Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta

Modul do serverové aplikace pro rozpoznávání identifikačních údajů z osobních dokladů

Diplomová práce

Bc. Miroslav Bartyzal

Školitel: Ing. Miroslav Skrbek, Ph.D.
Konzultant: Ing. Zbyněk Novák (GoPay s.r.o.)

České Budějovice 2018

BARTYZAL, Miroslav. *Modul do serverové aplikace pro rozpoznávání identifikačních údajů z osobních dokladů. [Server module for personal information recognition from identity documents. Mgr. Thesis, in Czech.]*. České Budějovice, Czech Republic, 2018. Diplomová práce. Faculty of Science, University of South Bohemia.

Anotace:

This Master's thesis deals with the creation of a server-side system used for the automated reading of personal information from photographed identity documents. It is focused on the processing of photographs made by camera phones with respect to various quality of their images. Text localization in images and its recognition by means of neural network are the subject of this thesis. The final system is tested by the client application which was created for the Android operating system.

Prohlašuji, že svoji diplomovou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své diplomové práce, a to v nezkrácené podobě, elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích

dne: …… ……

Podpis autora práce

Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta

ZADÁVACÍ PROTOKOL MAGISTERSKÉ PRÁCE

Student: Bc. Miroslav Bartyzal .....
(jméno, příjmení, tituly)

Obor – zaměření studia: Aplikovaná informatika 1802T001 .....

Katedra: Ústav aplikované informatiky

Školitel: .Ing. Miroslav Skrbek, Ph.D. .....
(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

Garant z PřF: .....
(jméno, příjmení, tituly, katedra – jen v případě externího školitele)

Školitel – specialista, konzultant: ing. Zbyněk Novák , GoPay s.r.o. .....
(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

Téma magisterské práce: Modul do serverové aplikace pro rozpoznávání identifikačních údajů z osobních dokladů

Cíle práce :

Navrhněte a implementujte modul do serverové aplikace pro rozpoznávání údajů z osobních dokladů pro účely registrace do platebních systémů. Vstupem do aplikace je fotografie dokladu. Výstupem bude záznam údajů do databáze a zpětná vazba pro uživatele o kvalitě fotografie a rozpoznaných údajích. Zaměřte se na zpracování obrazové informace s ohledem na různou kvalitu obrazu a interakci s uživatelem. V případě potřeby upravte a doplňte existující klientskou aplikaci o pořízení fotografie, detekci špatné kvality fotografie a zpětnou vazbu pro uživatele. Bezpečnostní aspekty projektu jsou mimo rámec této práce. Vytvořenou aplikaci otestujte a vyhodnoťte spolehlivost rozpoznání údajů. Konkrétní rozsah práce upravte po dohodě s vedoucím práce.

Základní doporučená literatura : dodá vedoucí práce a konzultant

Financování práce :.....

Vedoucí práce : Ing. Miroslav Skrbek, Ph.D. .....podpis : .....

U externích vedoucích fakultní garant práce.....podpis :.....

Garant oboru mag. studia ..... podpis : .....

Vedoucí katedry ..... podpis .....

Případný souhlas vedoucího ústavu AV .....podpis : .....

V Českých Budějovicích dne 5. 11. 2014

Převzal/a dne 5.11.2014 ..... podpis : .....

PODĚKOVÁNÍ

Rád bych touto cestou poděkoval všem členům mé rodiny, mým přátelům, známým a kolegům z práce za to, že při mně stály po celý ten čas, který jsem u diplomové práce trávil, a že na mne i přes mé „domácí vězení“ nezapomněli.

Velké děkuji patří i mé přítelkyni Nikole, která mě, spolu s tchořem Aryou, udržovala při smyslech a v kontaktu s realitou.

Děkuji i všem, kteří mi pomohli ve formě dobrovolné účasti ve sběru dat s vyhodnocením konečného produktu této práce.

Za cenné rady a doporučení při psaní mé diplomové práce děkuji i mému školiteli Ing. Miroslavu Skrbkovi, Ph.D.

OBSAH

1 Úvod

Jen obtížně lze vyloučit to, že technologie, která v příštích padesáti letech ovlivní svět nejvíce, bude umělá inteligence. Její specializované varianty se prolínají životem člověka již dnes, ať už se jedná o použití ve zdravotnictví, dopravě nebo ekonomii, a její využití neustále roste. Do mnoha oborů totiž často přináší efektivitu, prosperitu a nové příležitosti, a bude-li v budoucnu umělá inteligence zaváděna uváženě, jistě tomu tak i zůstane.

Jedním z odvětví umělé inteligence, kterému se v nynější době dostává mnoha pokroků, je i zpracování obrazu a jeho klasifikace. Každoročně jsou na tato témata vyhlašovány soutěže [1, 2, 3], ve kterých se ukazuje, že hranice přesnosti umělých neuronových sítí, které v některých disciplínách již překonávají i výkon člověka [4, 5], zdaleka ještě nejsou dosaženy. Zpracování obrazu je proto v současnosti velmi živé a fascinující téma a právě jím se zabývá i tato práce, jejíž předmětem je vytvoření systému pro automatizované čtení identifikačních údajů z fotografií osobních odkladů.

1.1 Motivace

Řada úkonů, týkajících se například manipulace s penězi, je podmíněna provedením identifikace subjektu, který o daný úkon usiluje. Tuto povinnost nařizuje například zákon č. 253/2008 Sb., o některých opatřeních proti legalizaci výnosů z trestné činnosti a financování terorismu a také zákon č. 186/2016 Sb., o hazardních hrách. Ověření totožnosti fyzické osoby, potažmo i čtení identifikačních údajů z osobních dokladů (např. občanský průkaz, cestovní pas), je proto běžnou součástí interních procesů ne mála organizací. Kromě případů, kdy je doklad předložen majitelem na místě, osobně, je pak možné se setkat se zasíláním kopie dokladu dálkovým způsobem, například elektronickou formou. Samotná procedura ověření totožnosti pak může zahrnovat buď ruční přepsání klientských údajů z dokladu do systému anebo ověření dat vyplněných samotným klientem vůči obdrženému dokladu. Oba případy jsou podmíněny prací člověka, která se s rostoucím počtem dokladů stává nepříjemnou a zdlouhavou. Právě zde by mohl budoucí systém pro vyčítání identifikačních údajů z osobních dokladů pomoci a alespoň částečně celý proces zautomatizovat a ulehčit tak práci jeho lidskému elementu.

Řešení pro čtení údajů z osobních dokladů by samozřejmě bylo možné využít při online registraci i do systémů, které explicitně kopii osobního dokladu klienta nevyžadují.

Volitelně by se tak pomocí nasnímání dokladu přes webkameru či fotoaparát chytrého telefonu mohl celý proces registrace klienta urychlit.

Jako potenciálního uživatele systému pro čtení identifikačních údajů z dokladů si lze představit například online sázkovou kancelář. Té již zmiňované zákony nařizují o klientech sbírat více informací, než je běžné. Tipsport uvádí [6], že kromě jména, příjmení, bydliště, mobilního telefonu a emailu, je pro registraci vyžadováno i datum a místo narození, pohlaví, státní občanství, rodné číslo, trvalý pobyt, typ, číslo a platnost průkazu a orgán, který průkaz vydal. Většina těchto informací by bylo možné strojově vyčíst z osobního dokladu. V neposlední řadě má o budoucí systém zájem i finančně technologická společnost GoPay s.r.o., která je zadavatelem této práce.

1.2 Cíle práce

Cílem práce je navrhnout a implementovat modul do serverové aplikace pro rozpoznávání identifikačních údajů z osobních dokladů pro účely automatizace jejich vyčítání. Vstup aplikace tvoří fotografie dokladu. Výstupem bude záznam identifikačních údajů do databáze a zpětná vazba pro uživatele o kvalitě fotografie a rozpoznaných údajích.

Cílem je zpracování obrazové informace s ohledem na různou kvalitu obrazu. Komunikaci se serverovou částí bude zajišťovat klientská aplikace schopna pořízení fotografie, detekce špatné kvality fotografie a sdělení zpětné vazby pro uživatele. Bezpečnostní aspekty projektu jsou mimo rámec této práce.

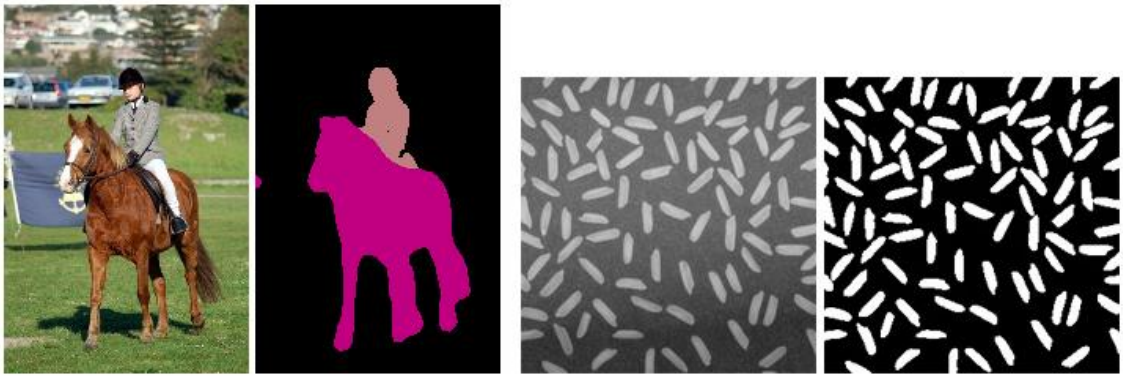
Vytvořená aplikace bude otestována a bude vyhodnocena spolehlivost rozpoznávání údajů.

