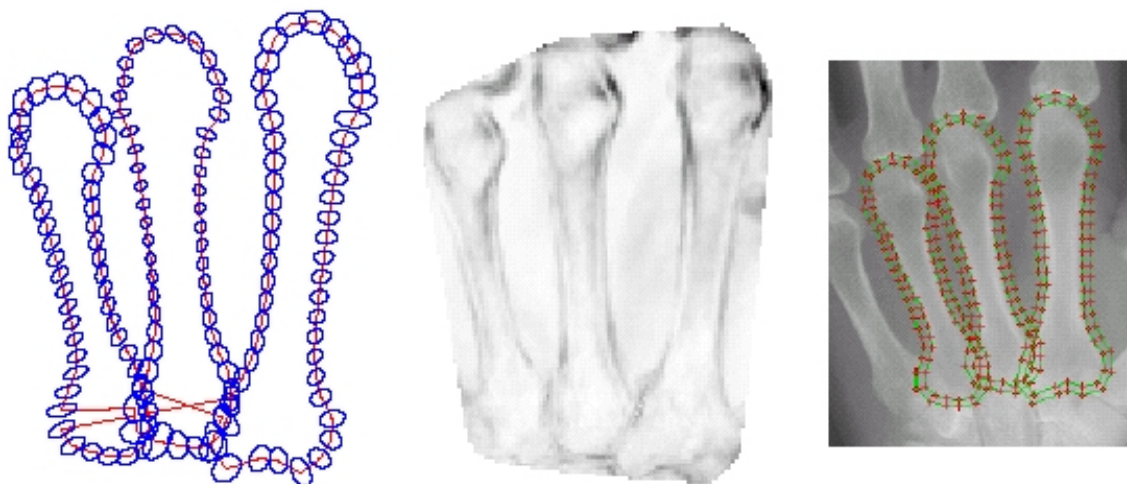
Každý pixel je reprezentován vektorem $x = [x_1, x_2, \dots, x_N]$, který obsahuje výsledky měření. Například pro charakteristické příznaky, jako je barva nebo jas. Příznaky volíme tak, aby byly v různých oblastech různě ohodnocené nebo aby byly vlastnosti okolí jako střední hodnota okolních pixelů, rozptyl apod. rozdílné. [27]

Při samotném shlukování se shluky spojují do shluků s nejbližším středem, pokud však jsou všechny středy existujících shluků daleko a jsou splněna příslušná kritéria, je vytvořen shluk nový. Nalezení kritérií pro vytvoření nového shluku i samotný výsledný počet shluků je netriviální. Výhodou je, že existuje spousta modifikací shlukovacích metod, které lze použít pro různé druhy úloh.

3.2.6 Statistické modely

Existují dvě metody, které se používají pro segmentaci objektu ve vstupním obraze pomocí statistických modelů. První z nich je *Active Shape Models* (ASM), která využívá pouze informaci o tvaru modelu [26] a druhá *Active Appearance Models* (AAM), využívá jak informaci o tvaru, tak informaci o textuře [27].

Obě metody využívají statistického modelu, který je vytvořen pomocí statistické metody zpracování dat, *PCA* (Principal Component Analysis) z manuálně segmentovaných trénovacích dat. Parametry takto získaného modelu se iterativně přizpůsobují novému obrázku. Ukázkou segmentaci pomocí AAM můžeme vidět na obrázku 3.3.



Obrázek 3.3: Segmentace záprstních kostí pomocí AAM. Nejprve je provedena nezávislá analýza hraničních bodů, poté analýza změn v textuře, a nakonec výpočet AAM modelu. [27]

Nevýhoda obou metod je že, příprava trénovací množiny spočívá v manuálním označování hraničních bodů, což může být u velké trénovací množiny velmi časově náročné. Při porovnání modelu s novým obrazem mohou obě metody selhávat, je velmi důležitá poměrně přesná inicializace (odhad polohy objektů v testovaném obraze).

3.3 Rozpoznávání syntakticky popsaných objektů

Pro rozpoznávání objektů máme několik možností. Nejčastější přístup je, že objekt segmentujeme a následně ho popíšeme vektorem příznaků, mezi které může patřit například délka, obvod, velikost, podlouhlost, nekompaktnost a další. Takto získaný příznakový vektor se následně předá natrénovanému klasifikátoru, který popisovaný objekt zařadí do správné třídy.

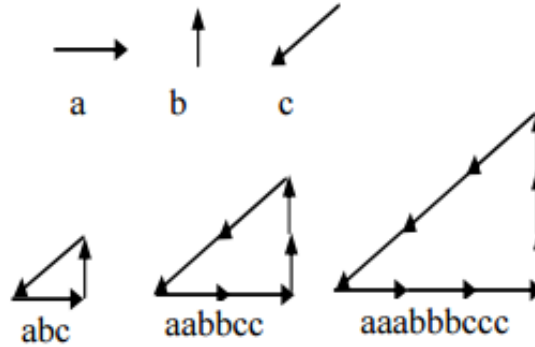
Tato podkapitola se zabývá méně častou metodou rozpoznávání objektů a to pomocí jejich syntaktického popisu. Následující text vychází z [35].

Syntaktické rozpoznávání je založeno na předpokladu, že rozpoznávané objekty je možné rozložit na jednoduché části, které se nazývají primitiva. Objekt je následně reprezentován obrazem, který vyjadřuje vztahy mezi primitivy popisovaného objektu.

Než rozdělíme objekt na primitiva, musíme určit množinu primitiv, ze které budeme vybírat. Pokud zvolíme malou množinu, nemusíme dobře popsat složitější objekt, pokud však zvolíme příliš velkou množinu, můžeme mít problémy při učení klasifikátoru. Dále musíme určit, jak budeme rozložení na primitiva popisovat. Nejčastěji se k popisu používá řetěz, strom nebo graf. Nejjednodušší je popis pomocí řetězu, na který se omezíme v dalším popisu.

Syntaktické rozpoznávání klasifikuje obrazy do obecně R tříd pomocí syntaktické analýzy. Obrazy (v tomto případě řetězové popisy) jsou chápány jako věty a primitiva jako množina terminálních symbolů. Pokud gramatika pro obrazy třídy r přijme všechny věty, které reprezentují obrazy dané třídy, a zároveň odmítne všechny věty reprezentující obrazy jiných tříd, můžeme klasifikaci převést na problém učení gramatiky.

Ukázkový příklad syntaktického rozpoznávání trojúhelníku zobrazený na obrázku 3.4. Předpokládáme tři primitiva popsaná symboly a , b , c , z nichž je možné poskládat různé velké trojúhelníky.



Obrázek 3.4: Příklad tří podobných trojúhelníků a jejich obrazů. [35]

Obrazy (věty) všech tří trojúhelníků přijímá a současně odmítá veškerých jiných objektů gramatika regulárního jazyka $L = \{a^n b^n c^n \mid n = 1, 2, 3\}$ a pomocí učení syntaktického analyzátoru nalezneme přepisovací pravidla.

Hledání přepisovacích pravidel se nazývá inference gramatik. Mějme regulární gramatiku $G(V_N, V_T, P, S)$, kde V_N značí množinu neterminálních symbolů, V_T množinu terminálních symbolů, symbol P množinu přepisovacích pravidel a symbol S ($S \in V_N$) startovací symbol. Abeceda V je dána sjednocením terminálních a neterminálních symbolů a symbol V^* označuje množinu všech slov nad abecedou V . Jazyk L je podmnožinou V^* a pro jazyk generovaný gramatikou $G(L(G))$ musí platit $L(G) = \{x \mid x \in V^*, S \rightarrow x\}$. Pro trénování, potřebujeme trénovací množinu $T = \{S^+, S^-\}$, kde S^+ je množina pozitivních příkladů (patřících do $L(G)$) a S^- je množina negativních příkladů (nepatřících do $L(G)$).

Trénování následně probíhá postupným průchodem trénovací množiny. Pokud je vybrán pozitivní příklad a gramatika ho negeneruje, upravíme ji tak, aby vybraný příklad generovala. Pokud by byl vybrán negativní příklad a gramatika ho generovala, upravíme ji tak, aby ho negenerovala. Podrobný popis inference gramatik můžeme najít v [24, 35].

3.4 Detekce objektu v obraze

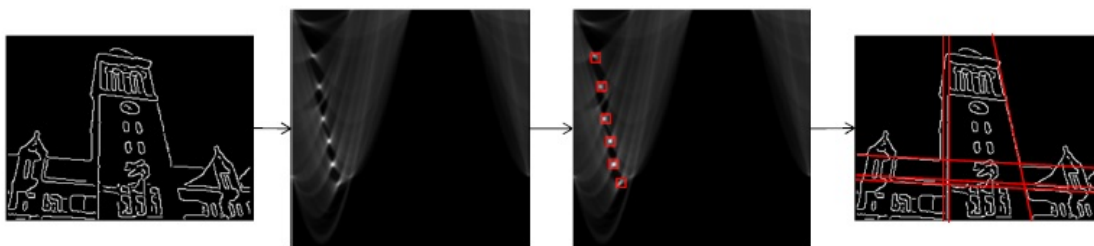
Pro některé úlohy počítačového vidění nám postačí objekt pouze detekovat, například systémy v moderních automobilech dokáží detekovat chodce nebo dopravní značky. V jiných úlohách se provede nejprve detekce a až následně segmentace, jako příklad můžeme uvést rozeznávání státní poznávací značky, kdy se nejprve detekuje v obraze a až následně je provedena segmentace jednotlivých písmen. Dalším důvodem pro detekci může být, že některé segmentační metody (ASM / AAM) potřebují znát polohu segmentovaného objektu.

Detekci objektů můžeme provést také tak, že celý obraz segmentujeme a jednotlivé oblasti klasifikujeme do tříd, což může být pro některé druhy úloh velmi pomalé. V této podkapitole jsou popsány metody, které se přímo zabývají detekcí objektů v obraze.

3.4.1 Houghova transformace

Houghova transformace je metoda pro nalezení objektů v obraze, u kterých známe jejich analytický popis. Nejčastěji se používá k nalezení přímk, kružnic, nebo elips. Jako předzpracování se používá nalezení hran v obraze. Výhoda této metody je odolnost vůči šumu v obraze.

Myšlenka této metody je dobře patrná na úloze detekce přímk v obraze. Přímka je dána dvěma body $A = (x_1, x_2)$, $B = (x_1', x_2')$. Všechny přímky procházející bodem A musí vyhovovat rovnici přímky $x_2 = kx_1 + q$. Protože x_1, x_2 jsou konstanty, můžeme rovnici chápat jako rovnici v prostoru parametrů k, q . Tam všechny přímky procházející v obrazovém prostoru bodem A reprezentovány přímku $q = x_1k - x_2$. Podobně můžeme všechny přímky procházející bodem B v prostoru parametrů reprezentovat přímku $q = x_1'k - x_2'$. Jediným společným bodem obou přímk v prostoru parametrů je bod (k, q) , který odpovídá v obrazovém prostoru přímce spojující body A a B . [24]



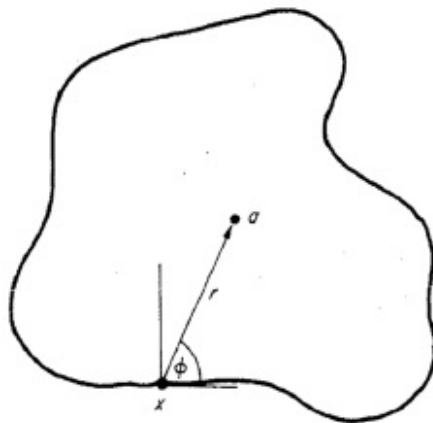
Obrázek 3.5: Nalezení přímk v obraze pomocí Houghovy transformace. [15]

Při implementaci se vytvoří n -rozměrná datová struktura čítačů (akumulátor) a pro každý pixel hrany se přičte k akumulátoru v prostoru parametrů 1 na pozicích odpovídajícím přímám, které mohou procházet daným bodem. Následně se vyhledají pozice maxim v akumulátoru a ty jsou námi hledané přímky (obrázek 3.5).

V některých případech nemůžeme rovnici hranice námi hledaného objektu parametricky vyjádřit. Pro tyto případy vznikla *generalizovaná Houghova transformace*. Popis hranice hledaného vzoru využívá odhad směru hranových segmentů.

Kdekoli v obraze umístíme referenční bod, kterým vedeme polopřímku směrem k hranici, a v průsečíku polopřímky s hranicí určíme směr hranice. (obrázek 3.6). Do tabulky si uložíme parametry referenčního bodu jako funkci směru hranice v průsečíku. [24]

Takto nalezené body uspořádáme podle směru do takzvané *R-tabulky*. Složitost narůstá s počtem transformací, ke kterému chceme, aby byla detekce invariantní. Pro posun jsou to dva parametry, další parametr musíme přidat pro změnu měřítka nebo rotaci.



Obrázek 3.6: Generalizovaná Houghova transformace. [15]

Na vzorovém obrázku si vytvoříme R-tabulku popisující hledaný tvar a datovou strukturu reprezentující potencionální referenční body. Pro všechny obrazové elementy, pro které je hodnota nad určeným prahem, určíme směr hrany a podle R-tabulky vypočítáme souřadnice potencionálního referenčního bodu pro všechny možné hodnoty velikosti a natočení a zvýšíme čítač tohoto bodu [24]. Následně najdeme maxima v čítači potencionálních referenčních bodů, čímž získáme polohu referenčního bodu, natočení a zvětšení.

3.4.2 Kaskádový klasifikátor

Detekce objektu v obraze probíhá tak, že po obraze posouváme a zvětšujeme *detekční okno* (sliding window) a každý výřez obrazu klasifikujeme. Klasifikujeme pouze do dvou tříd, které reprezentují, zda se ve výřezu nachází pozadí nebo námi hledaný objekt.

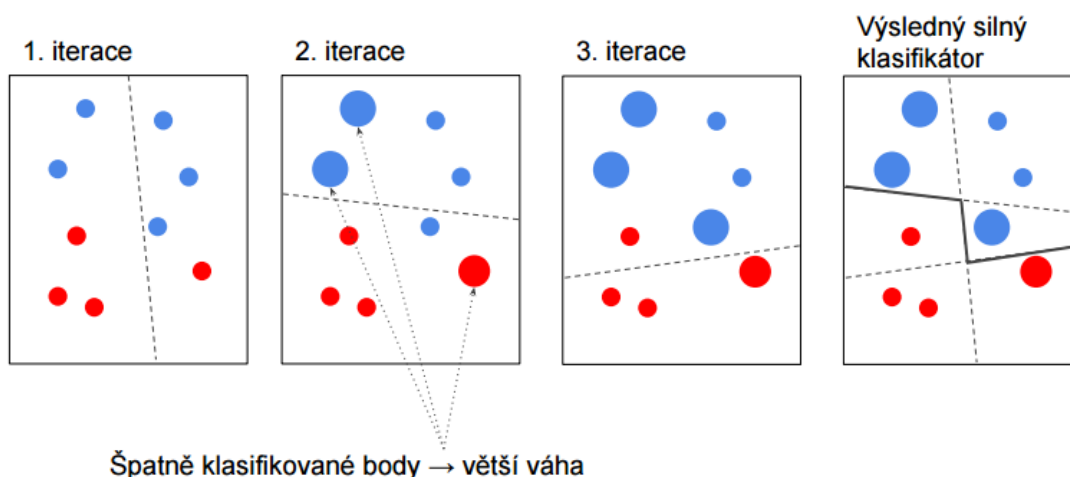
Kaskádový klasifikátor se skládá ze silných klasifikátorů složených do kaskády, kde jeden klasifikátor není schopen vyhodnotit, zda se jedná nebo nejedná o námi hledaný objekt, ale je schopen vyhodnotit, že se o něj nejedná a jde tudíž o pozadí, čímž se klasifikace ukončí. Až pokud výřez projde celou kaskádou, můžeme říci, že se v něm nachází námi hledaný objekt.