2 PŘEHLED TEORIE A TERMINOLOGIE

Pro to, aby se bylo možné věnovat tématu zpracování obrazu, je nutné definovat některé pojmy a popsat některé teoretické základy, jež s daným tématem přímo souvisejí a s nimiž bude ve zbytku práce operováno. V následujících podkapitolách bude proto pozornost zaměřena zejména na témata spojená se segmentací obrazu a neuronovou sítí a jejím učením.

2.1 SEGMENTACE OBRAZU

Proces segmentace obrazu zajišťuje rozdělení obrazu na vstupu na jednotlivé segmenty tak, aby je bylo snadné rozlišit. Jednotlivé segmenty jsou rozděleny do dvou kategorií. Do kategorie objektů zájmu, které následně postupují do dalšího zpracování, a do kategorie pozadí, které pro další zpracování není zajímavé a vyřazuje se [7]. Segmentaci obrazu je proto možné prezentovat i jako proces detekce oblastí zájmu na obrázku. Většina segmentačních metod pro zjednodušení problému vynechává ze vstupního obrazu barevnou informaci a pracuje pouze s obrazem ve stupních šedi, respektive s obrazem o jednom kanálu. Výstup segmentace je možné prezentovat taktéž obrazem a jeho jednotlivé barvy pak znázorňují různé kategorie segmentů, které byly rozlišeny (viz Obr. 1). Často má pak takový výstup podobu binárního obrazu, ve kterém bílá barva symbolizuje objekt zájmu a černá barva pozadí k odfiltrování (viz Obr. 1, dvojice vpravo), nebo naopak.

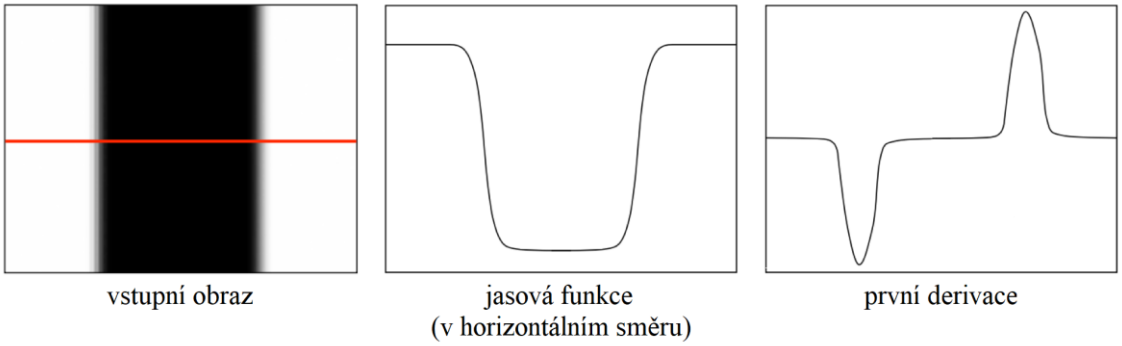


Obr. 1: Segmentace obrazu – dvojice vstup a výstup (převzato z [8, 9])

„Segmentace je nejčastěji založena na *detekci kontur (hran)* ohraničujících jednotlivé objekty nebo na *detekci celých oblastí*, kterými jsou jednotlivé objekty v obraze reprezentovány.“ [7] Příklady obou jmenovaných přístupů budou stručně přiblíženy v následujících podkapitolách.

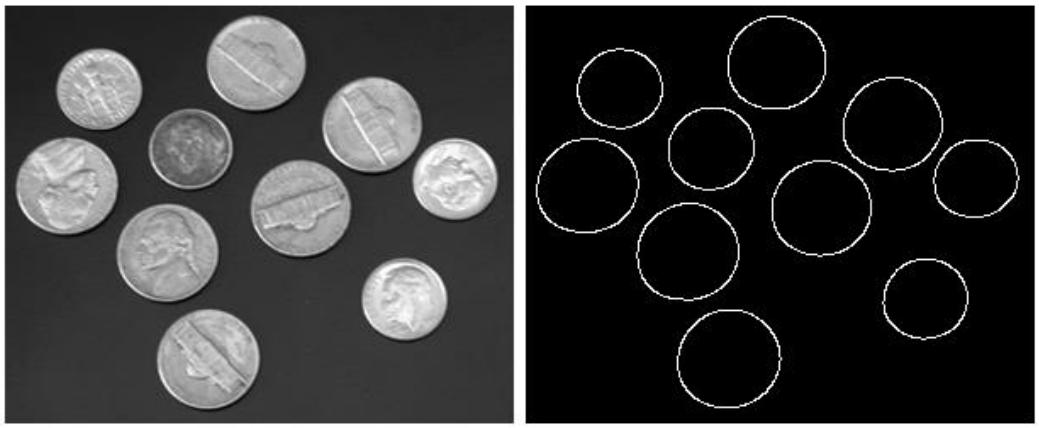
2.1.1 SEGMENTACE DETEKCÍ HRAN

Segmentace obrazu pomocí detekce hran je založena na předpokladu, že obrys zájmového objektu způsobuje v obrazu náhlou změnu jasu. Tento jev lze pozorovat poměrně spolehlivě v případech, ve kterých je objekt zájmu barevným odstínem světlejší či tmavší než jeho pozadí nebo jsou-li příznivé světelné podmínky a změny zakřivení povrchu objektu jsou doprovázeny i změnami jasu (např. v podobě stínů či odlesků). Hranami jsou potom nazývány právě takové body v obraze, ve kterých dochází k obzvlášť prudké změně jasu [10].



Obr. 2: Detekce hran na základě derivace jasové funkce (převzato z [11])

Řešení detekce hran je obvykle založeno na skutečnosti, že první derivace jasové funkce tvoří v místech hran extrémy (viz Obr. 2) [7]. Na stejném základě pracuje i Cannyho detektor hran [12], který je v současnosti považovaný za standard a jeho implementaci lze nalézt ve všech populárních knihovnách pro práci s obrazem [13, 14, 15, 16]. Výsledek detekce hran pomocí Cannyho algoritmu, který je binárního formátu, lze pozorovat na obrázku níže.



Obr. 3: Detekce hran pomocí Cannyho algoritmu – dvojice vstup a výstup (vstupní obrázek převzat z [17])

Protože detektory hran dokáží zpravidla pracovat s hranami pouze na úrovni bodů v obraze a neudávají tak na výstupu pozici ani vlastnosti objektů zájmu jako celku, je pro dokončení segmentace nutné podniknout další zpracování. Níže jsou proto popsány tři různé metody, které celý proces segmentace obrazu pomocí detekce hran uzavírají.

2.1.1.1 NALEZENÍ KONTUR

Nejjednodušší z popisovaných metod. Algoritmus pro nalezení kontur zanalyzuje binární obraz vzniklý detekcí hran a pro každou skupinu spojených bodů jedné barvy utvářející uzavřenou křivku vytvoří záznam o jejím obrysu. Tento záznam pak představuje právě jednu konturu. Zpravidla má tvar výčtu bodů polygonu a jeho rozlišení závisí na použité aproximační metodě. Volitelně lze obvykle záznam rozšířit i o informaci o globální hierarchii kontur, pomocí které je možné rozlišit kontury zapouzdřeného typu a na jejíž základě lze později kontury filtrovat [18]. Filtraci, pomocí které je možné vyřadit kontury, které nepředstavují objekty zájmu, lze provést i na základě vlastností spojených komponent popsaných v kapitole 2.1.3.

2.1.1.2 HOUGHOVA TRANSFORMACE

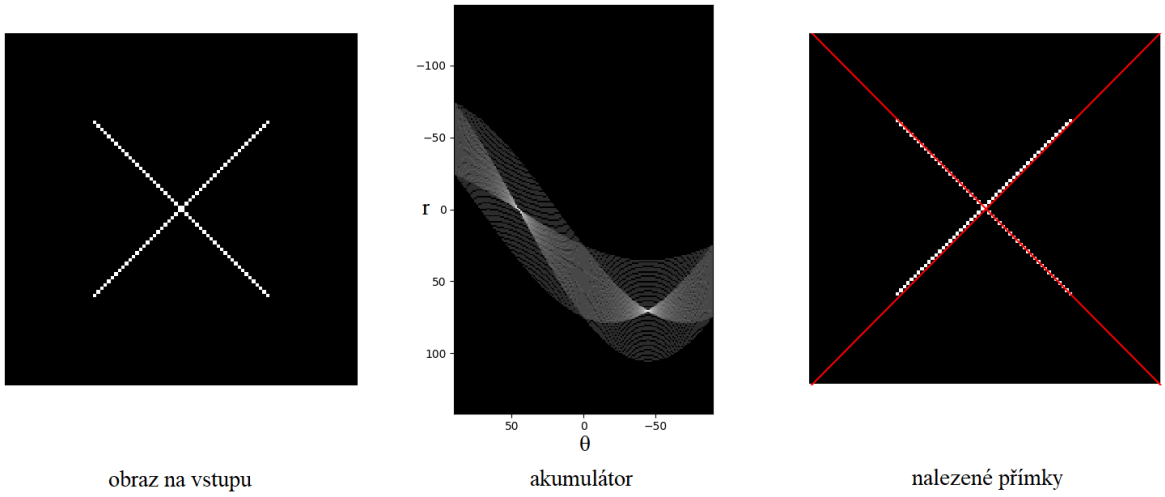
Problém segmentace obrazu je často omezen na případy, ve kterých je ve vstupním obraze hledán objekt známého tvaru. Nejsou-li však vstupní obrazová data v perfektním stavu, například vlivem šumu nebo rozostření, není možné přesně detekovat hrany v nich obsažené a použití triviálních metod, jako je nalezení kontur (viz kapitola 2.1.1.1), nepodává v takovém případě spolehlivé výsledky. Detekované hrany jsou totiž vlivem nekvalitních dat nepřesné nebo nekompletní a často neutvářejí uzavřené tvary. Nalezení objektu známého tvaru lze však řešit jeho optimálním proložením dostupnými hranami, čemuž se věnuje právě metoda známa pod názvem Houghova transformace [19].

Nejpopulárnější varianta Houghovy transformace se věnuje detekci přímek [20]. Pro svou činnost využívá normálové rovnice přímky (viz rovnice níže), ve které r je délka normály od přímky k počátku souřadnic a θ *(théta)* je úhel mezi normálou a osou x.

$$r = x\cos\theta + y\sin\theta \qquad \textbf{(1)}$$

Metoda pracuje na principu dvourozměrné akumulační matice, tzv. akumulátoru, jež představuje prostor, jehož osy tvoří právě r a θ z rovnice výše. Jsou-li do této rovnice dosazeny souřadnice některého z bodů, které byly získány detekcí hran, pak množina všech řešení (r, θ) vytvoří v prostoru akumulátoru spojitou křivku ve tvaru sinusoidy. V místech, kterými

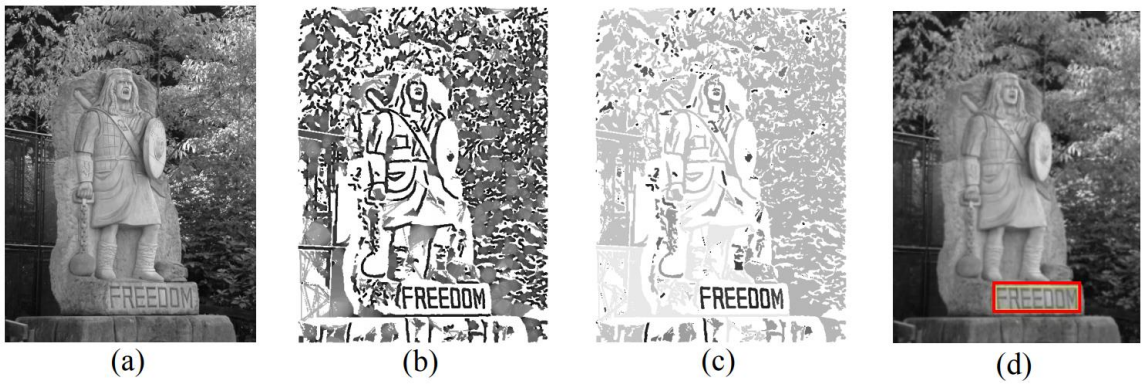
prochází tato křivka, jsou buňky akumulátoru inkrementovány. Pakliže je ve vstupním obraze přítomna přímka, reprezentovaná několika body v jednom směru, projeví se tato skutečnost lokálním maximem v akumulátoru. Je tomu tak proto, že v prostoru, který je tvořen r a θ, odpovídá jediný bod popisu právě jedné přímky. Výběrem lokálních maxim lze pak získat informace o přímkách ve vstupním obraze (viz Obr. 4) [7].



Obr. 4: Houghova transformace (převzato z [21])

2.1.1.3 Obrazový operátor SWT

Jako další způsob segmentace pomocí detekce hran lze považovat použití obrazového operátoru SWT (z angl. Stroke Width Transform) [22], který je zaměřen na detekci textu v obraze. Operátor pracuje tak, že po detekování hran pomocí Cannyho algoritmu vypočítá pro každý pixel šířku tahu (z angl. stroke), viz Obr. 5b. Následně je soubor získaných šířek zanalyzován algoritmem, který spojené komponenty s podobnými šířkami tahů označí jako kandidáty textu (viz Obr. 5c), které následně použitím heuristických pravidel zformuje do textových bloků (viz Obr. 5d) [23].



Obr. 5: Segmentace pomocí SWT (převzato z [22])

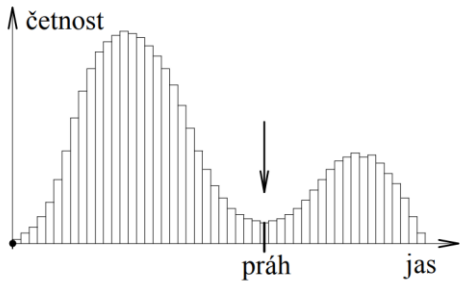
2.1.2 SEGMENTACE PRAHOVÁNÍM

Segmentace obrazu prahováním patří na rozdíl od segmentace pomocí detekce hran do skupiny založené na detekci celých oblastí. Pro svou rychlost a vyšší odolnost vůči šumu je v případech, ve kterých je ji možné použít, často před detektory hran upřednostňována [7].

Prahování je založeno na předpokladu, že body objektu zájmu mají jednolitý a od pozadí odlišitelný jas. Samotný proces prahování pak spočívá v tom, že na základě zvolené hodnoty prahu *t*, je každý bod vstupního obrazu podle jeho hodnoty jasu převeden buď na hodnotu pozitivního nálezu objektu, 1, nebo na hodnotu negativního nálezu, 0, podle předpisu níže [ibid.]. Výstupem prahování je pak binární obraz (viz Obr. 7).

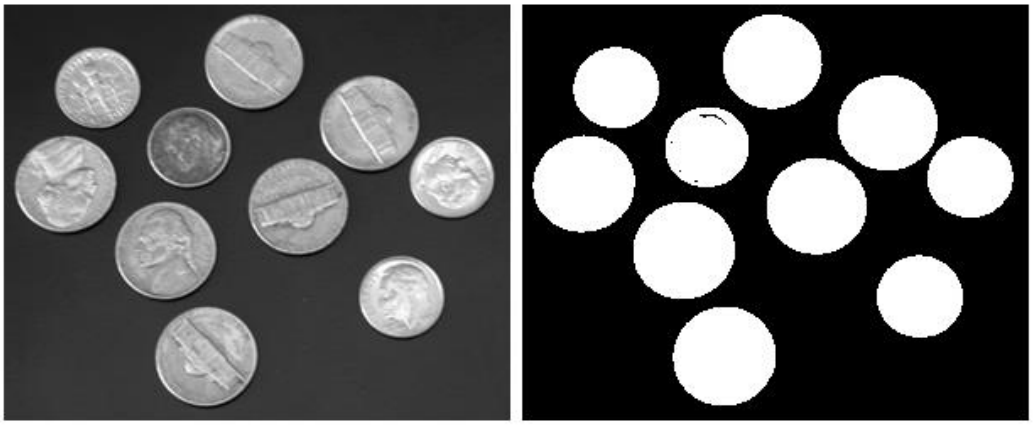
$$g(x,y) = \begin{cases} 1, f(x,y) \geq t \\ 0, \quad jinak \end{cases} \tag{2}$$

Úspěšnost prahování závisí na zvolené hodnotě prahu *t*, která však ve většině případů není předem známa, a tak je ji nutné prvně určit. Pokud platí předpoklad naznačený v předchozím odstavci, tedy že objekt zájmu je jasem odlišitelný od pozadí, pak, je-li sestrojen histogram jasu takového obrazu, bude bimodálního typu (histogram se dvěma vrcholy). Ideální hodnota prahu *t* je pak často v jasové úrovni mezi oběma vrcholy histogramu (viz Obr. 6) [7]. Pro automatické vyhledání ideální prahové hodnoty lze využít bezparametrické Otsuovy metody [24].



Obr. 6: Určení prahu v bimodálním histogramu jasu (převzato z [7])

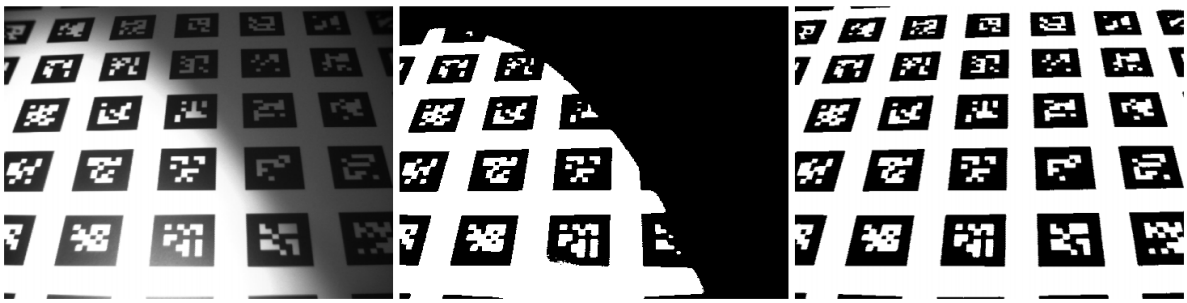
Po získání výstupního obrazu v binárním formátu je pak postup segmentace podobný jako v případě nalézání kontur (viz kapitola 2.1.1.1) s tím rozdílem, že nyní algoritmus pro sběr objektů v obraze hledá jakékoliv spojené komponenty pozitivních nálezů bez dalšího kritéria. Nalezené objekty pak lze dále filtrovat například pomocí vlastností spojených komponent (viz kapitola 2.1.3). Výsledný binární obraz lze přímo využít i jako masku originálního obrazu na vstupu tak, aby odfiltroval pozadí a ponechal jen data náležící objektům zájmu, které pak již pro další zpracování nemusí být omezeny jen na úroveň jasu.