Silný klasifikátor je složen ze slabých klasifikátorů. To jsou jednoduché klasifikátory, které mají úspěšnost vyšší než 0.5 a jejich složením vzniká silný klasifikátor. Klasifikátory mohou pracovat s různými příznaky, často se používají Haarovy příznaky nebo Local Binary Patterns, které jsou popsány dále.

AdaBoost

Algoritmus AdaBoost slouží k učení silných klasifikátorů. Algoritmus má na vstupu množinu slabých klasifikátorů, pozitivní (obrazy, kde se nachází hledaný objekt) a negativní (obrazy, kde se nenachází hledaný objekt) množinu obrázků.

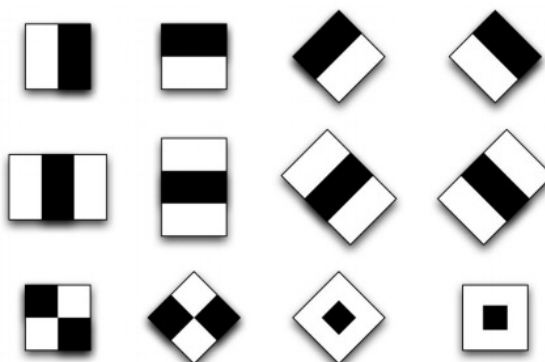
Při učení se nejprve nastaví váhy trénovacích dat na výchozí hodnotu a poté se v cyklu z množiny slabých klasifikátorů v závislosti na vahách, vybere ten, který dosahuje nejlepších výsledků. Pro vybraný klasifikátor se na základě jeho úspěšnosti vypočítá koeficient a upraví váhy trénovacích dat (špatně klasifikovaná data mají vyšší váhu). Takto postupuje, dokud nedosáhne dostatečné přesnosti. Průběh učení je znázorněn na obrázku 3.7.



Obrázek 3.7: Učení silného klasifikátoru pomocí algoritmu AdaBoost. [20]

Haarovy příznaky

Jedná se o velmi jednoduchý příznak používaný k popisu obrázku, který může mít spoustu různých tvarů (příklad na obrázku 3.8)



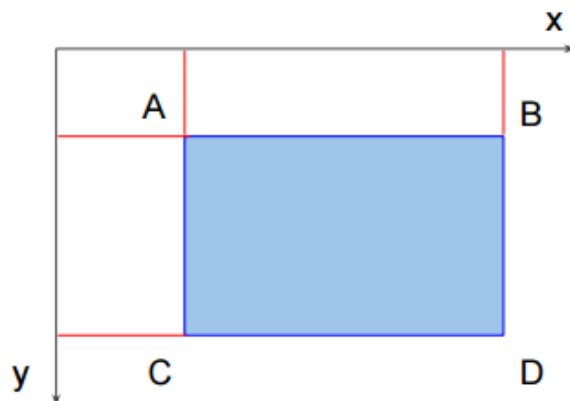
Obrázek 3.8: Příklad Haarových příznaků. [20]

Při výpočtu se počítá se rozdíl intenzity v regionech (bílá a černá barva). Tyto příznaky mohou být umístěny na libovolné pozici v detekčním okně a mohou mít různou velikost. Pokud by se musela u každého příznaku počítat intenzita pro jednotlivé regiony a ty od sebe odčítat, prováděli bychom několikrát ty samé výpočty a to by výsledný detektor zbytečně zpomalovalo. Z toho důvodu se obraz převádí na takzvaný *integrální obraz*.

Pomocí integrálního obrazu můžeme snadno a rychle vypočítat sumu intenzit pixelů původního obrazu v jakékoliv oblasti. Integrální obraz má stejné rozměry jako původní obraz a v každém bodě se nachází suma všech pixelů nad a vlevo od něj. Na obrázku 3.9 je tedy v bodě A suma intenzit pixelů v červeném obdélníku nad tímto bodem.

$$\text{suma intenzit pixelů v ABCD} = I(D) - I(B) - I(C) + I(A) \quad (3.3)$$

Sumu intenzit pixelů v obdélníku vymezeným body A,B,C,D vypočítáme podle rovnice (3.3), kde $I(A)$, $I(B)$, $I(C)$ a $I(D)$ jsou hodnoty integrálního obrazu v daných bodech [20].



Obrázek 3.9: Výpočet součtu intenzit pixelů v obdélníku vymezeným body A,B,C,D v integrálním obraze. [20]

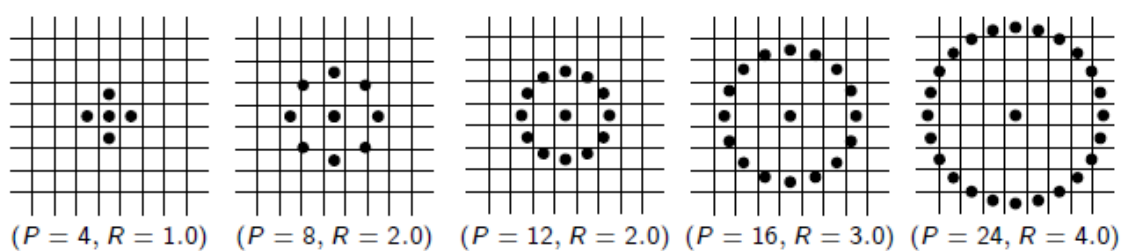
Local Binary Patterns (LBP)

Při vytváření tohoto příznaku se prochází všechny pixely, které mají minimálně určitý počet pixelů v okolí a pomocí funkce (3.4) [5] se vypočítá jeho hodnota. Výsledný příznak je dán histogramem hodnot. Jako parametry pro výpočet se používají parametr R , který určuje vzdálenost bodů od aktuálního pixelu, a parametr P , který udává počet bodů v jeho okolí.

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \quad (3.4)$$

Funkce $s(x)$ nabývá hodnoty 1, pro $x \geq 0$ a jinak 0, g_c je hodnota centrálního pixelu, pro který se počítá LBP hodnota a g_p značí hodnotu pixelu na pozici p .

Jako předzpracování obrazu postačí převést obraz do stupňů šedi. Následně se pro každý pixel uvažuje jeho okolí (obrázek 3.10) a nad všemi prvky se provede prahování a výsledek se vynásobí s danou váhou, čímž získáme LBP hodnotu pro centrální pixel.

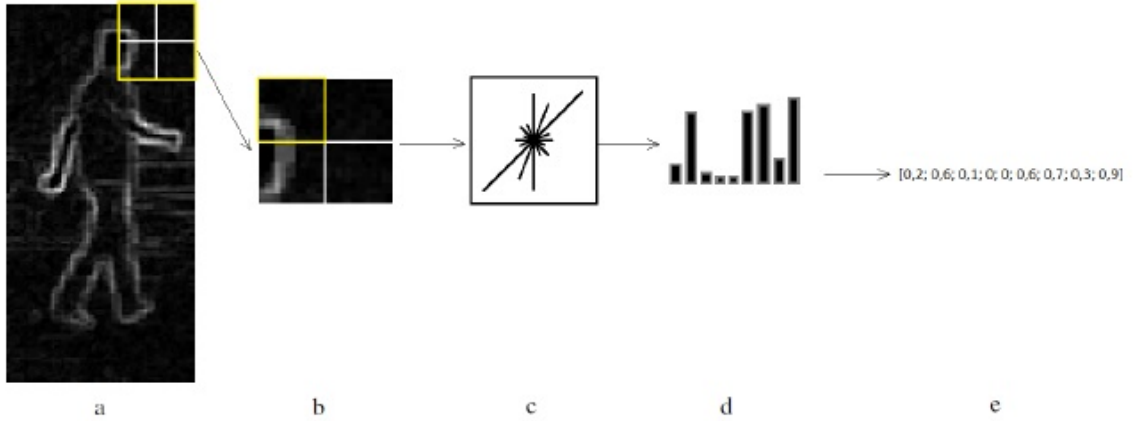


Obrázek 3.10: Příklady okolí pro výpočet LBP. [5]

Pro LBP můžeme velmi jednoduchým způsobem zajistit rotační invarianci. K tomu postačí, aby byla provedena bitová rotace a jako výsledek vybrána nejmenší hodnota.

3.4.3 Histogram orientovaných gradientů

Základní myšlenkou této metody je, že tvar objektu v obraze je možné charakterizovat pomocí intenzity gradientů. Tato metoda byla nejdříve použita pro detekci chodců [7]. Jako předzpracování postačí pouze redukce RGB barevného modelu do stupňů šedi. Samotný výpočet deskriptoru můžeme vidět na obrázku 3.11.



Obrázek 3.11: Ukázka výpočtu příznakového vektoru: vypočítané gradienty v obraze (a), jeden blok, obsahující 4 buňky (b), orientace gradientů v buňce(c), určení histogramu orientovaných gradientů v buňce (d), výsledný příznakový vektor (e). [21]

Pro celý obraz se vypočítají orientace gradientů pomocí derivační masky $[-1, 0, 1]$ v horizontálním i ve vertikálním směru. Pro výpočet je určeno p binů (kanálů) do kterých se dělí orientace gradientů. Například v [7] nabývají orientace gradientů hodnoty od $0^\circ - 180^\circ$ a dělí se do 9 binů, tedy na jeden bin připadne interval po dvaceti stupních. Obraz se rozdělí na buňky o předem dané velikosti. Následně se pro každou buňku určí histogram gradientů, gradienty se tedy rozdělí do jednotlivých binů podle jejich orientace.

V rámci celého obrazu se velikosti gradientů velmi liší, například kvůli nerovnoměrnému osvětlení. Z toho důvodu se zavedla normalizace. Ta je realizována tím způsobem, že jsou buňky sloučeny do bloků. Z jednotlivých buněk v bloku je vytvořen vektor histogramů a ten je následně normalizován. Využívá se normalizační schéma $L2 - Hys$, které je stejné jako $L2 - norm$, jež je dáno vztahem:

$$v \rightarrow \frac{v}{\sqrt{\|v\|_2^2 + \epsilon^2}} \quad (3.5)$$

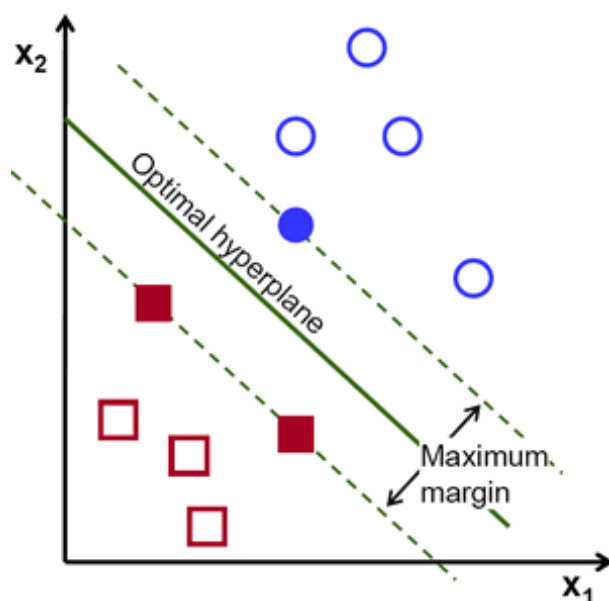
kde v je nenormalizovaný vektor obsahující všechny histogramy v určitém bloku a ϵ je malá konstanta, pak $\|v\|_k$ je k -norma pro $k = 1, 2$. $L2 - Hys$ Pouze oproti $L2 - norm$ limituje velikost každé složky vektoru na 0,2 [7, 21].

Konkatenací normalizovaných histogramů získáme *deskriptor*. Při detekci objektů podobně jako u kaskády klasifikátorů procházíme vstupní obraz pomocí detekčního okna. Detekční okno se skládá z určitého počtu bloků (okno může mít čtvercový nebo obdélníkový tvar) a postupně se posouvá po obraze o určitý počet buněk. Většinou se volí posun detekčního okna o jednu buňku. Konkatenace deskriptorů, získaných z bloků nacházejícím se v detekčním okně, se předloží klasifikátoru. Klasifikátor pro tuto metodu byl zvolen lineární SVM a výsledek klasifikace udává, zda výřez obrázku patří do třídy pozadí (této třídě

se většinou přiřadí hodnota -1), a nebo zda se v něm nachází námi hledaný objekt (třída s hodnotou 1).

Support Vector Machine (SVM)

SVM je metoda strojového učení používaná pro klasifikaci. Velká výhoda této metody je, že dokáže pomocí jádra transformovat nelineární klasifikační úlohu do prostoru vyšší dimenze a tam provést lineární separaci. Metoda hledá pro trénovací data rozdělovací nadplochu, která v novém prostoru, optimálně rozdělí vstupní data podle náležitostí do tříd. To znamená, že vzdálenost libovolných bodů patřících do rozdílných tříd je maximální. Toto rozdělení můžeme vidět na obrázku 3.12. Body ležící v nejbližším okolí rozdělovací nadplochy se nazývají *podpůrné vektory* (support vectors), podle kterých metoda dostala svůj název.



Obrázek 3.12: Ukázka rozdělení prostoru pomocí rozdělovací nadplochy. Tvary a barva jednotlivých trénovacích dat udává jejich klasifikační třídu. Vyplněné tvary značí podpůrné vektory a nevyplněné tvary jsou ostatní trénovací data. [3]

Uvažujme vstupní data ve tvaru $\{x_i, y_i\}$, kde x_i je vstupní vektor a y_i příslušná klasifikační třída (u binární klasifikace $y_i \in \{-1, 1\}$). Metoda vypočítá rozdělovací nadplochu a následně je schopná pro nové hodnoty predikovat do které ze tříd patří.

3.5 Matematická morfologie

Matematická morfologie je samostatnou oblastí analýzy obrazu, která úlohu zpracování obrazu geometrizuje. Zaměřuje se především na tvar objektů a transformace, které ho zachovávají. Následující text vychází z [24].

Matematická morfologie vychází z bodových množin a využívá se především pro předzpracování obrazu, zdůraznění struktury objektu nebo jeho popis číselnými charakteristikami.