Obr. 7: Segmentace pomocí prahování – dvojice vstup a výstup (vstupní obrázek převzat z [17])

2.1.2.1 ADAPTIVNÍ PRAHOVÁNÍ

V důsledku například nerovnoměrného osvětlení vstupního obrazu (viz Obr. 8 vlevo) se lze setkat se situacemi, ve kterých není možné určit takovou hodnotu prahu, aby objekty zájmu od pozadí oddělila bez chyb (viz Obr. 8 uprostřed). V takových případech je možné využít takzvaného adaptivního prahování, které namísto aby pracovalo s globální hodnotou prahu pro celý obraz, jako tomu je u standardního prahování, pracuje s prahem lokálním. Hodnotu lokálního prahu potom algoritmus mění v závislosti na okolí zpracovávaného bodu v obraze a je tímto způsobem schopen zpracovat i obrazy s velmi nehomogenním osvětlením (viz Obr. 8 vpravo).



Obr. 8: Porovnání segmentace Otsuovou metodou (uprostřed) a adaptivním prahováním (vpravo) – obrázek vlevo a vpravo převzat z [26]

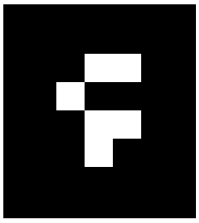
Algoritmus adaptivního prahování lze implementovat například volbou prahu v závislosti na průměru hodnot jasu v okolí, které je dáno klouzavým oknem [27]. Využít lze ale i některé ze sofistikovanějších metod, jako je Niblackovo prahování [25], Bradleyho prahování [26], Fengovo prahování [28] nebo Sauvolaho prahování [29].

2.1.3 VLASTNOSTI SPOJENÝCH KOMPONENT

Po úspěšné segmentaci obrazu například prahováním (viz kapitola 2.1.2) vzniká binární obraz, z něhož jsou pomocí nalezení spojených komponent extrahovány všechny potenciální objekty zájmu. Ne všechny z těchto objektů jsou však těmi opravdu hledanými a k jejich rozlišení je možné použít právě jejich vlastnosti, zejména pak popis jejich tvaru. Popis tvaru je vypočítán z plošného rozložení bodů (pixelů) náležících objektu. Způsobů, jak popsat tvar objektu, je mnoho a kromě některých vybraných, které jsou přiblíženy v následujících podkapitolách, je možné se setkat mnohými dalšími (např. obvod, kompaktnost, orientace, pravoúhlost, …) [30].

2.1.3.1 VELIKOST

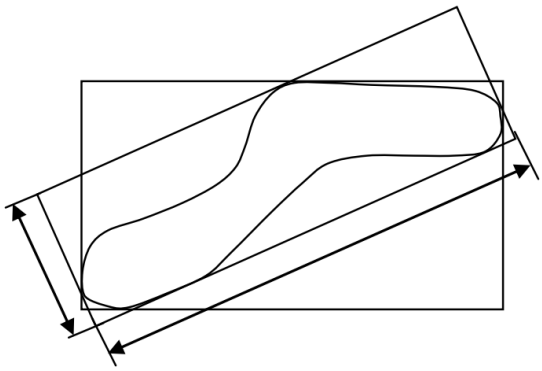
Nejzákladnější vlastnost spojených komponent, která definuje jejich obsah. Je dána počtem bodů, které náleží dané komponentě [30]. V anglické literatuře je vyjadřována pojmem *Area* [31].



Obr. 9: Velikost spojené komponenty

2.1.3.2 PODLOUHLOST

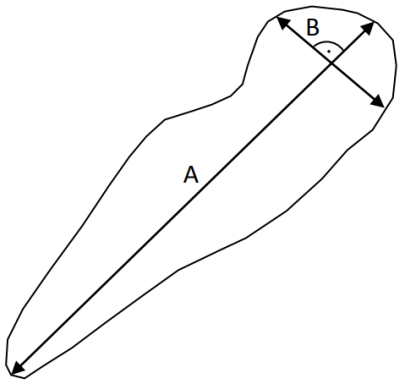
Podlouhlost, nebo také *roztažení*, vyjadřuje poměr mezi délkou a šířkou obdélníku s nejmenším možným obsahem, který daný objekt ohraničuje [32]. V anglické literatuře je vyjadřována pojmem *Aspect ratio* [31].



Obr. 10: Podlouhlost spojené komponenty (převzato z [33])

2.1.3.3 VÝSTŘEDNOST

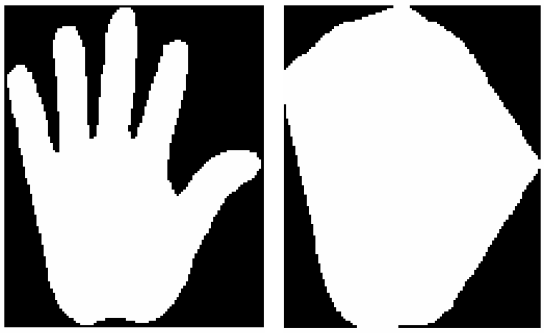
Výstřednost, nebo také *excentricita*, je nejjednodušeji charakterizována jako poměr délek nejdelší tětivy *A* a nejdelší k ní kolmé tětivy *B* (viz obrázek níže) [34]. V anglické literatuře je vyjadřována pojmem *Eccentricity* [ibid.].



Obr. 11: Výstřednost spojené komponenty (převzato z [33])

2.1.3.4 KONVEXNOST

Konvexnost spojené komponenty udává míru podobnosti objektu ke své konvexní schránce. Je dána poměrem své velikosti a velikosti svého konvexního obalu [30]. V anglické literatuře je vyjadřována pojmem *Solidity* [31].



Obr. 12: Spojená komponenta (vlevo) a její konvexní obal (vpravo) – převzato z [30]

2.2 NEURONOVÁ SÍŤ

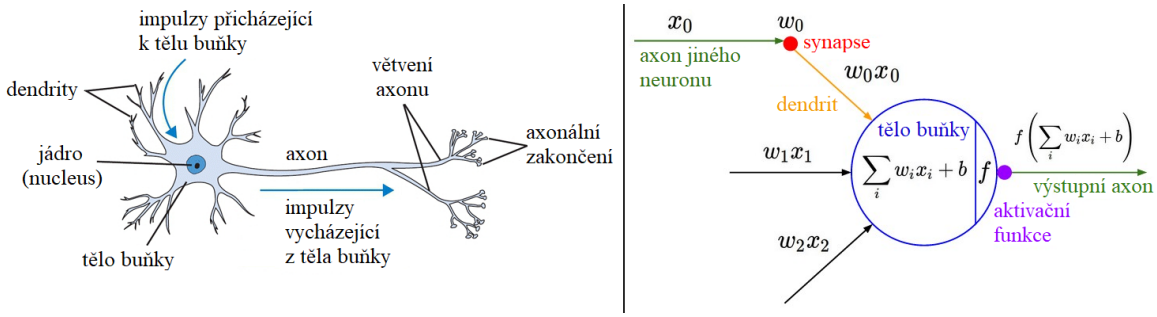
Valná většina úloh, které jsou založeny na transformaci vstupního vektoru na výstupní vektor a které je člověk schopen řešit rychle a ve velkém množství, je za předpokladu dostatečně velkého modelu a dostatečného množství trénovacích dat řešitelná pomocí neuronové sítě [35]. Neuronové sítě se již dnes úspěšně využívají například pro rozpoznávání (klasifikaci) objektů v obraze [1], které bývá prováděno jako navazující krok ihned po jejich segmentaci (viz kapitola 2.1) a jímž se značnou měrou zabývá i tato práce. V následujících podkapitolách

proto budou popsány základní principy a terminologie neuronových sítí, přičemž pozornost bude věnována pouze neuronovým sítím *dopředného typu* (viz kapitola 2.2.2), se kterými se je možné v praxi setkat nejčastěji a které jsou využity i v této práci.

2.2.1 NEURON

Původní myšlenkou, která motivovala k bádání v oblasti umělých neuronových sítí, bylo napodobit jimi funkčnost biologického nervového systému. Tato myšlenka však postupem času ustoupila praktickým úlohám především jejich optimalizaci pro strojové učení a v kontextu moderních umělých neuronových sítí tak zůstává spíše historickou vzpomínkou [36]. I přesto je však umělý neuron, jakožto základní jednotka umělé neuronové sítě, často uváděn v přirovnání k jeho biologickému protějšku a stejným způsobem bude k jeho popisu přistupováno i v následujících odstavcích.

Neuron je základní výpočetní jednotkou biologického mozku. V lidském nervovém systému jich je přibližně 86 miliard a jsou mezi sebou propojeny přibližně 10^{14} až 10^{15} *synapsemi*. Na obrázku níže je nalevo vyobrazena kresba biologického neuronu a napravo jeho matematický model, známý také pod pojmem *perceptron*. Každý neuron přijímá vstupní signály pomocí svých *dendritů* a produkuje výstupní signály svým (jediným) *axonem*. Axon se pak dále větví a pojí se synapsemi k dendritům dalších neuronů. V matematickém modelu neuronu signály putující axony (x_0) interagují s dendrity napojeného neuronu pomocí násobení ($w_0 x_0$) na základě síly dané synapse (w_0). Základní mechanismus tkví v tom, že míry sil synapsí (tzv. váhy[1] w) řídí vliv jednoho neuronu na další a jsou spolu s prahem[2] (b) ovlivněny učením. Signály přenesené dendrity k tělu buňky jsou následně sečteny a dosáhl-li součet určité kritické meze, je vytvořen nový impulz, který axonem putuje k dalšímu neuronu. V matematickém modelu neuronu je frekvence těchto impulzů reprezentována *aktivační funkcí* (f) [36].

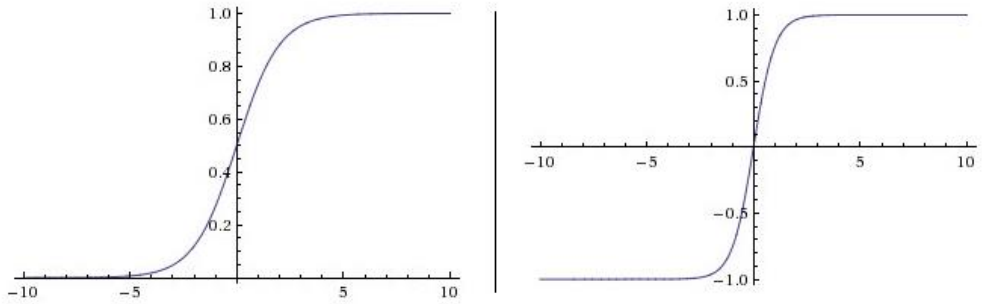


Obr. 13: Biologický neuron (vlevo) a jeho matematický model (vpravo) – převzato z [36]

[1] Z anglického slova *weight*
[2] Z anglického slova *bias*

2.2.1.1 AKTIVAČNÍ FUNKCE NEURONU

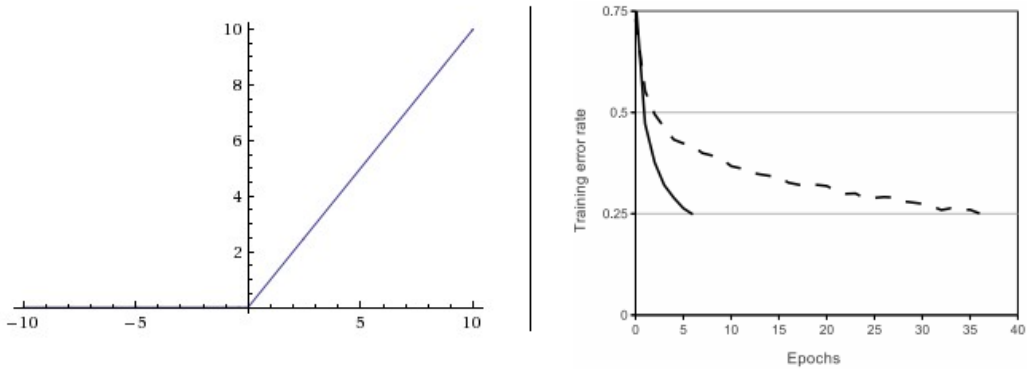
Aktivační funkce neuronu je nelineární funkce o jednom vstupním parametru, na kterém provádí určité neměnné matematické operace. Historicky byla volena funkce sigmoidy (viz Obr. 14 vlevo) nebo hyperbolického tangens (viz Obr. 14 vpravo), postupně se od nich ale začalo opouštět ve prospěch nesaturovaných[3] funkcí, které při učení neuronové sítě (viz kapitola 2.2.3) vykazují lepší vlastnosti [36].



Obr. 14: Průběh funkcí sigmoida (vlevo) a hyperbolický tangens (vpravo) – převzato z [36]

Zřejmě nejdoporučovanější aktivační funkcí je k dnešnímu dni takzvaná ReLU[4] funkce (viz rovnice níže a Obr. 15 vlevo), jejíž použitím například Krizhevsky et al. [38] pozorovali oproti tradičním aktivačním funkcím šestinásobné zrychlení učení neuronové sítě vzhledem k počtu provedených trénovacích kroků (viz Obr. 15 vpravo). Věří se, že je toto zrychlení spojeno s tím, že ReLU funkce není saturovaná[3] [36].

$$f(x) = \max(0, x) \tag{3}$$



Obr. 15: Vlevo průběh ReLU funkce a vpravo průběh učení s ReLU funkcí (plná křivka) oproti učení s tanh funkcí (čárkovaná křivka) – převzato z [36]

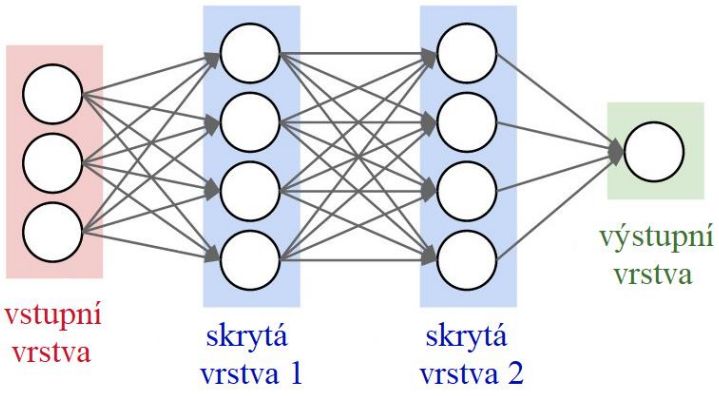
[3] Saturační charakter funkce znamená, že se od určité hodnoty vstupního argumentu již výstupní signál funkce dále nezvyšuje / nesnižuje [37].

[4] Z anglického sousloví *rectified linear unit*

2.2.2 DOPŘEDNÁ NEURONOVÁ SÍŤ

Dopředná neuronová síť (z angl. feedforward neural network), známá také pod názvem *vícevrstvá perceptronová síť* (z angl. multilayer perceptron network), je charakteristická tím, že obsahuje neurony uskupené do shluků tak, že společně tvoří acyklický graf. Znamená to, že výstupy některých neuronů jsou přivedeny na vstup jiných, přičemž jejich výstup již nesmí být napojen na takový neuron, který byl, byť i nepřímo, předtím součást jejich vstupu. Shluky neuronů jsou v neuronových sítích často organizovány do vrstev, přičemž nejčastěji je možné se setkat s vrstvami takzvaně *plně propojenými* (z angl. fully connected layer), ve kterých jsou neurony v sousedních vrstvách propojeny každý s každým, zatímco neurony v rámci jedné takové vrstvy nesdílejí žádné propojení (viz Obr. 16) [36].

Společný cíl pro všechny dopředné neuronové sítě spočívá v aproximaci funkce f^*. Například, pro síť typu klasifikátor, funkce $y = f^*(x)$ mapuje vstup x do kategorie y. Aby bylo dosaženo co nejlepší aproximace podobné funkce, definuje dopředná neuronová síť mapování $y = f(x; \theta)$, pro něž je schopna se naučit co nejideálnější parametry θ (viz kapitola 2.2.3), odpovídající vahám a prahům obsažených neuronů. Cennou vlastností neuronových sítí je kromě aproximace funkce i jejich schopnost *generalizace,* neboli schopnost správně reagovat na data, která nebyla součástí procesu učení [35].



Obr. 16: Příklad dopředné neuronové sítě (převzato z [36])

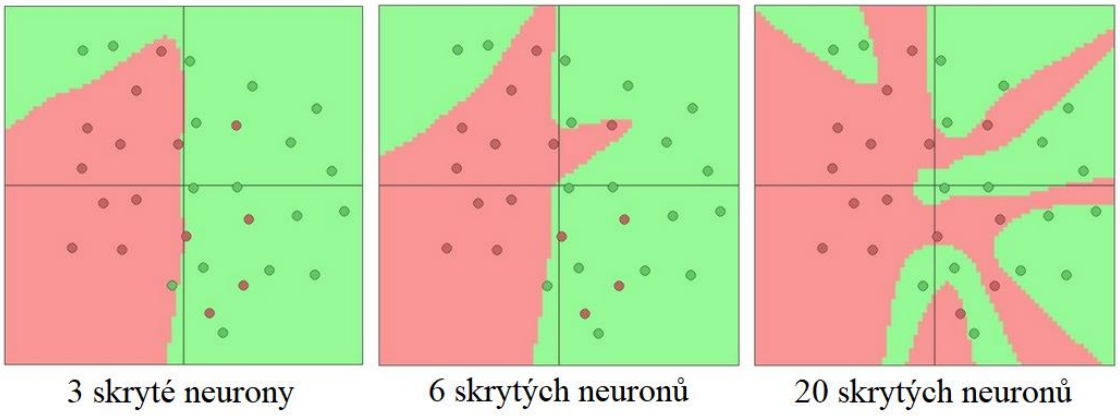
Dopředné neuronové sítě jsou nazývány sítěmi, protože jsou typicky tvořeny zřetězením mnoha různých funkcí. Jako příklad lze uvést dopřednou neuronovou síť tvořenou funkcí $f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x)))$, jež odpovídá i neuronové síti na obrázku 16. V tomto případě x odpovídá *vstupní vrstvě,* v níž se přímo neodehrávají žádné výpočetní operace, $f^{(1)}$ představuje první vrstvu neuronové sítě (první skrytou vrstvu), $f^{(2)}$ druhou vrstvu (druhou skrytou vrstvu) a tak dále. Poslední funkce ($f^{(3)}$) je vždy nazývána *výstupní vrstvou* neuronové sítě. Celkový počet takových zřetězených funkcí/vrstev pak odpovídá *hloubce* neuronové sítě.

Protože vstupní vrstva slouží pouze jako prostředek k předání vstupních dat, nezapočítává se do celkové hloubky sítě a na obrázku 16 je proto dle jmenné konvence zobrazena síť o třech vrstvách. Další metrikou neuronové sítě je její *šířka*. Ta je definována dimenzionalitou skrytých vrstev, která je dána počtem neuronů v nich obsažených. Neuronová síť jako celek ve smyslu její architektury je běžně označována pojmem *model* [35].