Morfologické transformace jsou realizovány jako relace obrazu (bodová množina X) s jinou menší bodovou množinou B , které se říká *strukturní element*. Morfologickou transfor-

maci si můžeme představit, jako bychom strukturním elementem B systematicky pohybovali po obraze. Výsledek transformace je zapsán do okamžitého bodu obrazu.

Základními transformacemi matematické morfologie jsou dilatace, eroze, otevření a uzavření.

Dilatace skládá body dvou množin pomocí vektorového součtu a je definována jako:

$$X \oplus B = \{d \in E^2 : d = x + b, x \in X, b \in B\} \quad (3.6)$$

kde E^2 je Euklidův dvojrozměrný prostor se systémem všech podmnožin, X bodová množina reprezentující objekt, B je strukturní element.

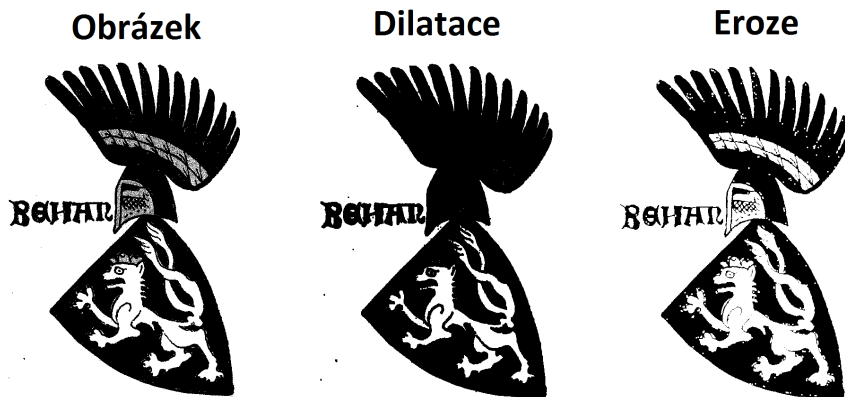
Dilatace se používá k zaplnění malých děr a úzkých zálivů, může však porušit souvislost objektu. Dilatace je dále komutativní, asociativní, lze vyjádřit jako sjednocení posunutých bodových množin, je invariantní vzhledem k posunutí a je rostoucí transformací.

Eroze skládá dvě bodové množiny s využitím rozdílu vektorů. Je duální transformací k dilataci (ne však inverzní) a je definována:

$$X \ominus B = \{d \in E^2 : d + b \in X \text{ pro } \forall b \in B\} \quad (3.7)$$

Eroze se používá pro zjednodušení struktury objektů. Objekty spojené tenkou čarou se mohou rozdělit. Pomocí eroze můžeme dále snadno nalézt obrys objektu v obraze tím způsobem že, odečteme od původního objektu jeho erozi.

Eroze je invariantní vůči posunutí a rostoucí transformace.



Obrázek 3.13: Ukázka dilatace a eroze. První zleva původní obraz (bodová množina X), následuje dilatace a eroze s strukturním elementem velikosti 3×3 pixelů (bodová množina B). U dilatace vidíme, že opravdu došlo ke ztrátě detailů především v přilbě, klenotu a u lva. U eroze je efekt opačný. Původní erb převzat z [32].

Ukázka dilatace a eroze je zobrazena na obrázku 3.13. Použitím erozi následovanou dilatací, získáme transformaci otevření. Dilatace následovaná erozí je uzavření.

Kapitola 4

Konvoluční neuronové sítě

Konvoluční neuronové sítě (CNN) jsou speciálním druhem neuronových sítí, které jsou určené pro zpracování dat s charakteristickou strukturou. Nejčastěji se využívají pro zpracování obrazu (klasifikace obrazu, detekce objektů v obraze), ale lze je také využít při zpracování řečových signálů.

Poprvé byly konvoluční neuronové sítě využity již v roce 1980 pro rozpoznávání ručně psaných číslic [9], ale větší pozornosti se jim dostává až od roku 2012 [19], kdy v soutěži ImageNet Large Scale Visual Recognition Challenge překonala všechny ostatní metody.

V této kapitole je nejprve stručně popsána matematická operace konvoluce, následně architektura konvolučních neuronových sítí, popis jejich učení a nakonec jejich využití pro segmentaci objektů v obraze.

4.1 Konvoluce

Konvoluce je matematický operátor, který zpracovává dvě funkce. Konvoluce dvourozměrných spojitých funkcí f a h se používá při lineárním přístupu ke zpracování obrazu a dle [27] je definována:

$$f(x, y) * h(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x - a, y - b) h(a, b) da db \quad (4.1)$$

kde funkci $h(x, y)$ nazýváme *konvolučním jádrem*.

Pro zpracování digitálních obrazů se používá *diskrétní konvoluce*, která je diskrétní dvojrozměrnou podobou předchozího vztahu [27]:

$$I(x, y) * h(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k I(x - i, y - j) h(i, j) \quad (4.2)$$

kde $I(x, y)$ je diskrétní obraz a $h(x, y)$ jádro konvoluce (konvoluční maska).

Při výpočtu diskrétní konvoluce můžeme konvoluční masku vnímat jako matici hodnot. Vstupní masku postupně posouváme po vstupním obraze a každou hodnotu pixelu překrytý maskou vynásobíme příslušnou hodnotou a nakonec všechny tyto násobky sečteme, čímž získáme hodnotu pixelu ve výstupním obraze. Výpočet je zobrazen na obrázku 4.1.

Část operací předzpracování obrazu se nazývá *filtrace*. Do této oblasti patří metody využívající pro výpočet jasů bodu ve výstupním obraze jen lokálního okolí odpovídajícího bodu ve vstupním obraze. Filtraci můžeme chápat jako diskrétní konvoluci. Konvoluční

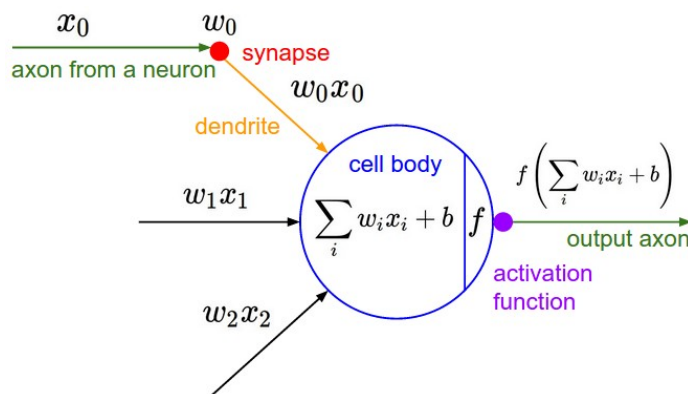


Obrázek 4.1: Konvoluce obrázku s maskou. [13]

lineární operace (*lineární filtry*) jsou použitelné pro vyhlazování, ale také pro detekci hran a dalších příznaků. [27, 24]

4.2 Architektura

Základní výpočetní jednotkou umělých neuronových sítí je umělý neuron (dále jen neuronová síť a neuron), které jsou stejně jako biologické neurony propojeny pomocí *synapsí*. Na obrázku 4.2 je zobrazen matematický model neuronu. Každý neuron přijímá signály skrze *dentrity*, které v těle zpracuje a skrze *axon* pošle po synapsích do dalších neuronů výstupní signál. [18]



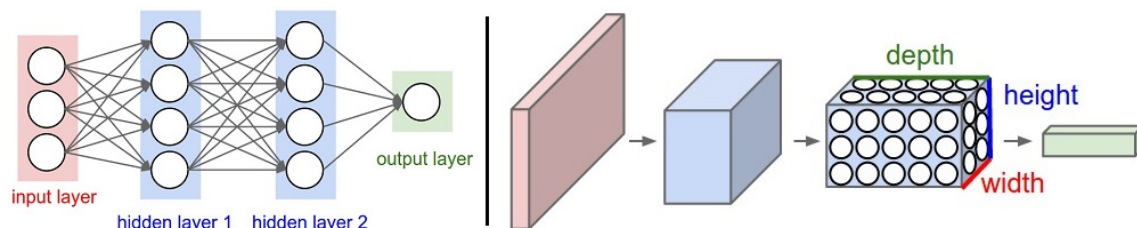
Obrázek 4.2: Matematický model neuronu. [18]

Každý vstup neuronu (x) má svou váhu (w), která určuje, jakou mírou se uplatní při výpočtu vnitřního potenciálu neuronu. Vnitřní potenciál se vypočítá takzvanou *bázovou funkcí*, kterých může být vícero druhů, ale nejčastěji se používá *lineární bázová funkce*, což je vážený součet vstupů plus *bias* (nedovolí síti při učení zůstat v lokálním minimu). Výstupní signál se vypočítá z vnitřního potenciálu pomocí *aktivační funkce*, zde opět můžeme zvolit z více možností, ale u konvolučních neuronových sítí se využívá *Rectified Linear Unit* (ReLU), která je definována jako [19]:

$$f(x) = \max(0, x) \quad (4.3)$$

Učení neuronových sítí spočívá v nastavování vah tak, aby se neuronová síť naučila řešit požadovanou úlohu.

Neurony se následně spojují do neuronových sítí. Těch existuje opět několik druhů, ale konvoluční neuronové sítě jsou *dopředné neuronové sítě*. Což znamená, že síť je rozdělená do vrstev a výstupy neuronů v jedné vrstvě mohou být napojeny jen na vstupy neuronů v následující vrstvě. Standardně se první vrstva nazývá *vstupní vrstva*, poslední *výstupní vrstva* a všechny ostatní mezi nimi se nazývají *skryté vrstvy* (obrázek 4.3 vlevo).



Obrázek 4.3: Ilustrace neuronových sítí. Vlevo klasická dopředná neuronová síť s plně propojenými vrstvy a vpravo konvoluční neuronová síť se třírozměrnými vrstvami. [18]

Vrstvy konvoluční neuronové sítě mají neurony uspořádané do tří dimenzí: do šířky, výšky a hloubky (obrázek 4.3 vpravo). Základní typy vrstev, ze kterých jsou sestaveny architektury konvolučních neuronových sítí, jsou popsány v následujících podkapitolách, které vycházejí z [18].

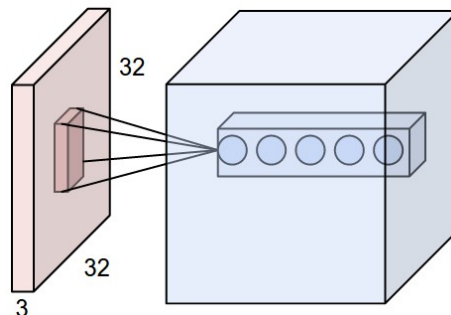
4.2.1 Konvoluční vrstva

Konvoluční vrstva je základním stavebním blokem konvolučních neuronových sítí a probíhají v ní nejnáročnější výpočty. Konvoluční vrstva je tvořena konvolučními filtry, které jsou prostorově malé, ale pokrývají všechny vrstvy vstupu. Například první konvoluční vrstva může mít velikost $5 \times 5 \times 3$, kde hloubka 3 je zvolena právě z důvodu, že v tomto případě se vstupní obraz skládá ze tří kanálů (červená, zelená, modrá). Při učení se síť učí právě parametry těchto konvolučních filtrů.

Při konvoluci přes celý vstup vrstvy vzniká pro každý filtr dvojrozměrná aktivační mapa, která udává odezvu uvedeného filtru pro každé prostorové umístění. Síť se naučí parametry filtrů tak, aby se aktivovaly, když uvidí nějaký vizuální příznak. Například hranu o určité orientaci, skvrnu dané barvy ve vstupní vrstvě a další. Filtry se posouvají o určitý krok, který se nazývá *stride*. Při konvoluci dochází ke zmenšení mapy. Abychom tomuto zabránili, používá se takzvaný *zero-padding*, kdy se okraje vstupní mapy doplní nulami tak, aby výsledná mapa měla stejnou šířku a délku jako vstupní mapa před rozšířením.

Při zpracování více dimenzionálních vstupů je nepraktické propojit všechny neurony s neurony v sousední vrstvě. Místo toho se propojí každý neuron pouze s lokální oblastí (obrázek 4.4). Prostorový rozsah tohoto propojení se nazývá *receptive field* (je ekvivalentní velikosti filtru).

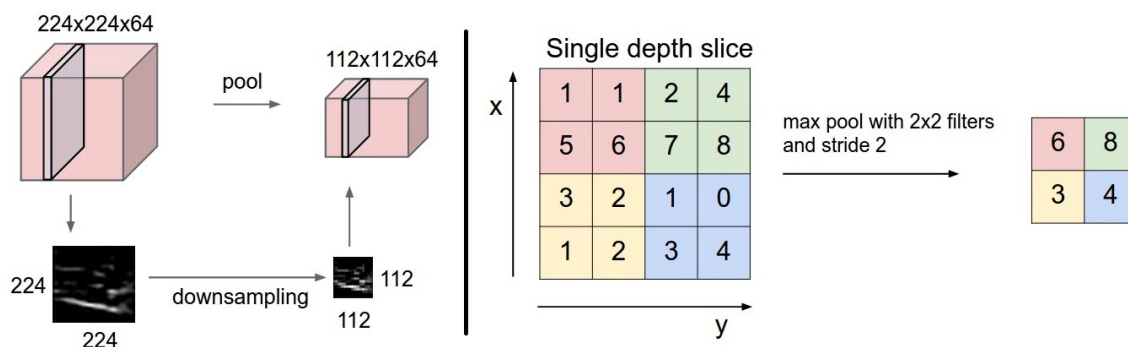
Pro snížení počtu parametrů se používá *sdílení parametrů*. Všechny neurony na stejné pozici v každém kanálu konvoluční vrstvy používají stejné váhy a bias (obrázek 4.4). Vychází se z předpokladu, že pokud je jeden příznak užitečný pro výpočet pro jednu pozici (x, y) , pak by měl být užitečný i v jiné pozici (x_2, y_2) .



Obrázek 4.4: Příklad projení neuronů v první konvoluční vrstvě. Každý neuron v konvoluční vrstvě je připojen pouze k lokální oblasti ve vstupním svazku neuronů po jeho celé hloubce (v tomto případě všechny barevné kanály). Všechny neurony na stejné pozici v každém kanálu mají sdílené parametry (5 zvýrazněných neuronů). [18]

4.2.2 Pooling vrstva

V architektuře konvolučních neuronových sítí se obvykle za konvoluční vrstvu vkládají pooling vrstvy. Jejich funkce je snižování prostorové velikosti (podvzorkování) vnitřní reprezentace, čím se sníží počet parametrů, rychlost výpočtu v síti a také pomáhá s problémem přetrénování.



Obrázek 4.5: Ukázka pooling vrstvy. Vlevo postupné podvzorkování každého kanálu vstupní mapy. Vpravo ukázka MAX pooling, kde je oblast rozdělena na nepřekrývající se oblasti o velikosti 2 x 2 a z každé je vybráno maximum. [18]

Pooling vrstva pracuje nezávisle v každém kanálu vstupu (obrázek 4.5 vlevo) a mění jeho prostorovou velikost pomocí vybrané operace. Používá se několik operací jako MIN (minimum), AVG (průměr), ale nejčastěji se používá operace MAX (maximum). Nejběžnější forma pooling vrstvy pracuje s filtry velikosti 2 x 2, které se posouvají s krokem (stride) 2 (obrázek 4.5 vpravo), čím se zmenší šířka a výška vstupní mapy na polovinu.