2.2.2.1 NÁVRH DOPŘEDNÉ NEURONOVÉ SÍTĚ

Zatímco návrh vstupní a výstupní vrstvy neuronové sítě je jasně dán povahou vstupních dat a požadovaným výstupem, určení počtu a vlastností skrytých vrstev není zdaleka tak přímočaré. Pro návrh skrytých vrstev totiž neexistují žádná pravidla, na jejichž základě by bylo možné ihned sestavit optimální neuronovou síť, a jejich realizace tak zůstává spíše předmětem obecných pouček a intuice [39].

Vliv různého počtu neuronů ve skryté vrstvě dvouvrstvé plně propojené neuronové sítě lze pozorovat na obrázku 17. Neuronová síť je zde použita jako binární klasifikátor dvourozměrného prostoru. Z obrázku je možné vypozorovat, že s přibývajícími neurony ve skryté vrstvě roste i komplexnost produkované separační funkce – roste *kapacita* neuronové sítě. Na první pohled se může zdát, že síť s dvaceti neurony si v separaci obou tříd vede nejlépe, pravidlem to však být nemusí. Neuronová síť s nejvyšší kapacitou zde má totiž sklony k takzvanému přeučení (viz kapitola 2.2.3), což se projevuje tím, že sice dokáže pomocí složité funkce bezchybně určit data při trénování (viz obrázek níže), ale nedokáže již tak dobře oddělit data, která ji při učení nebyla předložena. Tento nedostatek generalizace lze vysvětlit tím, že data, která síť o dvaceti neuronech oddělila v prostoru vpravo, představují spíše šum, než charakteristickou oblast dané třídy dat, a modely o nižších počtech neuronů by tak v praxi mohly dosahovat lepší generalizace [36].



Obr. 17: Vliv počtu skrytých neuronů na kapacitu neuronové sítě (převzato z [36])

Co se počtu skrytých vrstev neuronové sítě týká, Cybenkot [40] v roce 1989 předvedl, že již s jedinou skrytou vrstvou (dvouvrstvá neuronová síť) lze za předpokladu dostatečného množství skrytých neuronů aproximovat jakoukoliv spojitou funkci. Empirickým pozorováním se však navzdory tomuto faktu ukázalo, že hlubší neuronové sítě fungují o něco lépe a v praktickém nasazení se proto lze velmi často setkat s jejich třívrstvými variantami [36].

2.2.3 Učení neuronové sítě

Cílem učení neuronové sítě je nastavit parametry neuronové sítě (např. v podobě vah a prahů) tak, aby transformací přiložených vstupních hodnot na výstupní v dané aplikaci vytvářely co nejuspokojivější odezvu [41]. Ačkoliv samotná transformace vstupních hodnot na výstupní je v rámci neuronových sítí velmi rychlá operace a bývá dokončena v řádech milisekund [42], učení neuronové sítě je proces, který je i při využití nejmodernějších počítačových technologií stále značně časově i výpočetně náročný a v případě velmi hluboké neuronové sítě může trvat několik dnů, někdy i týdnů [38, 74]. Je-li navíc započtena i skutečnost, že mnoho rozhodnutí, ovlivňujících výslednou přesnost naučení neuronové sítě, nelze jednoznačně určit předem a je předmětem pokusů a omylů, není výjimkou, že dosažení uspokojivě naučené neuronové sítě přichází až po uplynutí několika měsíců. Po naučení bývá však neuronová síť nasazena do praktických úloh právě ve variantě transformace vstupních hodnot na výstupní, bez schopnosti dalšího učení, a proces učení je tak pouze jednorázový.

Nalezení ideálních parametrů neuronové sítě je obecně možné provést dvěma způsoby. První, s názvem *učení s učitelem*, spočívá v tom, že je výstup neuronové sítě po předložení předem připraveného vstupu porovnán s očekávanou výstupní hodnotou. Parametry neuronové sítě jsou pak upravovány tak, aby rozdíl mezi těmito dvěma hodnotami byl minimální [41]. Množina předem připravených hodnot vstupů a očekávaných výstupů je známa pod pojmem *trénovací množina*. Druhý způsob učení neuronové sítě je nazýván *učení bez učitele*. Trénovací množina je v tomto případě složena pouze z předem připravených vstupních hodnot a návaznost na očekávaný výstup v ní chybí. Parametry neuronové sítě jsou pak nastavovány tak, aby její výstup byl konzistentní, respektive aby síť poskytovala stejnou odezvu na takové vstupní signály, které jsou stejné nebo mají podobné hodnoty. Učící proces ve variantě učení bez učitele tak ve skutečnosti provádí shlukovou analýzu vstupních dat [ibid.]. Následný text bude věnován výhradně variantě učení s učitelem, která je použita i v praktické části této práce.

2.2.3.1 PROCES UČENÍ NEURONOVÉ SÍTĚ

Učení neuronové sítě spočívá v nalezení shody její funkce $f(x)$ s funkcí $f^*(x)$, která představuje (pomyslnou) ideální funkci pro řešení dané úlohy. Trénovací množina, s níž algoritmus učení neuronové sítě pracuje, přitom představuje mapování $f^*(x)$ v různých bodech, pro které obsahuje, s různou měrou zašumění a přesnosti, trénovací vzory. Každý vzor x je v trénovací množině doprovázen i očekávaným výstupem (angl. označován jako *label*) $y \approx f^*(x)$. Trénovací vzory tak přímo určují, jak mají v každém bodě x vypadat hodnoty na výstupní vrstvě sítě. Zatímco výstupní vrstva je vedena k produkci hodnot blízkých y, chování ostatních (skrytých) vrstev neuronové sítě trénovací množina nijak neurčuje. Je pak na algoritmu učení neuronové sítě, aby rozhodl, jak tyto vrstvy použít tak, aby síť co nejlépe aproximovala f^* [35].

Téměř všechny algoritmy, které se v současnosti pro učení neuronových sítí používají, jsou založeny na minimalizaci *chybové funkce sítě* [5] za pomoci výpočtu *gradientu* [35]. Aby takové učení bylo možné, je před započetím učení nutné tuto chybovou funkci definovat a připojit ji po čas učení na výstupní vrstvu neuronové sítě. Protože algoritmus učení se bude chybovou funkci E snažit minimalizovat, musí její průběh odpovídat míře odlišnosti mezi požadovaným výstupem sítě y a skutečným výstupem sítě $\hat{y} = f(x;\theta)$, kde x je vzor z trénovací množiny a θ jsou parametry neuronové sítě (např. v podobě vah a prahů). Výstupem samotné chybové funkce je vždy skalární hodnota. Jako typický příklad chybové funkce je často uváděna takzvaná *Mean Squared Error* (MSE) funkce (viz rovnice (4)), v praxi se ale při užití neuronové sítě jako klasifikátor používá spíše takzvaná *Cross Entropy* funkce (viz rovnice (5)) [43]. V obou rovnicích n představuje počet vzorů v trénovací množině.

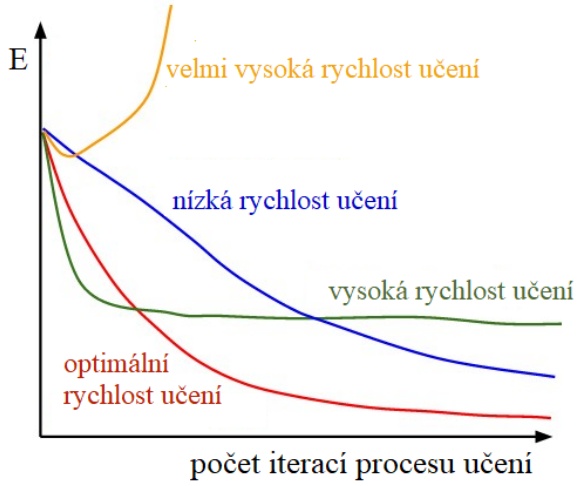
$$\mathrm{E} = \frac{1}{n}\sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})^2 \tag{4}$$

$$\mathrm{E} = -\frac{1}{n}\sum_{i=1}^{n}\left[y^{(i)}\log\left(\hat{y}^{(i)}\right) + \left(1 - y^{(i)}\right)\log\left(1 - \hat{y}^{(i)}\right)\right] \tag{5}$$

Další krok v procesu učení neuronové sítě spočívá v analýze průběhu stanovené chybové funkce E v závislosti na trénovacích vzorech a změnách jejích parametrů θ tak, aby změny parametrů θ vyvolaly její pokles. Technika, která se za tímto účelem využívá, je známa pod pojmem *Gradient Descent* (česky klesání podle gradientu), která pro určení sestupného

[5] Z anglického sousloví *error function*. Známa také jako *cost function* nebo *loss function* [35].

trendu chybové funkce využívá derivací, respektive parciálních derivací chybové funkce podle parametrů θ. Soubor těchto parciálních derivací tvoří *gradient* a jeho hodnota určuje sklon chybové funkce, na jehož základě lze určit takové úpravy parametrů sítě θ, aby vyvolaly její pokles. Nalezení těchto gradientů je v kontextu vícevrstvých neuronových sítí realizováno pomocí metody zpětného šíření chyby (z angl. *Back-Propagation*, viz [44]), pomocí jíž lze gradienty určit, na rozdíl od jejich numerických evaluací, výpočetně nenáročnou cestou. Nové hodnoty parametrů θ a velikost jejich úprav jsou pak určeny na základě kladné skalární veličiny zvané *rychlost učení* (z angl. learning rate). Protože technika klesání podle gradientu upravuje parametry θ podle vzorů v trénovací množině iterativně, musí být pro úspěšné nalezení minima průběhu chybové funkce hodnota rychlosti učení dostatečně malá, aby minimum neminula nebo nad ním neoscilovala, zároveň ale i dostatečně velká na to, aby k nálezu došlo v použitelném čase (viz Obr. 18) [35].

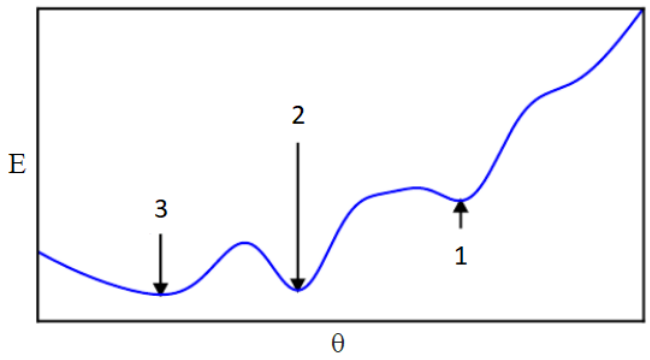


Obr. 18: Vliv hodnoty rychlosti učení na průběh chybové funkce při učení neuronové sítě (převzato z [36])

Bod, ve kterém hodnota chybové funkce dosahuje absolutního minima, se nazývá jejím *globálním minimem*. Bodů, které odpovídají globálnímu minimu, může mít tato funkce více, častěji jsou ale ostatní minima pouze lokálního charakteru, známa jako *lokální minima*. Proces učení, který chybovou funkci optimalizuje k jejímu minimu, musí cestou k uspokojivému výsledku čelit mnoha nástrahám v podobě jak lokálních minim, ve kterých může v důsledku nízké rychlosti učení uváznout, tak v podobě plochého průběhu chybové funkce, ve kterém má obtíže pomocí gradientu určit směr, na základě kterého by upravil parametry sítě. Chybová funkce mívá navíc při současných neuronových sítích mnoho milionů parametrů, které problém optimalizace dále zesložiťují. V rámci učení hluboké neuronové sítě je proto za

uspokojivý výsledek označován i takový bod chybové funkce, který není přímo jejím globálním minimem, ale jehož hodnota je globálnímu minimu blízká [35].

Na obrázku 19 je vyobrazena velmi zjednodušená chybová funkce o jediném vstupním parametru, která proces hledání uspokojivého minima ilustruje. Lokální minimum označené číslem 1 je od globálního minima označeného číslem 3 hodnotou E příliš vzdálené, a pokud by na tomto bodě učící algoritmus uvázl, síť by pravděpodobně nepodávala uspokojivé výsledky. Lokální minimum číslo 2 je naopak globálnímu minimu velmi blízké, a pokud by optimalizační proces skončil právě na jeho místě, pravděpodobně by se jednalo o velmi uspokojivý výsledek.



Obr. 19: Lokální minima (1, 2) a globální minimum (3) zjednodušené chybové funkce (převzato z [35])

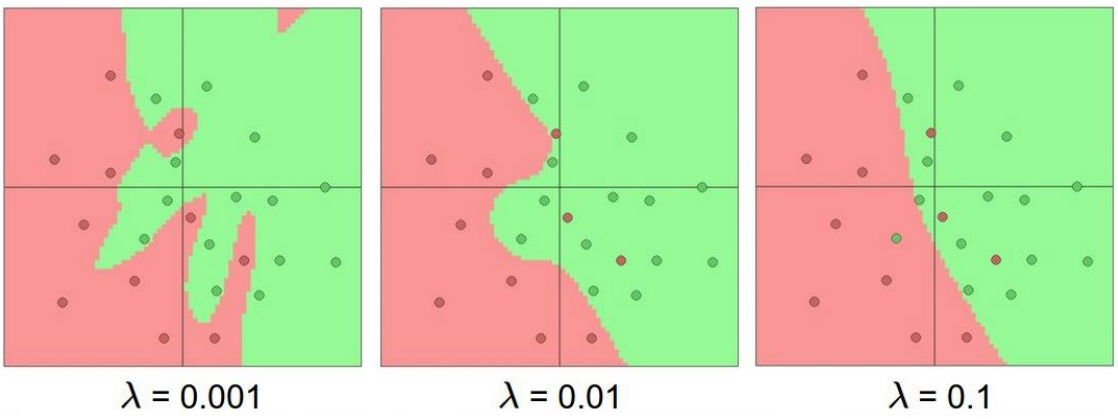
V praxi je pro úlohu optimalizace chybové funkce obvykle využívána i *setrvačnost*, která často dokáže celý proces učení neuronové sítě urychlit [39]. Setrvačnost (z angl. momentum) je algoritmus, jenž do výpočtu změn parametrů neuronové sítě při jejím učení metodou Gradient Decent zavádí vazbu na změnu parametrů z předchozí iterace a propůjčuje tak celému procesu učení krátkodobou paměť. Jak je již z jeho názvu patrné, jeho zakomponováním získá proces optimalizace chybové funkce určitý element setrvačnosti, který v případě stabilních gradientů posílí rychlost učení v jejich směru a v případě menších lokálních minim pomůže stabilizovat jejich průchod [45]. Čím víc se navíc proces učení chýlí ke konci, tím menší jsou i změny parametrů sítě a tedy i setrvačnost, která tak nebrání dosažení konečného (globálního) minima chybové funkce.

2.2.3.2 REGULARIZACE

V kapitole 2.2.2.1, zabývající se návrhem architektury neuronové sítě, byl na příkladu v podobě obrázku 17 popsán vliv počtu skrytých neuronů na výslednou kapacitu celého modelu. Na příkladu zde bylo předvedeno, že vysoká kapacita ne vždy vede k lepším

výsledkům a že síť může vzhledem k vyšší kapacitě ztrácet schopnost generalizace. Tato ztráta generalizace spojená s přílišným přizpůsobením se trénovací množině je nazývána *přeučení* (z angl. overfitting). Přeučení obecně není žádané, a aby mu bylo zabráněno, jsou používány různé takzvané *regularizační techniky*, které dokáží manipulovat i s celkovou kapacitou neuronové sítě.

Příklad na obrázku 17 může budit dojem, že pro úlohy, které nejsou příliš komplexní, je pro prevenci přeučení vhodné volit menší počet skrytých neuronů. Ve skutečnosti ale takový postup často nevede k optimálním výsledkům, protože optimalizace chybové funkce sítí s menším množstvím parametrů má tendence při učení pomocí metody Gradient Decent uváznout v lokálních minimech, které nejsou jako výsledek učení uspokojivé. Sítě s vyšším počtem parametrů však tímto neduhem tolik netrpí, a i přesto, že jejich chybové funkce mají lokálních minim často více než jejich mělčí varianty, učení mnohem konzistentněji končí v přijatelnějších konstelacích [36]. Pro prevenci přeučení je proto v praktických úlohách před snižováním počtu skrytých neuronů preferována regularizace, jejíž efekt lze pozorovat na obrázku 20. Na obrázku je použita stejná síť s dvaceti neurony jako na obrázku 17 vpravo, avšak tentokrát je při učení použita různá síla regularizace λ. Ve směru zleva doprava je možné pozorovat, jak se zvyšující se silou regularizace λ klesá kapacita sítě a tím potenciálně roste její schopnost generalizace za současného zachování vlastností učení hluboké neuronové sítě. Hodnotu síly regularizace je samozřejmě nutné volit s citem tak, aby kapacita sítě neklesla příliš a síť byla v dané úloze stále schopna podávat uspokojivé výsledky.



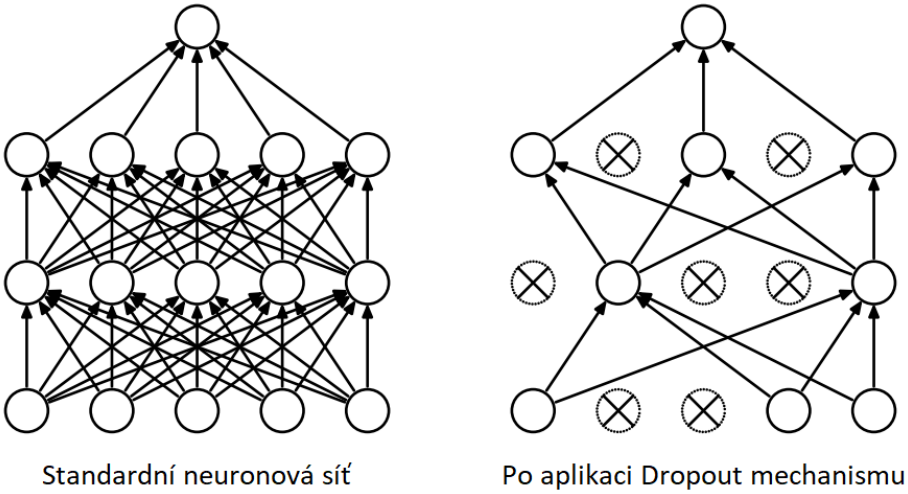
Obr. 20: Vliv síly regularizace na kapacitu neuronové sítě (převzato z [36])

Jedna z nejpoužívanějších technik regularizace je známa pod anglickým názvem *Weight Decay* nebo také *L2 regularizace*. Tato technika spočívá v modifikaci chybové funkce sítě tak, že k ní přidá sumu čtverců všech vah v síti, jejíž konečnou hodnotu řídí silou regularizace λ, viz rovnice (6).

$$\mathrm{E}_r = \mathrm{E} + \frac{\lambda}{2n}\sum_w w^2 \qquad \textbf{(6)}$$

Z rovnice výše lze vyvodit, že činnost L2 regularizace spočívá, vzhledem k optimalizaci chybové funkce minimalizací, v preferenci malých hodnot vah w před velkými, které jsou kvůli kvadrátu penalizovány více. Proces učení se tak ubírá k vysokým hodnotám vah pouze tehdy, je-li jejich vlivem značně minimalizována první, původní část chybové funkce (E). Penalizace vysokých hodnot vah má za následek to, že je síť učením vedena k rozpoznávání spíše obecných znaků trénovacích dat, než přítomného šumu, který by vedl k přeučení [39].