4.2.3 Plně propojená vrstva

Na konec konvoluční neuronové sítě se obvykle vkládá jedna nebo více plně propojených vrstev. Konečný výstup sítě se zpravidla následně upraví pomocí Softmax funkce, která vezme výstupní vektor sítě (z) o délce K (počet tříd) a pro každou třídu vypočítá s jakou pravděpodobností do ní vstupní data náleží. Softmax funkce je definována jako [18]:

$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}} \quad (4.4)$$

kde $f_j(z)$ znamená, že na pozici j ve výstupním vektoru softmax funkce je ohodnocení třídy s indexem j ve výstupním vektoru plně propojené vrstvy z . Výstupní hodnoty jsou v intervalu $<0, 1>$ a jejich součet je roven 1.

4.3 Učení konvolučních neuronových sítí

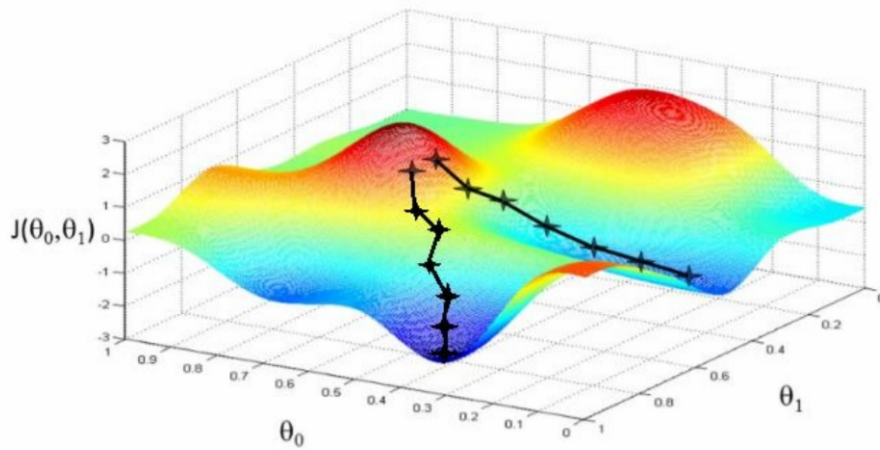
Když se v reálném životě učíme vykonávat nějakou činnost, měříme jak moc jsme v ní dobří a snažíme se neustále zlepšovat. Ve strojovém učení se postupuje velmi podobně, jen se neměří jak moc je model (v našem případě konvoluční neuronová síť) dobrý, ale naopak, jak moc je chybový. Říkáme tomu *ztrátová funkce* (loss function) a ta nám udává, jak se náš model liší od požadovaného výsledku a tuto chybu se snažíme postupně minimalizovat.

Základní funkce pro výpočet ztráty našeho modelu se nazývá *cross-entropy*, vychází z teorie o kompresi informací a dle [4] se definuje jako:

$$H_{y'}(y) = - \sum_i y'_i \log(y_i) \quad (4.5)$$

kde y je naše predikované rozdělení pravděpodobnosti do tříd a y' je správné rozdělení ve formě takzvaného *one-hot vectoru*, kde třída do které vstupní data náleží má hodnotu 1 a ostatní 0.

Pro minimalizaci funkce je nutné vypočítat její gradient, který udává ve kterém směru funkce roste. Pokud se bude dostatečně dlouho postupovat v opačném směru, tak se časem skončí v minimu (ilustrace na obrázku 4.6) a neuronová síť bude správně naučena.



Obrázek 4.6: Ilustrace algoritmu Stochastic Gradient Descent. Je zde zobrazen model neuronové sítě se 2 parametry a v prostoru parametru jsou hodnoty chyby neuronové sítě. Váhy sítě bývají inicializovány náhodně, na obrázku je ukázáno, že pokud začneme s výpočtem i na mírně rozdílném místě, můžeme se dostat do jiného minima. [14]

Pro učení konvolučních neuronových sítí se používá optimalizační algoritmus *Stochastic Gradient Descent*. Jedná se o iterační algoritmus, který pracuje s podmnožinou trénovací množiny a vypočítá gradienty parametrů sítě. Obrázky z použité podmnožiny trénovací

množiny se nepoužívají znovu, dokud nebyly pro trénování použity všechny obrázky z celé trénovací množiny. Jeden takový cyklus se nazývá *epocha*.

Výpočet jedné iterace probíhá tak, že při dopředném průchodu sítě spočítá a uloží aktivace všech neuronů. Následně se spočítá parciální derivace všech hodnot výstupních neuronů a při zpětném průchodu se rekurzivně spočítá parciální derivace (z nich se skládá gradient) všech neuronů v síti z parciálních derivací hodnot neuronů v následující vrstvě [14]. Nakonec se pomocí gradientu upraví v jednotlivých vrstvách (l) váhy (W) a biasy (b) podle vzorců [14]:

$$W_{ij}^{(l)} = W_{ij}^{(l)} - \alpha \frac{\partial}{\partial W_{ij}^{(l)}} J(W, b) \quad (4.6)$$

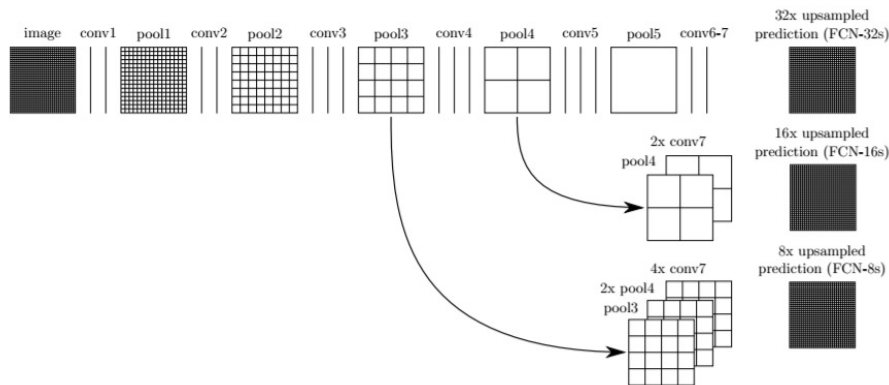
$$b_i^{(l)} = b_i^{(l)} - \alpha \frac{\partial}{\partial b_i^{(l)}} J(W, b) \quad (4.7)$$

kde α je konstanta, které se říká *koefficient učení* (learning rate), a udává s jakou mírou se uplatní vypočítané změny hodnot parametrů v opačném směru oproti gradientu a $J(W, b)$ je celková ztrátová funkce přes danou podmnožinu trénovací množiny.

4.4 Sémantická segmentace

Jak již bylo zmíněno, konvoluční neuronové sítě se v počítačovém vidění používají převážně pro klasifikaci obrazu, následně se začaly využívat pro detekci objektů, ale v této disciplíně nejsou už tolik rychlé.

V posledních letech probíhá výzkum, jak konvoluční sítě využít pro *sémantickou segmentaci*, což je technika, která klasifikuje každý pixel v obraze do příslušné třídy podle jeho sémantického významu.



Obrázek 4.7: Achitektury FCN sítě. FCN-16s kombinuje predikci z výstupní vrstvy sítě a s predikci z vrstvy *pool4* a upsampling s faktorem 16. FCN-8 přidává ještě predikci z vrstvy *pool3* a upsampling s faktorem 8. [8]

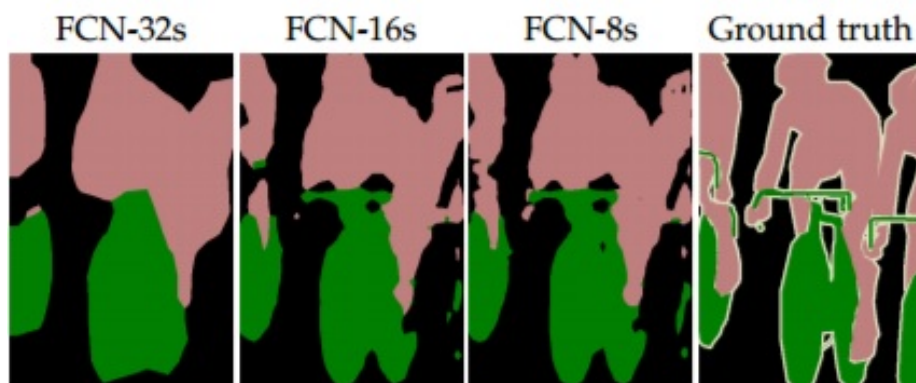
Autoři článku [8] převedli 16 vrstvou VGG síť (určená pro klasifikaci obrazu) na *plně konvoluční síť* (FCN) tím způsobem, že odstranili poslední klasifikační vrstvu, všechny plně propojené vrstvy převedli na konvoluční vrstvy a na konec přidali konvoluční vrstvu o velikosti $1 \times 1 \times 21$ (PASCAL dataset¹, na kterém byla síť trénována obsahuje 21 tříd).

¹<http://host.robots.ox.ac.uk/pascal/VOC/>

Výstup takto upravené sítě by byl pro sémantickou segmentaci nepoužitelný, protože je velmi malý. Proto následně autoři použili *upsampling* (převzorkování obrazu na vyšší rozlišení) s faktorem 32 (kolikrát se zvětší výstupní obraz) a tuto novou síť nazvali FCN-32s.

Výstupní segmentační mapa z této sítě byla však stále velmi hrubá. Proto autoři následně vytvořili další dvě sítě s faktorem upsamplingu 16 a 8. V těchto nových sítích pro výslednou predikci využili kombinaci jak výstupní vrstvy sítě, tak predikce z předchozích vrstev, na které použili upsampling (obrázek 4.7), tuto techniku pojmenovali jako *skip architektura*.

Původní FCN-32s síť byla trénována s využitím vah původní VGG-16 sítě (kromě poslední vrstvy). Síť FCN-16s byla trénována s váhami sítě FCN-32s a stejně tak váhy naučené FCN-16s sítě byly využity pro trénování FCN-8s sítě. Nejlepších výsledků dosáhli pomocí FCN-8s sítě, jak lze vidět na obrázku 4.8.



Obrázek 4.8: Porovnání výsledků FCN sítí. Ground truth je požadovaný výsledek. [8]

Autoři vyzkoušeli FCN sítě na několika datasetech a pro PASCAL VOC 11/12 dataset dosáhli 30% zlepšení oproti do té doby nejlepší metodě.

Kapitola 5

Návrh a implementace systému

V této kapitole je popsán návrh a implementace systému pro rozpoznávání a klasifikaci rodových erbů. Výsledný systém musí být schopný nalézt jeden nebo více erbů na jakékoli pozici v obraze. Dále musí být schopen nalezený erb dekomponovat na součásti a ty klasifikovat.

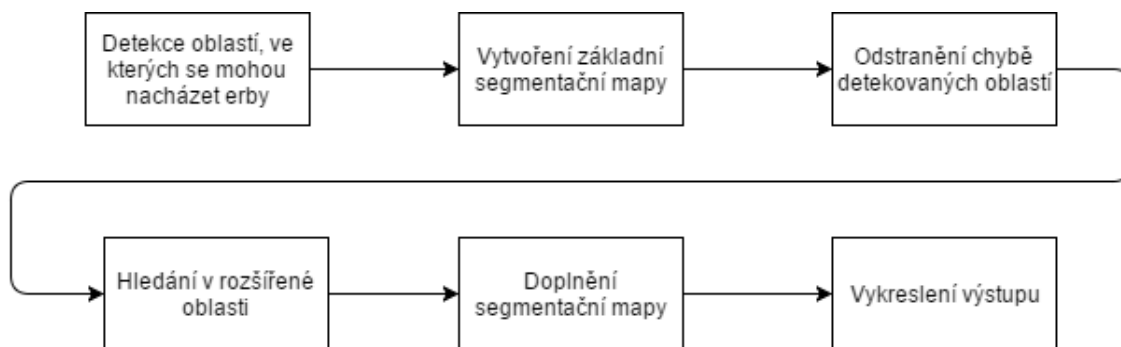
V první části kapitoly je popsáno rozdělení systému na jednotlivé části, dále je popsán výběr metod a v následujících podkapitolách je podrobně popsána implementace jednotlivých částí.

5.1 Rozdělení systému

Systém jsem rozdělil na následující hlavní části:

1. *Detektor štítu.* V prvním kroku jsou nalezeny oblasti, ve kterých se nacházejí erby. Postačí detekovat pouze štít, který je jedinou povinnou součástí erbu. Detektor může nalézt oblasti, ve kterých se štít nenachází, tyto oblasti budou později vyřazeny. Je výhodnější prozkoumat několik oblastí, ve kterých se erb nenachází, než oblast, ve které se erb nachází, vynechat.
2. *Segmentace a klasifikace štítu, přilby, helmovní koruny, točenice a honosných kusů.* Tyto součásti mají přesně definovanou podobu a bude snadnější je nalézt. Druhá část systému tedy segmentuje nalezenou oblast na vybrané součásti a provede klasifikaci štítu a přilby. Pokud by se nepodařilo štít segmentovat/klasifikovat, víme, že detektor štítu našel chybnou oblast a hledání v této části obrazu je ukončeno.
3. *Segmentace klenotu a překryvadla.* U těchto součástí není známý jejich přesný tvar a proto je složité je segmentovat. Je však známa jejich poloha vzhledem k ostatním součástem. Proto se tato část na základě získané segmentační mapy z předchozího kroku pokusí nalézt zbylé součásti. Tato část systému je použita pouze pokud se v segmentační mapě nachází přilba.

Na obrázku 5.1 je zobrazen celý návrh systému. Detekování oblastí, ve kterých se nachází erby by mělo probíhat tím způsobem, že se vyřízne malá oblast kolem štítu. Z pozorování nejrozličnějších kreseb erbů jsem vypožadoval, že postačí 2,1 násobek výšky štítu, abychom měli zaručeno, že se ve výřezu bude nacházet také přilba, točenice nebo helmovní koruna. Po získání segmentační mapy se součástmi se známou podobou dojde k vyřazení nevhodných



Obrázek 5.1: Návrh výsledného systému. První dvě části odpovídají popsaným částem výše. Mezi druhý a třetí krok je vloženo odstranění chybně detekovaných částí a rozšíření oblasti, protože není nutné první součásti hledat v tak velké oblasti jako je potom potřeba hledat přikryvadla a klenoty. Doplnění segmentační mapy odpovídá třetí popsané části.

oblastí. Hlavní kritérium by mělo být, že se ve výřezu nebude nacházet žádná nebo velmi malá oblast se štítem.

V dalším kroku bude následovat hledání zbylých součástí v rozšířené oblasti podle rozšířené segmentační mapy z předchozího kroku. Postačí hledanou oblast zvětšit na dvojnásobnou velikost, abychom mohli s jistotou předpokládat, že se v prohledávané oblasti budou nacházet obě zbylé součásti. Na závěr systém vykreslí segmentační mapu na vybranou oblast a pokud byl nalezen honosný kus, určí titul vlastníka erbu.

5.2 Výběr metod

V této podkapitole je popsán výběr metod, pomocí kterých budou implementovány hlavní části systému.

5.2.1 Metoda pro detekci štítu

Pro detekci štítu jsem vybíral z metod popsaných v kapitole 3.4. Kaskádový klasifikátor by zde nebyl vhodný, protože se v každém štítu objevují velmi odlišné figury a pro každý štít by byly jak Haarovy příznaky tak LBP příznaky velmi rozdílné.

Pro detekci štítu jsem vyzkoušel generalizovanou Houghovu transformaci. S ní jsem dosáhl nejen velmi špatných výsledků, ale především výpočet trval v řádu minut, což jsem vyhodnotil jako nepřijatelné.

Nakonec jsem zvolil metodu Histogramu orientovaných gradientů (kapitola 3.4.3), protože je vhodná pro detekci objektů podle jejich tvaru, což je pro řešenou úlohu ideální.

Metoda Histogramu orientovaných gradientů je invariantní vzhledem k posunutí, ale není invariantní vůči změně velikosti. Z toho důvodu bude nutné prohledat obrázek v několika rozlišeních. K tomu účelu se využívá takzvaná *pyramida obrázků*, kdy je nejprve obrázek prohledán v největším rozlišení a postupně se zmenšuje.

Nakonec je nutné souřadnice všech nalezených oblastí přepočítat tak, aby bylo možné výřez oblasti, ve které budou hledány součásti erbu, provést na obrázku s největším rozlišením. Pokud by štít zabíral velkou část obrázku a byl by tedy nalezen v obrázku s nejmenším rozlišením, tak by výřez z toho obrázku přišel o důležité detaily.

5.2.2 Metody pro segmentaci a klasifikaci jednotlivých součástí erbu

Vybrat správné segmentační metody bylo velmi složité, protože hlavní součást, štít, nesplňuje ani jedno z tvrzení popsaných v úvodu kapitoly 3.2, protože se zpravidla skládá z velmi odlišných částí.

Základní metody jako prahování nebo detekce hran jsou pro takto složitý problém nepoužitelné. Pro segmentaci přileb a helmovních korun by se hodila metoda rozvodí nebo shluková analýza. Ale pouze pro barevné obrazy, bohužel velká část převážně starší literatury obsahuje pouze černobílé kresby erbů, na které tyto metody nelze použít.

Štíty, helmovní koruny, točenice a honosné kusy i když mají daná pravidla, jak se kreslí, nelze je segmentovat pomocí syntaktického popisu, protože každý kreslíř si je může mírně upravit a tato metoda by zde selhávala. Vhodnou metodou by zde mohla být ASM (metodu AAM nelze použít, protože bere v potaz texturu objektu), ale bylo by nutné natrénovat spoustu různých modelů. Dále metoda ASM potřebuje velmi přesnou inicializaci, kde se hledaný objekt nachází. Což by bylo snadné u štítu, protože ten je lokalizován pomocí detektoru, ale přilba a její součásti mohou být na různých pozicích a každý kreslíř je kreslí různě veliké.

Jako nejvhodnější metodu pro segmentaci štítů, přileb, helmovní koruny, točenice, knížecí čepice a hraběcí koruny jsem vybral sémantickou segmentaci pomocí konvolučních neuronových sítí (kapitola 4.4). Vybrané součásti mají předem definovanou podobu a i s rozlišenostmi u různých kreslířů si dokáží konvoluční neuronové sítě poradit. Dále je metoda invariantní vzhledem k posunutí. Metoda také nalezené objekty zároveň klasifikuje do správné třídy.

Pro segmentaci klenotů a přikryvadel jsem vybral aktivní kontury (kapitola 3.2.4). Tyto dvě součásti mohou mít různé tvary, proto nejvhodnější způsob jejich segmentace je obejmout konturou celý erb, odstranit dříve nalezené součásti a část erbu nad helmovní korunou, točenicí nebo přilbou prohlásit za klenot a zbývající části za přikryvadla.

5.3 Detektor štítu

Prvním krokem bylo vytvořit datovou sadu. Pro trénovací množinu jsem použil knihy *České erby* od Jiřího Loudy a Josefa Janáčka [16], *Lexikon české šlechty* (díl 1-3) od Jana Haldy [10, 11, 12] a *Erbovník* od Milana Myslivečka [22, 23]. Celkem jsem pro trénovací množinu získal 599 štítů. Pro testovací množinu jsem využil stejné knihy, ale jen takové štíty, které nebyly využity pro trénovací množinu. Testovací množinu jsem dále rozšířil o obrázky z *Atlasy erbů a pečeti české a moravské středověké šlechty* od Augusta Sedláčka (svazky 2-4) [32, 33, 34]. Nakonec jsem do testovací množiny přidal vlastní fotografie erbů z historických měst Moravy a několika moravských hradů a zámků. V testovací množině se celkem nachází 271 obrázků se 300 štíty. Negativní množina je vytvořena z výřezů ostatních součástí erbu z knih. Dále byly přidány náhodné výřezy z fotografií z historických částí města Znojma, Olomouce a Brna. Celkem negativní množina obsahuje 2054 obrázků.

5.3.1 Optimalizace nastavení

Jak bylo zmíněno v kapitole 3.4.3, metoda HOG potřebuje detekční okno rozdělené na bloky, které se dále dělí na buňky. Dále potřebuje určit počet binů. Autoři článku [7] použili pro detekci chodců detekční okno 64 x 128 pixelů. Pro detekci štítů postačí detekční okno tvaru

čtverce, jehož velikost jsem zvolil 64 x 64 pixelů. Pro zvolení velikosti bloků, buněk a počtu binů bylo potřeba provést pokusy a porovnat výsledky.

K tomuto účelu jsem naprogramoval v Emgu CV¹ aplikaci, která natrénuje SVM podle dané datové sady. V HOG se používá SVM s lineárním jádrem. V knihovně OpenCV² je dále nutné vybrat typ SVM. Zvolil jsem typ $\mathcal{E}$ -Support Vector Regression, protože pro základní nastavení HOG dosahoval nejlepších výsledků. Tento typ přijímá dva parametry a to p , což je nejmenší vzdálenost příznakových vektorů trénovací množiny a rozdělující nadplochou, a parametr c , který se používá pro penalizaci odlehlých hodnot [3]. Jako výchozí hodnoty jsou $p = 0,1$ a $c = 0,01$.

Velikosti bloků jsem si určil jako dělitele 64 patřící do intervalu $\langle 8, 32 \rangle$ a velikost buněk patřící do intervalu $\langle 4, 16 \rangle$. Protože buňka musí být menší než velikost bloku. Pro počet binů jsem určil možné hodnoty dělitelů 180 z intervalu $\langle 6, 36 \rangle$. Hodnoty parametrů p a c jsem ručně testoval se základním nastavením a zkoumal, s jakými hodnotami se mění výsledky a po jakém kroku. U p jsem zvolil maximální hodnotu 0,45 a u c 0,8. Obě s krokem 0,5. Kombinací všech 5 parametrů bylo více jak 160 000, což nebylo v mých silách vyzkoušet. Rozhodl jsem se proto nejprve vyzkoušet kombinací velikosti bloků, buněk a počtu binů a až pro nejlepší kombinace těchto prvních 3 parametrů zkoušet měnit parametry p a c .

Z natrénovaných detektorů byl nakonec vybrán detektor s bloky o velikosti 32 pixelů, dělených do buněk o velikosti 8 pixelů a s počtem binů 6. Ověření funkčnosti natrénovaných detektorů a výběr nejlepšího detektoru je popsán v kapitole 5.3.

5.4 Segmentace a klasifikace štítu, přilby, helmovní koruny, točence a honosných kusů

Jak je popsáno v kapitole 5, pro segmentaci těchto vybraných součástí jsem zkoušel a studoval různé metody z [24, 27], ale žádná metoda nebyla pro řešený program vhodná. Proto jsem se rozhodl využít sémantickou segmentaci pomocí konvolučních neuronových sítí.

Podle článku [8] jsem reimplementoval FCN-8s síť, se kterou autoři dosáhli nejlepších výsledků. Knihovnu pro strojové učení jsem vybral *TensorFlow*³, na které nejprve pracovali výzkumníci od Googlu, ale v listopadu roku 2015 byla zveřejněna jako open-source a na jejích vývoji se v současné době podílí velká komunita vývojářů.

Prvním krokem bylo implementovat upsampling, jehož implementaci jsem převzal z [25]. Implementaci autor porovnával s knihovnou *skimage*.

Druhým krokem bylo implementovat architekturu FCN-8s sítě. Pro úpravu architektury šestnáctivrstvé VGG sítě popsanou v kapitole 4.4 byla využita knihovna *TF-Slim*⁴, což je knihovna pro definici, trénování a vyhodnocení modelů knihovny TensorFlow. Nabízí celou škálu modelů a mezi nimi je i model sítě VGG-16. Pomocí těchto dvou knihoven byla architektura FCN-8s sítě reimplementována. Při reimplementaci bylo postupováno podle článku a také podle Caffe modelu na oficiálním GitHub repozitáři autorů⁵, jehož přepsáním byla získána správná implementace FCN-8s sítě.

Pro natrénování sítě bylo potřeba shromáždit co největší množství obrázku a ručně je segmentovat. Podařilo se mi shromáždit trénovací množinu o 460 vzorcích. Pro tvary štítů jsem shromáždil dostatek vzorků štítu gotického, španělského, kolčího, francouzského

¹<http://www.emgu.com/>

²<http://opencv.org/>

³<https://www.tensorflow.org/>

⁴<https://github.com/tensorflow/models/blob/master/inception/inception/slim/README.md>

⁵<https://github.com/shelhamer/fcn.berkeleyvision.org>

a renezančního. Ze tvarů přileb jsou v datové množině zastoupeny hrncová, kolčí a turnajská přilba. Z ostatních součástí jsou zastoupeny točenice, helmovní koruny, knížecí čepice a také hrabčí koruny. Všechny obrázky pochází z [16, 10, 11, 12, 22, 23]. Trénovací množina obsahuje pro každý obrázek požadovanou výstupní segmentační mapu. Síť VGG-16 očekává na vstupu obrázky o velikosti 224 x 244 pixelů, proto všechny obrázky v trénovací množině byly převedeny do této velikosti.

Autoři článku uvádí, že nejlepších výsledků dosáhli fixním koeficientem učení 10^{-4} , podmnožinou trénovací množiny velikosti 1 (pro trénovací krok se použije jeden obrázek) a následně dvakrát koeficient učení podělili stem.

Pro získávání potřebné podmnožiny trénovací množiny při trénování jsem implementoval třídu *TrainDataSetReader*, která pomocí funkcí z knihovny TS-Slim upraví vstupní obrázky a anotaci obrázků (matice, která na každé pozici obsahuje index třídy, do které daný pixel patří, vytvořená ze segmentační mapy) upraví tak, že vytvoří pravdivostní tensor o velikosti anotace a hloubce počtu tříd. Ve kterém každá vrstva má pravdivostní hodnotu True na pozici, kde se v anotaci nachází pixel patřící do dané třídy. Dále pro každou epochu provede permutaci obrázků, aby se vždy vracely v jiném pořadí.

Dále jsem vytvořil skript *train.py* pro natrénování modelu. Skriptu lze pomocí argumentů nastavit vše důležité od počtu tříd, přes počet trénovacích kroků, koeficient učení a další. Umožňuje také po zadaném počtu kroků vytvářet záchytné body a z nich trénování později obnovit. Každý krok zapisuje do log souborů hodnoty všech významných proměnných a jejich hodnoty lze poté vynést do grafů pomocí nástroje *TensorBoard*.

Trénování bylo rozděleno na tři části, kde každá z nich měla počet trénovacích kroků zvolených tak, aby se vykonalo 50 epoch a poté následovala další část se sníženým koeficientem učení. Autoři článku uvádí, že nejlepších výsledků dosáhli alespoň po 175 epochách. Tento počet byl pro mé možnosti příliš vysoký a také mám menší datovou sadu s méně třídami, proto jsem pro první pokus vyzkoušel již zmíněných 50 epoch. Takto zvolený počet epoch se ukázal více než dostačující, jak lze vidět na obrázku 5.2.



Obrázek 5.2: Výstup cross-entropy funkce (osa x) během jednotlivých kroků učení (osa y) FCN-8s sítě.

5.4.1 Sjedení štítu a přilby

Ne vždy se podařilo štít i přilbu správně segmentovat, ale v části jednoho štítu nebo přilby byly občas pixely klasifikovány do jiných tříd, jak lze vidět na obrázku 6.5.

Zlepšení výsledků se mi podařilo dosáhnout tak, že jsem vytvořil binární obraz pro štít do kterého se zapsaly všechny pixely, které byly klasifikovány do jedné z tříd štítů a zaznamenával se počet výskytů jednotlivých tříd. Poté jsem v binárním obraze našel kontury pomocí metody aktivních kontur (kapitola 3.2.4), vybral tu největší a tu jsem vykreslil do segmentační mapy s hodnotou dominantní třídy. Stejný způsob byl zopakován i pro přilbu. Tímto způsobem došlo k sjednocení různě klasifikovaných částí a také díky výběru největší kontury došlo k odstranění malých shluků, které byly chybně klasifikovány. Většinou se tyto shluky nacházely v oblasti klenotů nebo přikryvadel.

5.5 Segmentace klenotu a přikryvadel

Segmentaci posledních součástí lze provádět pouze pokud se v předchozí části podařilo úspěšně segmentovat přilbu a štít. Bez znalosti polohy těchto dvou částí není možné pokračovat.

Jak již bylo zmíněno, mezi přilbu a klenot se může vkládat točenice nebo helmovní koruna, ale také klenot může ležet přímo na přilbě. Jako první krok je nutné prozkoumat prostor přímo nad přilbou, jestli se v něm nachází točenice nebo helmovní koruna a získat její pozici. Pokud bylo hledání úspěšné, je nutné si zapamatovat střed dolní hrany jako *dělicí bod*. Pokud bylo úspěšné nalezení točenice nebo helmovní koruny, je jako *dělicí bod* uložen střed vrchní hrany přilby.

Dále je nutné získat konturu celého erbu. Nejlepších výsledků se mi podařilo dosáhnout tak, že jsem pomocí Cannyho detektoru hran detekoval hrany v obraze. Následně je nutné pomocí dilatace se strukturním elementem velikosti 2×2 pixelů hrany spojit tak, aby mezi nimi nebyly mezery. Tento způsob však bohužel nenávratně spojil část přikryvadel s klenotem, pokud byly velmi blízko u sebe.

Následně jsem v obraze našel všechny kontury a hledal nejmenší z nich, ve které se nachází střed štítu, přilby a popřípadě střed točenice nebo helmovní koruny. Na získanou konturu jsem použil erozi o stejné velikosti strukturního elementu jako u dilatace, abych alespoň částečně odstranil její negativní následky.

Posledním krokem bylo konturu rozdělit na dvě části v *dělicím bodě*. Opět jsem našel konturu, seřadil jsem je podle velikosti a vybral dvě největší kontury. Následně jsem je seřadil podle velikosti souřadnice y a první konturu (ležící nejvýše) jsem prohlásil za konturu klenotu a druhou za konturu přikryvadel. V kontuře přikryvadel jsem na závěr odstranil části, které náležejí štítu a přilbě.

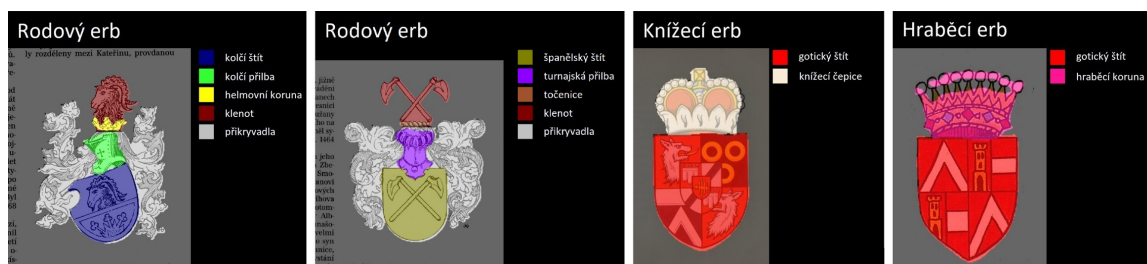
5.6 Propojení částí systému

Propojení jednotlivých částí systému bylo provedeno stejným způsobem, jaký je popsán v kapitole 5.1.

V prvním kroku je prohledán vstupní obraz ve více rozlišeních pomocí detektoru štítu (kapitola 5.3). Nejprve bylo nutné určit jak velký štít se bude v obraze hledat. Opět z pozorování různých knih obsahujících erby bylo vyvozeno, že čtvercová oblast ve které se nachází štít musí zabírat alespoň 20% velikosti vstupního obrazu. Následně je vstupní obraz zmenšen na takovou velikost, že velikost okna detektoru tvoří 20% delší hrany vstupního obrazu a ten je následně zmenšován. Tato část je implementována pomocí knihovny OpenCV, která pro HOG detektor obsahuje metodu *detectMultiScale* hledající objekty ve více rozlišení. Proto postačí vstupní obraz pouze dvakrát zmenšit na polovinu.

Další krok je vytvoření výřezu ve kterých se budou hledat součásti erbu. Již bylo zmíněno, že pro zajištění, aby se v malém výřezu nacházely všechny součásti se známým tvarem, postačí nalezenou oblast se štítem rozšířit na 2,1 násobek. V této části však došlo k problému, protože detektor občas označil oblast, ve které nebyl štít celý a v těchto případech většinou scházel spodek štítu, takže rozšíření směrem nahoru a do stran nebylo dostačující. Opět z pozorování výsledku bylo vyvozeno, že postačí nalezenou oblast nejprve posunout směrem dolů o $1/5$ velikosti štítu a až následně oblast rozšířit na 2,1 násobek. V tomto kroku byl vytvořen taký velký výřez (dvojnásobek malého) pro hledání součásti z neznámým tvarem. Erby se na stránkách knih často nachází na kraji a při vytváření výřezů by se mohlo zasáhnout mimo vstupní obraz. Z toho důvodu byl vstupní obraz před vytvářením výřezů rozšířen o černé okraje dvojnásobné velikosti než velikost štítu.

Následně jsou malé výřezy předány na zpracování FNC-8s síti, jejíž implementace je popsána v kapitole 5.4, díky níž získáme predikované segmentační mapy. Následně je nutné vyřadit nevalidní výřezy. Za tímto účelem se spočítá jakou část zabírají pixely, které byly klasifikovány do jedné ze tříd štítů. Pro určení hranice, kdy je výřez považovat za validní, bylo nutné provádět experimenty. Tato hranice byla nakonec nastavena na pouhých 10% (kolik v segmentační mapě musí zabírat štít), i takhle nízká hodnota však stačila na vyřazení výřezu, kde se erb nenachází a zároveň zachovala takové výřezy, kde byl štít nalezen v menší velikosti. Při hledání štítu ve více rozlišeních byl zpravidla jeden štít nalezen několikrát. K odstranění těchto duplicit byly nalezeny překrývající se výřezy a ponechán pouze ten s největším podílem štítu.



Obrázek 5.3: Ukázka výstupu systému na erbech, které nebyly v trénovací množině. Na vstupu systému byly použity erby z knih: zleva 2x [12] a [16, 23].

Následující krok záleží na tom, zda byla ve výřezu nalezena přilba (zjištěno již při sjednocení štítu a přilby popsané v kapitole 5.4.1) a jakou část segmentační mapy zabírá honosný kus. Pokud přilba byla nalezena, jedná se o klasický rodový erb, a proto jsou ve velkém výřezu nalezeny přikryvadla a klenot (kapitola 5.5) a odstraněny pixely klasifikovány jako honosný kus. Pokud přilba nalezena nebyla nebo honosný kus zabírá větší část segmentační mapy než přilba, tak je klasifikován titul vlastníka erbu (kníže, hrabě) a odebrány pixely klasifikovány jako přilba nebo její výstroj.

Po získání konečné segmentační mapy je vykreslena do vstupního obrazu a pokud byl nalezen klasický rodový erb, tak je vyřazeno okolí erbu. Okraj výřezu byl zvolen $1/5$ delší strany nejmenšího obdélníku, do kterého se vejde kontura celého erbu (která je nalezena při hledání přikryvadla a klenotu). V závěrečném kroku je do výstupního obrazu přidán popis o jaký erb se jedná a vygenerována legenda součástí nalezených v erbu. Výstup systému je zobrazen na obrázku 5.3.

Kapitola 6

Ověření funkčnosti a experimenty

V této kapitole jsou nejprve představeny vybrané metriky pro měření úspěšnosti detekce objektů a segmentace. Následně je popsáno ověření funkčnosti jednotlivých částí systému popsaných v kapitole 5 a na závěr jsou uvedeny experimenty s výsledným systémem.

6.1 Výběr metrik

Pro hodnocení detektorů štítu jsem zvolil metriku *Intersection over Union* (dále jako IoU). Pro použití této metriky je nutné na testovacích obrázcích vyznačit oblast, ve které se nachází hledaný štít, a tato oblast se porovná s vyznačenou oblastí od detektoru. IoU se vypočítá podle vzorce [29]:

$$\text{IoU} = \frac{\text{plocha překrytí}}{\text{plocha sjednocení}} \quad (6.1)$$

Výsledek metriky je hodnota z intervalu $<0,1>$. Jako nejmenší přípustná hodnota se považuje 0,5.

Pro měření úspěšnosti segmentace jsem využil dvě metriky. Metrika pixel accuracy [8] udává, jak dobře byl celý obrázek segmentován:

$$\text{pixel accuracy} = \frac{\sum_i n_{ii}}{\sum_i t_i} \quad (6.2)$$

kde n_{ij} je počet pixelů třídy i predikované do třídy j a $t_i = \sum_j n_{ij}$ je celkový počet pixelů třídy i .

Jako druhou metriku jsem použil jinou verzi Intersection over Union, kterou budu ve shodě s autory článku [8] označovat jako IU. Tato metrika udává jak dobře se podařilo segmentovat jednotlivé objekty a dle [28] ji vypočítáme jako:

$$\text{IU} = \frac{\text{TP}}{\text{FP} + \text{TP} + \text{FN}} \quad (6.3)$$

kde TP značí true positive, FP false positive a FN false negative. Tato metrika se počítá zvlášť pro každou třídu a pixely patřící do jiné třídy se chápou jako součást pozadí.

Obě metriky vrací výsledky v intervalu $<0, 1>$. Já však výsledky ve shodě s autory článku [8] přepočítávám do intervalu $<0, 100>$ a zaokrouhluji na jedno desetinné místo.

6.2 Výsledky testů pro ověření funkčnosti

V této podkapitole jsou popsány testy pro ověření funkčnosti jednotlivých součástí systému.

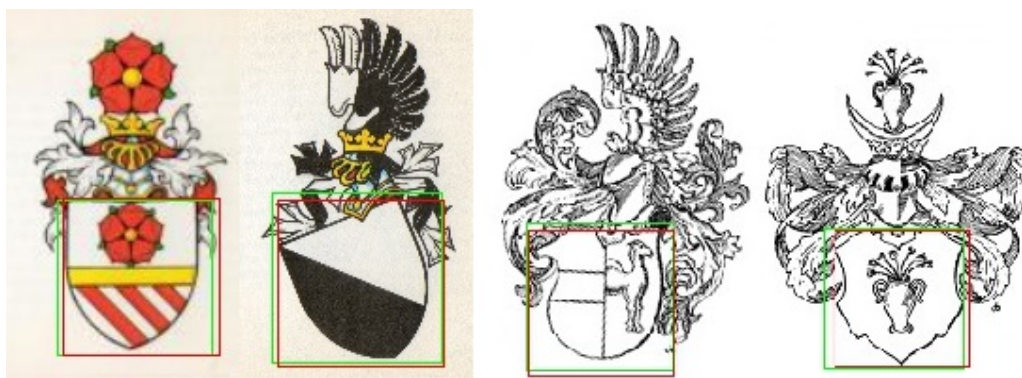
6.2.1 HOG detektor štítů

Pro vyhodnocení nejlepšího natrénovaného HOG detektoru jsem nejprve provedl testy pro 29 kombinací nastavení popsaných v kapitole 5.3. V tabulce 6.1 uvádím výsledky pro nejlepších 10 detektorů (kompletní výsledky jsou uvedeny v příloze A).

Tabulka 6.1: Výsledky nejlepších 10 detektorů pro různé velikosti (bloků, buněk, počtu binů), true positives, false positives, úspěšnost nalezení štítů, průměrná hodnota IoU a délka výpočtu pro celou testovací množinu.)

#	Detektor	TP	FP	Úspěšnost [%]	Prům. IoU	Výpočet [s]
1	(32, 4, 6)	271	1543	90,33	0,757	181,54
2	(16, 4, 6)	263	1372	87,67	0,749	96,67
3	(32, 8, 6)	261	577	87,00	0,768	24,44
4	(32, 8, 9)	260	844	86,67	0,762	24,95
5	(8, 4, 20)	260	1959	86,67	0,734	99,42
6	(32, 8, 20)	257	960	85,67	0,755	28,23
7	(8, 4, 6)	257	1848	85,67	0,751	70,50
8	(16, 4, 9)	256	1614	85,33	0,746	116,03
9	(32, 8, 30)	255	638	85,00	0,751	30,95
10	(32, 8, 15)	255	765	85,00	0,752	26,88

Jako nejlepší jsem vybral detektor, který se umístil na 3. místě. Má sice menší úspěšnost než první dva detektory, ale menší false positives, nejvyšší průměrnou hodnotu IoU u nalezených štítů a také je mnohem rychlejší. Detektor byl úspěšný jak na kresbách od kreslířů, kterých štítů byly použity v trénovací množině (obrázek 6.1) tak i u kreslířů, kteří v ní zastoupeni nebyli. Byl úspěšný i na větší části reálných štítů (obrázek 6.2). Detektor nenalezl několik štítů od jiných autorů, ale většina z neúspěšně nalezených štítů je z fotografiích, kde není štít tolik patrný (obrázek 6.3).



Obrázek 6.1: Detekované štítů na kresbách, které byly převzaty po dvojicích z [16, 11]. Zelenou barvou je manuálně označená požadovaná oblast a červenou je oblast, kterou našel detektor.



Obrázek 6.2: Detekované štíty na fotografiích. První fotografie zleva převzata z [32].



Obrázek 6.3: Štíty, které se nepodařilo detekovat.

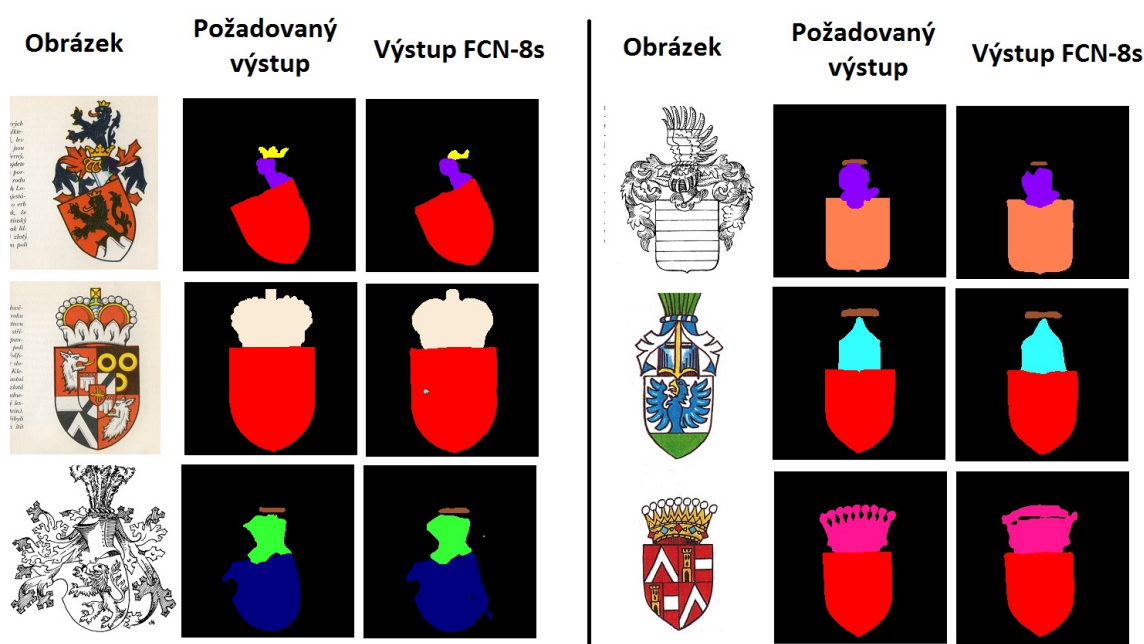
Při provádění testů pro různá nastavení SVM se nepodařilo dosáhnout lepších výsledků.

6.2.2 Segmentace součástí se známým tvarem pomocí FCN-8s sítě

Pro ověření funkčnosti jsem vytvořil testovací sadu o 35 obrázcích (a jejich segmentačních mapách), ve kterých jsou zastoupeny všechny tvary štítů, přileb, helmovní koruny, točenice a honosné kusy od všech kreslířů, jejich kresby byly použity v trénovací množině. Výsledky lze vidět na obrázku 6.4.

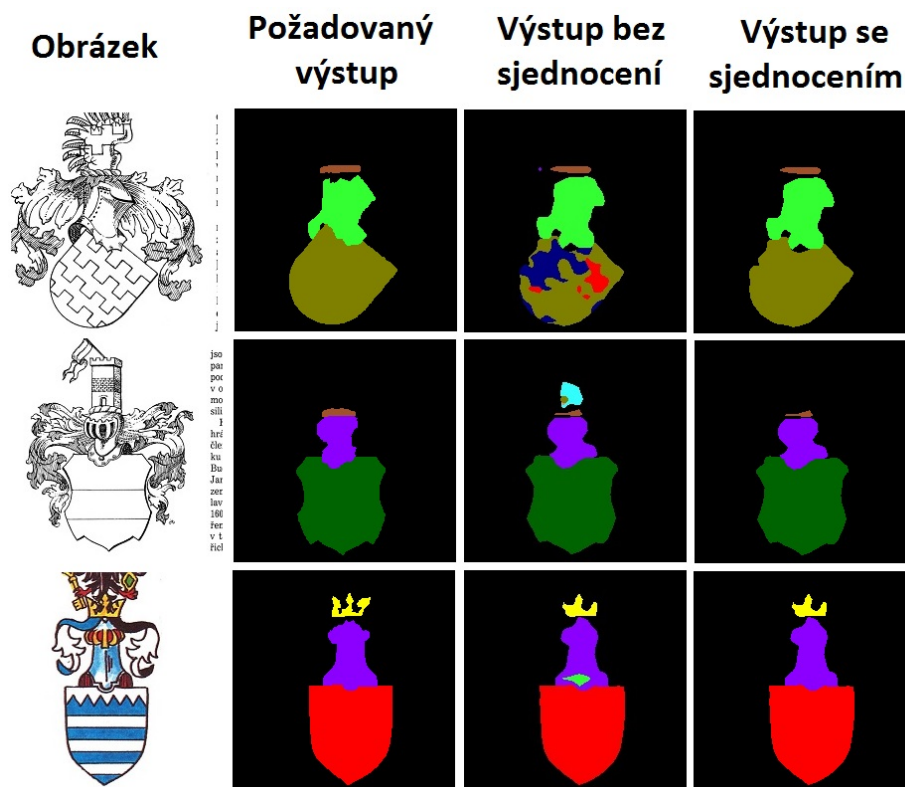
Tabulka 6.2: Výsledky jednotlivých metrik před a po použití sjednocení štítu a přileb.

Metrika	Před sjednocením	Po sjednocení
pixel accuracy	98,3	98,6
gotický štít (IU)	96,8	97,4
kolčí štít (IU)	87,0	94,2
španělský štít (IU)	84,4	94,6
francouzský štít (IU)	95,2	95,2
renezanční štít (IU)	95,3	95,3
hrncová přilba (IU)	87,7	89,9
kolčí přilba (IU)	86,0	86,9
turnajská přilba (IU)	85,0	85,7
točenice (IU)	63,3	63,3
helmovní koruna (IU)	51,7	51,7
knížecí čepice (IU)	95,7	95,7
hraběcí koruna (IU)	85,6	85,6



Obrázek 6.4: Výsledky segmentace pomocí FCN-8s sítě. Erby byly po dvojicích shora převzaty z [16, 11, 22].

V tabulce 6.2 jsou zobrazeny výsledky pro segmentaci pomocí FCN-8s sítě před i po sjednocení popsaném v kapitole 5.4.1. Kromě francouzského a renezančního štítu došlo po sjednocení (obrázek 6.5) ke zlepšení u všech ostatních tvarů štítů a všech tvarů přileb. Nejhorší dopadla segmentace helmovních korun. Zde však může být příčinou nedokonalost jak trénovací tak testovací množiny, protože jak helmovní koruna, tak točenice se velmi špatně ručně segmentovaly.



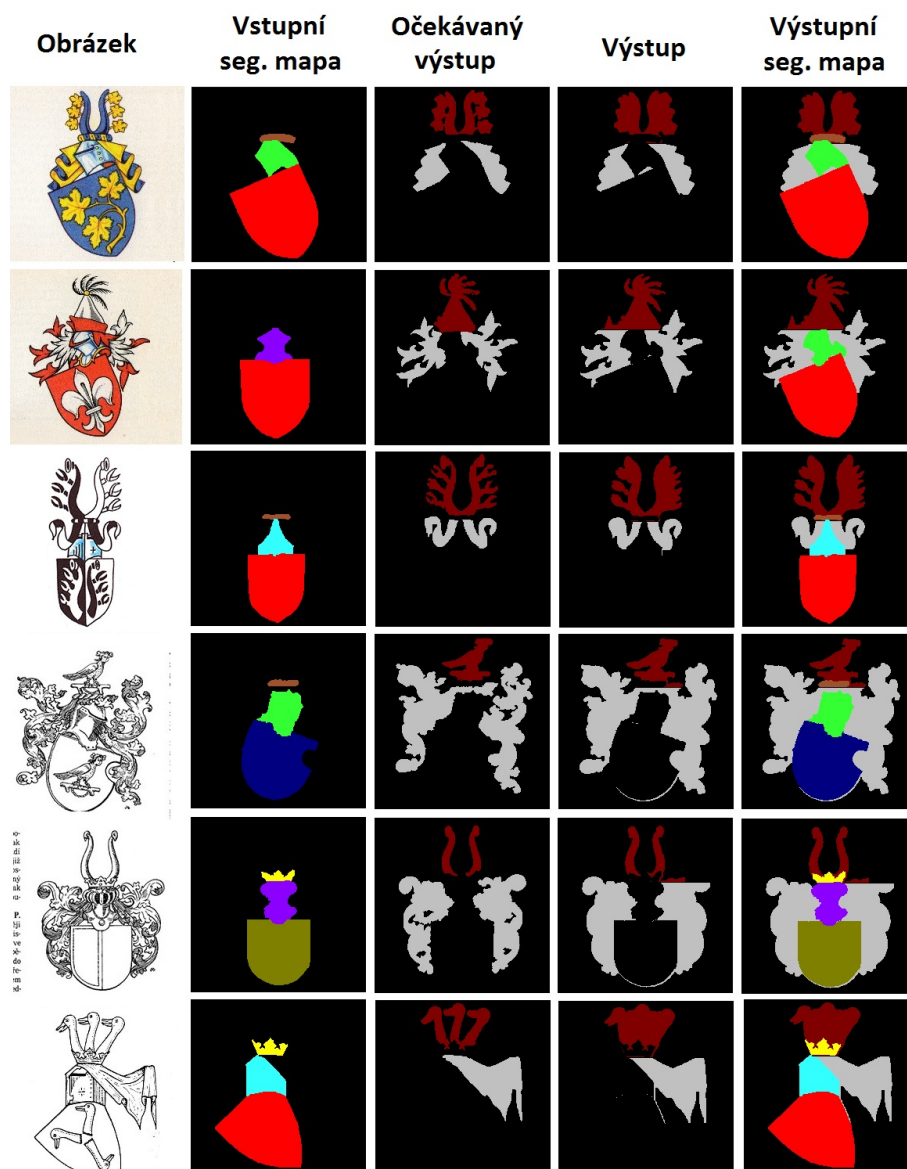
Obrázek 6.5: Výsledky segmentace po sjednocení štítu a přileb. Erby byly shora převzaty z [12, 11, 22].

6.2.3 Segmentace klenotů a přikryvadel pomocí aktivních kontur

Pro ověření funkčnosti této části systému jsem shromáždil 18 erbů s různými tvary přikryvadel. Pro každý obrázek byla ručně vytvořena požadovaná výstupní segmentační mapa pro segmentaci klenotu a přikryvadel. Tato segmentační mapa byla porovnána s výstupem metody podle stejných metrik jako v předchozí části systému. Výsledky jsou uvedeny v tabulce 6.3.

Tabulka 6.3: Výsledky jednotlivých metrik pro segmentaci klenotů a přikryvadel.

Metrika	Výsledky
pixel accuracy	96,3
klenot (IU)	77,2
přikryvadla (IU)	77,9

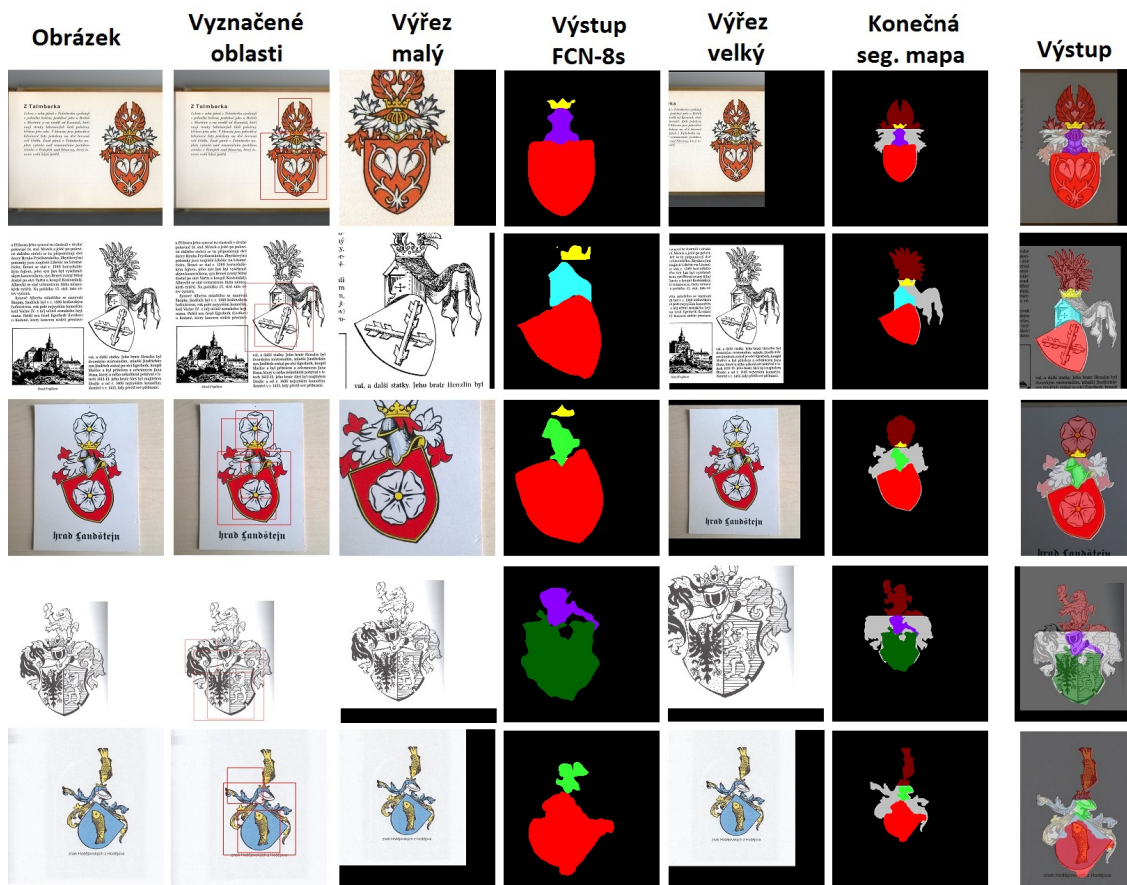


Obrázek 6.6: Výsledky segmentace klenotů a přikryvadla, zobrazení výsledné segmentační mapy. Erby byly shora převzaty z: první dva [16], třetí [22] a poslední dva [11].

Výsledky segmentace klenotů i přikryvadla můžeme vyhodnotit jako dobré. Ke snížení výsledků dochází kvůli použití dilatace, protože ta v některých případech způsobuje spojení klenotu s přikryvadly. Díky tomu se při dělení může část přikryvadla dostat do kontury klenotu a tím dochází ke ztrátě detailů u obou součástí. Kvůli použití kontur není možné segmentovat části vně klenotů a přikryvadla, které by měly náležet do pozadí. Poslední faktor, který snižuje výsledky je, že vstupní segmentační mapa nemusí mít přesně segmentovaný štít a po jeho odstranění zůstávají hrany klasifikované jako část přikryvadla, jak lze vidět na 4. a 5. erbu na obrázku 6.6.

6.3 Experimenty

Výsledný systém jsem vyzkoušel na několika různých obrázcích a výsledky byly velmi různorodé.



Obrázek 6.7: Zobrazeny jsou postupně vstupní obrázky a označené oblasti, ve kterých by se měly nacházet štíty. Následně je zobrazený malý výřez, výstup FCN-8s sítě z malého výřezu, velký výřez, výsledná segmentační mapa získaná z velkého výřezu a výstup systému (bez popisku a legendy). První dva erby shora pocházejí od kreslířů, jejichž erby jsou v trénovací množině a jsou převzat z [30, 12]. Ostatní erby pocházejí od kreslířů, kterých erby nebyly v trénovací množině. Třetí erb pochází z pohlednice z hradu Landštejn, čtvrtý z [31] a poslední z [32].

Jak můžeme vidět na obrázku 6.7 pro výřezy z knih, které byly použity pro trénovací množinu dopadly experimenty velmi dobře (první dva erby a další je možné vidět na obrázku 5.3) a stejně tak pro třetí erb, jehož autor kreslí velmi podobně jako kreslíři, jejichž erby se v trénovací množině nacházejí. Na čtvrtém erbu vidíme, že byl nepřesně segmentován štít a nebyla nalezena helmovní koruna. Na posledním erbu také není nalezena helmovní koruna a dále je chybně klasifikován tvar štítu.

Další experimenty byly prováděny nad kresbami erbů od kreslířů, jejichž erby nebyly zastoupeny v trénovací množině a jsou zobrazeny na obrázku 6.8.

Na prvním erbu shora lze vidět, že tvar štítu je správně segmentovaný, ale přetéká přes oblast, ve které by se měl nacházet. V trénovací sadě mají všechny erby okolo sebe jen bílé



Obrázek 6.8: Porovnání výstupů pro původní a ručně upravené obrázky. První erby shora převzat z [36] a ostatní z [1].

pozadí a pouze výjimečně se na okraj dostalo pár písmen z textu. Zkusil jsem odstranit spodní část rámu, ve kterém se erb nachází a následně došlo k velkému zlepšení. Štít již nepřetéká přes své okraje.

Ve druhém obrázku ne jen, že opět došlo k přetečení hranic štítu, ale navíc byl chybně klasifikován jeho tvar (klasifikován jako španělský štít). Opět jsem odstranil hrany okolo štítu. Přetečení hranic štítu nebylo sice úplně odstraněno, ale došlo ke zlepšení. Především však byl správně klasifikován tvar štítu (francouzský štít).

U posledního erbu došlo ke stejným chybám jako v předchozím. Štít přetéká přes své hranice a také byl chybně klasifikován jeho tvar (jako kolčí štít). Stejně jako v předchozích případech jsem odstranil hrany okolo štítu. Opět došlo ke zmírnění přetečení hranic štítu, ale již nedošlo ke správné klasifikaci. Všechny erby v trénovací množině mají každou součást ohraničenou tmavší černou barvou než erby na posledních dvou obrázcích. Zkusil jsem tedy dále obtáhnout hranu štítu černou barvou. Po tomto kroku došlo k velkému zlepšení. Ne jen, že bylo ještě více zmírněno přetečení hranic štítu, došlo ke správné klasifikaci tvaru štítu (jako gotický štít) a také byla oblast štítu více vyplněna.

Problémy, které vznikají u těchto kreseb by bylo možné vyřešit zásadním rozšířením trénovací množiny.

Výsledný systém jsem také vyzkoušel na sadě fotografií obsahující reálné erby. Na fotografiích jsou vyfocené erby na zdech kaplí, kostelů, zámků a vyšité erby na polstrování židlí. Zde však segmentace jednotlivých součástí úplně selhala. Na kresbách v trénovací množině jsou jednotlivé součásti erbu ohraničeny výraznou černou hranou (má vysokou velikost gradientu). Jedním z faktorů, proč k selhání dochází může být, že se konvoluční neuronová síť naučila tyto hrany vyhledávat (jak lze mimo jiné vidět na předchozím experimentu). U erbů vyrytých ve zdi není hrana tolik viditelná a v některých místech součásti erbu splývají mezi sebou nebo s pozadím. Další faktor, proč segmentace selhává je, že FCN-8s síť naučená na mé trénovací množině nejlépe segmentuje erby, které nemají kolem sebe žádné další hrany (jak lze opět vidět na předchozím experimentu) a ani tuto podmínku reálné erby většinou nespĺňují.

Kapitola 7

Závěr

V této práci byl popsán návrh a implementace systému, který dokáže ve vstupním obraze detekovat erb, provést jeho dekompozici na jednotlivé součásti a ty poté klasifikovat.

Erb je v obraze vyhledán pomocí metody Histogramu orientovaných gradientů. Pro natrénování detektoru byla vytvořena trénovací množina o 599 obrázcích, negativní množina o 2054 obrázcích a testovací množina o 271 obrázcích se 300 štíty. Bylo natrénováno několik detektorů a z nich následně vybrán jako nejlepší detektor s úspěšností 87%.

Pro segmentaci součástí erbu se známým tvarem (štíty, přilby, helmovní koruna, točnice, knížecí čepice a hraběcí koruna) byla natrénována FCN-8s konvoluční neuronová síť pro sémantickou segmentaci. Trénovací množina obsahuje 460 erbů a jejich segmentační mapy. Funkčnost sítě byla ověřena na testovací množině s 35 erby, ve které jsou zastoupeny všechny vybrané součásti od různých kreslířů. Metrikou pixel accuracy byla změřena přesnost 98,6 ze 100 a všechny jednotlivé součásti dosáhly v měření metrikou Intersection over Union větší než požadovanou hodnotu 50 ze 100. Nejlepších výsledků dosáhl gotický štít s hodnotou 98,6, naopak nejhorších výsledků dosáhla helmovní koruna s hodnotou 51,7.

Segmentace součástí s neznámým tvarem (přikryvadla a klenot) je prováděna pomocí aktivních kontur tak, že je nalezena kontura celého erbu, ze kterého se odstranily části nalezené pomocí FCN-8s sítě a zbytek kontury je rozdělen na konturu přikryvadla a klenotu. Ověření funkčnosti probíhalo na 18 erbech s různými tvary přikryvadla. Metrikou pixel accuracy byla naměřena hodnota 96,3. Klenot dosáhl v měření metrikou Intersection over Union hodnoty 72,2 a přikryvadla 77,9.

S výsledným systémem byly provedeny experimenty. Systém funguje velmi dobře pro výřezy z knih, které jsou zastoupeny v trénovací množině nebo pro kresby jim podobné. U ostatních kreseb dosahuje výsledků horších. Pomocí experimentů byly zjištěny některé faktory, které zhoršují segmentaci pomocí FCN-8s sítě. Předpokládám, že tyto problémy by bylo možné vyřešit zásadním rozšířením trénovací množiny. Na reálných erbech systém zcela selhává. Erby se ve většině případů daří správně detekovat, ovšem velmi se liší od kreseb, na kterých je naučena FCN-8s síť.

Systém je možné rozšířit mnoha způsoby. Lze hledat a klasifikovat další součásti erbu, rozpoznat jeho majitele nebo analyzovat figury nakreslené na štítu. Heroldské figury mají přesně daná pravidla a tudíž jsou dobře analyzovatelná. U obecných figur by bylo možné vyhledávat alespoň nejpoužívanější znamení. Pokud by se podařilo analyzovat celý erb, lze provést blasonování (formální popis erbu dle heraldických pravidel [30]) celého erbu.

Literatura

- [1] *Jihomoravské muzeum ve Znojmě*, inv. č. L767, Rodokmen rodu Abštorských od roku 1485 do roku 1904, v Řeznu 1904.
- [2] *Jihomoravské muzeum ve Znojmě*, inv. č. Lf 46, Povýšení do šlechtického stavu Audueas Gepleis z Radkersburku (Štýrsko) pečeti a vydává Wolfgang A. z Kaltenhausenu a Greifewteinu 15. 9. 1624 pergamen, 15. 9. 1624 ve Štýrsku.
- [3] *Introduction to Support Vector Machines*. 2014, [Online; navštíveno 09.02.2017].
URL http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html
- [4] *MNIST For ML Beginners*. 2017, [Online; navštíveno 15.04.2017].
URL https://www.tensorflow.org/get_started/mnist/beginners
- [5] Čadík, M.: *Analýza a extrakce příznaků z textur*: přednáška do předmětu Počítačové vidění. FIT VUT v Brně, 2016, Brno.
- [6] Canny, J.: A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, ročník 8, č. 6, Červen 1986: s. 679–698, ISSN 0162-8828, doi:10.1109/TPAMI.1986.4767851.
URL <http://dx.doi.org/10.1109/TPAMI.1986.4767851>
- [7] Dalal, N.; Triggs, B.: Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, ročník 1, June 2005, ISSN 1063-6919, s. 886–893 vol. 1, doi:10.1109/CVPR.2005.177.
- [8] Evan Shelhamer, J. L. a. T. D.: *Fully Convolutional Networks for Semantic Segmentation*. *CoRR*, ročník abs/1605.06211, 2016.
URL <http://arxiv.org/abs/1605.06211>
- [9] Fukushima, K.: Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biological Cybernetics*, ročník 36, 1980: s. 193–202.
- [10] Halada, J.: *Lexikon české šlechty: erby, fakta, osobnosti, sídla a zajímavosti*. Praha: Akropolis, 1992, ISBN 80-901020-3-4.
- [11] Halada, J.: *Lexikon české šlechty II: erby, fakta, osobnosti, sídla a zajímavosti*. Praha: Akropolis, 1993, ISBN 80-85770-04-0, ilustroval František Doubek.
- [12] Halada, J.: *Lexikon české šlechty III: erby, fakta, osobnosti, sídla a zajímavosti*. Praha: Akropolis, 1994, ISBN 80-85770-12-1, ilustroval František Doubek.

- [13] Horák, K.; Kalová, I.; Petyovský, P.; aj.: *Počítačové vidění*. FEKT VUT v Brně, 2008, Brno.
- [14] Hradiš, M.: *Konvoluční neuronové sítě*. OpenAlt konference, 2015, [Online; navštíveno 15.04.2017].
URL <https://openalt.cz/2015/data/Michal%20Hradis%20-%20Konvolucni%20neuronove%20site.pdf>
- [15] Hradiš, M.: *Houghova transformace*: přednáška do předmětu Počítačové vidění, FIT VUT. FIT VUT v Brně, 2016, Brno.
- [16] Janáček, J.; Louda, J.: *České erby*. Praha: Albatros, oko vydání, 1974.
- [17] Janáček, J.; Louda, J.: *České erby*. Praha: Albatros, oko vydání, 1988.
- [18] Karpathy, A.: Convolutional Neural Networks for Visual Recognition. [Online; navštíveno 14.4.2017].
URL <http://cs231n.github.io/>
- [19] Krizhevsky, A.; Sutskever, I.; Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*, editace F. Pereira; C. J. C. Burges; L. Bottou; K. Q. Weinberger, Curran Associates, Inc., 2012, s. 1097–1105.
URL <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>
- [20] Mráček, Š.; Dražanský, M.; Váňa, J.: *Rozpoznávání podle obličeje*: přednáška do předmětu Biometrické systémy. FIT VUT v Brně, 2015, Brno.
- [21] Mrázek, Z.: Aplikace pro demonstraci metody Histogram of Oriented Gradients pro detekci objektů. 2014.
URL <http://hdl.handle.net/11012/34220>
- [22] Mysliveček, M.: *Erbovník, aneb, Kniha o znacích i osudech rodů žijících v Čechách a na Moravě podle starých pramenů a dávných ne vždy věrných svědectví*. Praha: Horizont, 1993, ISBN 80-7012-070-3.
- [23] Mysliveček, M.: *Erbovník 2, aneb, Kniha o znacích i osudech rodů žijících v Čechách a na Moravě: podle starých pramenů a dávných ne vždy věrných svědectví*. V Praze: Horizont, 1997, ISBN 80-7012-091-6.
- [24] Šonka, M.; Hlaváč, V.: *Počítačové vidění*. Praha: Grada, 1992, ISBN 80-85424-67-3.
- [25] Pakhomov, D.: *Image Segmentation with Tensorflow using CNNs and Conditional Random Fields*. [Online].
URL <http://warmspringwinds.github.io/tensorflow/tf-slim/2016/12/18/image-segmentation-with-tensorflow-using-cnns-and-conditional-random-fields/>
- [26] Španěl, M.: *Segmentace obrazu*: přednáška do předmětu Počítačové vidění, FIT VUT. FIT VUT v Brně, 2015, Brno.

- [27] Španěl, M.; Beran, V.: *Obrazové segmentační techniky*. FIT VUT v Brně, 2005, [Online; navštíveno 28.12.2016].
URL <http://www.fit.vutbr.cz/~spanel/segmentace/>
- [28] Rahman, Y., Md Atiqur a Wang: *Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation*. In *International Symposium on Visual Computing*, Springer, 2016, s. 234–244.
- [29] Rosebrock, A.: *Intersection over Union (IoU) for object detection*. [Online; navštíveno 30.12.2016].
URL <http://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>
- [30] Schwarzenberg, K. F.: *Heraldika: heraldika, čili, Přehled její theorie se zřetelem k Čechám na vývojovém základě*. Ve Vyšehradu 2. Praha: Vyšehrad: Historica (Vyšehrad), třetí vydání, 2007, ISBN 978-80-7021-827-3.
- [31] Vavřínek, K.: *Almanach českých šlechtických a rytířských rodů 2027*. Vavřínek Zdeněk, 2016, ISBN 978-80-905324-9-6.
- [32] a Vladimír Růžek, A. S.: *Atlasy erbů a pečeti české a moravské středověké šlechty*, ročník svazek 2. Praha: Academia, 2001, ISBN 80-200-0935-3.
- [33] a Vladimír Růžek, A. S.: *Atlasy erbů a pečeti české a moravské středověké šlechty*, ročník svazek 3. Praha: Academia, 2002, ISBN 80-200-1043-2.
- [34] a Vladimír Růžek, A. S.: *Atlasy erbů a pečeti české a moravské středověké šlechty*, ročník svazek 4. Praha: Academia, 2003, ISBN 80-200-1094-7.
- [35] Zbořil, F. V.; Zbořil, F.: *Základy umělé inteligence: studijní opora do předmětu Umělá inteligence*. FIT VUT v Brně, 2012, Brno.
- [36] a Zdirad J. K, Č.: *Heraldika : bulletin pro pomocné vědy historické*. Jílové u Prahy : Heraldická sekce při Muzeu těžby a zpracování zlata, 1975.
- [37] Zenger, Z. M.: *Česká heraldika*. Praha: Vyšehrad, 1978.

Přílohy

Příloha A

Výsledky testů pro detektory štítu

Tabulka A.1: Výsledky detektorů štítu pro různé velikosti (bloků, buněk, počtu binů), true positives, false positives, úspěšnost nalezení štítů, průměrná hodnota IoU a délka výpočtu.)

#	Detektor	TP	FP	Úspěšnost [%]	Prům. IoU	Výpočet [s]
1	(32, 4, 6)	271	1543	90,33	0,757	181,54
2	(16, 4, 6)	263	1372	87,67	0,749	96,67
3	(32, 8, 6)	261	577	87,00	0,768	24,44
4	(32, 8, 9)	260	844	86,67	0,762	24,95
5	(8, 4, 20)	260	1959	86,67	0,734	99,42
6	(32, 8, 20)	257	960	85,67	0,755	28,23
7	(8, 4, 6)	257	1848	85,67	0,751	70,50
8	(16, 4, 9)	256	1614	85,33	0,746	116,03
9	(32, 8, 30)	255	638	85,00	0,751	30,95
10	(32, 8, 15)	255	765	85,00	0,752	26,88
11	(8, 4, 36)	253	1665	84,33	0,740	133,41
12	(32, 8, 36)	251	663	83,67	0,755	32,60
13	(16, 4, 20)	249	1348	83,00	0,748	195,54
14	(16, 4, 15)	249	1346	83,00	0,752	158,53
15	(8, 4, 9)	247	2915	82,33	0,734	77,50
16	(8, 4, 15)	243	2128	81,00	0,728	89,36
17	(16, 8, 36)	242	375	80,67	0,765	14,39
18	(16, 8, 30)	242	378	80,67	0,763	13,70
19	(8, 4, 30)	239	1598	79,67	0,739	119,89
20	(16, 8, 15)	237	384	79,00	0,757	12,25
21	(16, 8, 20)	236	370	78,67	0,761	12,74
22	(16, 8, 9)	232	402	77,33	0,760	11,55
23	(32, 16, 30)	227	287	75,67	0,747	7,73
24	(16, 8, 6)	226	332	75,33	0,765	11,32
25	(32, 16, 36)	224	309	74,67	0,750	7,71
26	(32, 16, 15)	222	418	74,00	0,751	7,54
27	(32, 16, 20)	217	530	72,33	0,748	7,59
28	(32, 16, 6)	187	80	62,33	0,729	7,44
29	(32, 16, 9)	17	0	5,67	0,708	7,22

Příloha B

Obsah přiloženého DVD

- **coa_recognizer** - Ve složce se nachází zdrojové soubory výsledného systému.
- **fcn_8s** - Složka obsahuje skript pro trénovací FCN-8s sítě, trénovací množinu obrázků, skript pro predikci a naučený model FCN-8s sítě.
- **hog** - Složka obsahující trénovací negativní množinu obrázků pro HOG detektor. Dále obsahuje zdrojové soubory programu pro trénování HOG detektoru a skript pro natrénování sady detektorů podle zadaných parametrů.
- **poster** - Složka obsahuje plakát shrnující výsledky této práce ve formátu PDF a .pptx.
- **technical_report** - Složka obsahuje technickou zprávu ve formátu PDF a její zdrojové soubory.
- **tests** - Ve složce se nachází skripty testů pro ověření funkčnosti, jejich výsledky a testovací množiny obrázků.