Další často využívaná forma regularizace je z angličtiny nazývána *Dropout* (viz [46]). Na rozdíl od L2 regularizace, která manipuluje s chybovou funkcí neuronové sítě, je Dropout založen na úpravě sítě samotné. Během učení totiž zavádí pravděpodobnost p, jež je hyperparametrem a volí se před začátkem učení (viz kapitola 2.2.3.3), na základě které během každého kroku učení rozhoduje, zda skrytý neuron zůstane ponechán, nebo bude ze sítě včetně jeho spojení dočasně vyřazen (viz Obr. 21). Díky tomuto mechanismu Dropout během učení z původní architektury sítě formuje mnoho různých podsítí a tím, že je neuronová síť nucena nespoléhat se na každý ze svých dílčích neuronů, zabraňuje jejímu přeučení a přispívá k její schopnosti generalizace [39].



Obr. 21: Ilustrace Dropout mechanismu (převzato z [46])

2.2.3.3 Další důležité pojmy

V následujících odstavcích budou postupně představeny a stručně popsány další důležité pojmy související s učením neuronových sítí, které budou plnit zejména funkci doplnění informací k tématům z předešlých kapitol.

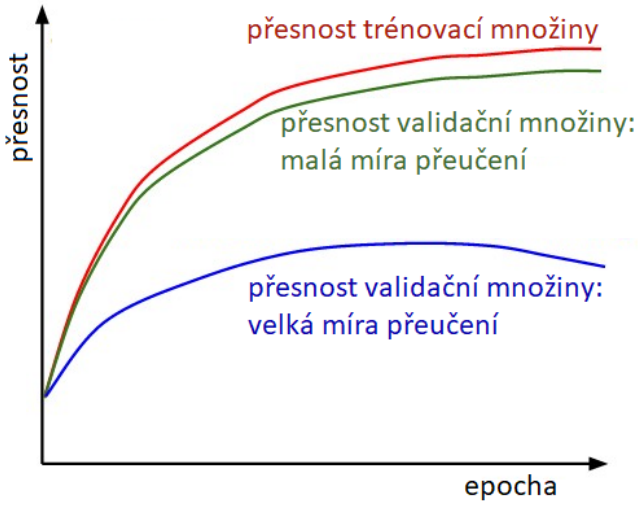
Stochastic Gradient Descent. Podobně jako Gradient Descent, představený v kapitole 2.2.3.1, je i Stochastic Gradient Descent algoritmus zaměřený na optimalizaci chybové funkce, kterou pomocí úprav parametrů neuronové sítě na základě gradientů minimalizuje. Na rozdíl od Gradient Descent, který každou úpravu parametrů sítě zakládá na sledování gradientu celé trénovací množiny, stochastický Gradient Descent používá ke stejnému účelu jen její části a znatelně tak celý proces učení urychluje. Velikost těchto částí, která zůstává po dobu učení neměnná, je anglicky nazývána *batch size* (česky velikost dávky) a část samotná je známa pod anglickým názvem *minibatch*. Protože jsou nároky na velikost trénovacích množin pro účely učení hlubokých neuronových sítí s časem a s rostoucí náročností úloh neustále zvyšovány a není výjimkou, že trénovací množina obsahuje trénovací vzory o součtu v řádech desítek milionů [47], nelze již standardní Gradient Descent v mnoha případech z praktických důvodů použít a jeho stochastická varianta je proto v současnosti v oboru strojového učení nejpoužívanějším optimalizačním algoritmem [35].

Epocha. Učení neuronové sítě je založeno na tom, že je optimalizačnímu algoritmu předložen dostatečný počet vzorů z trénovací množiny, na jehož základě je nalezeno optimální řešení dané úlohy. Celkový počet trénovacích vzorů, potřebný k optimálnímu naučení neuronové sítě, bývá však mnohem větší, než je velikost samotné trénovací množiny a vzory jsou proto optimalizačnímu algoritmu předkládány opakovaně. Jedna epocha pak označuje jeden průchod celou trénovací množinou. Typicky je neuronová síť učena po mnoho epoch.

Parametry neuronové sítě. Jako parametry jsou označovány takové proměnné neuronové sítě, jejichž ideální hodnoty jsou nalézány samotným procesem učení. Typickými zástupci parametrů neuronové sítě jsou její váhy a prahy.

Hyperparametry. Hyperparametry jsou proměnné, které přímo souvisejí s procesem učení a kapacitou neuronové sítě. Na rozdíl od parametrů neuronové sítě (viz výše) nelze však jejich hodnoty pomocí procesu učení vyvodit a je nutné je volit ještě před jeho započetím. Volba ideálních hyperparametrů je nejčastěji založena na empirickém bádání, intuici a na metodě pokusů a omylů. K ladění hyperparametrů slouží validační množina (viz dále). Typickým zástupcem hyperparametrů je například počet skrytých neuronů, počet skrytých vrstev neuronové sítě, způsob propojení neuronů, aktivační funkce neuronů, chybová funkce, volba optimalizačního algoritmu, způsob a míra regularizace, rychlost učení, velikost setrvačnosti, velikost dávky, inicializace parametrů a další.

Validační množina. Množina vzorů o stejném formátu jako množina trénovací, pomocí které lze v průběhu učení neuronové sítě odhadovat její schopnost generalizace a na základě které lze hledat ideální hodnoty hyperparametrů. Validační množina nesmí být použita, na rozdíl od množiny trénovací, k přímému učení neuronové sítě. Na místo toho jsou vzory z validační množiny na vstup neuronové sítě v pravidelných intervalech přikládány pouze při přerušení fáze učení tak, aby na jejich základě byla vyhodnocena schopnost modelu reagovat na data, která nejsou součástí trénovací množiny. Pro účel vzniku validační množiny je vždy použita část vzorů z trénovací množiny, která je natrvalo odejmuta, a obě množiny pak již zůstávají neměnné. Počet vzorů v trénovací a validační množině bývá běžně volen v poměru 8 ku 2 ve prospěch trénovací množiny. Na obrázku níže lze pozorovat, jak lze pomocí vyhodnocení přesnosti modelu na trénovací a validační množině určit míru jeho přeučení.



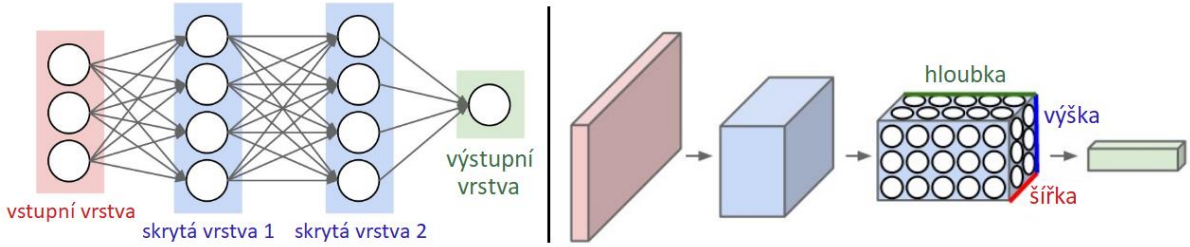
Obr. 22: Určení míry přeučení neuronové sítě pomocí porovnání její přesnosti na trénovací a validační množině (převzato z [36])

Testovací množina. Množina vzorů simulující data z praktického nasazení, vyčleněná pro určení konečné přesnosti neuronové sítě. Vzory v ní obsažené nesmí být součástí trénovací ani validační množiny a stejně jako vzory validační množiny nesmí být použity pro učení neuronové sítě. Testovací množina je v kontextu strojového učení vždy považována za velice vzácný zdroj dat, který by neměl být dotčen až do samotného konce procesu učení neuronové sítě. Nebylo-li by toto pravidlo dodrženo a byla-li by testovací množina použita podobně jako množina validační, hrozilo by, že by přítomné hyperparametry byly voleny tak, aby pracovaly dobře na testovací množině, kdežto na reálných datech by pak síť nepodávala očekávané výsledky. Se vzory testovací množiny proto musí být nakládáno velice obezřetně, aby nadále byly schopny simulovat taková data, se kterými se neuronová síť setká v praktickém nasazení.

2.2.4 KONVOLUČNÍ NEURONOVÁ SÍŤ

Konvoluční neuronové sítě jsou velmi podobné obyčejným dopředným neuronovým sítím, jimž se věnovaly předchozí kapitoly. Stále jsou tvořeny neurony, které obsahují váhy a prahy podléhající učení, jež jsou zakončeny nelineární aktivační funkcí. Výpočet výstupu neuronů se vzhledem k jejich propojení taktéž nemění a učení celé konvoluční neuronové sítě je rovněž řešeno metodou Stochastic Gradient Descent, založenou na optimalizaci chybové funkce. Všechny představené metody regularizace se navíc dají uplatnit i na konvolučních neuronových sítích [36].

Konvoluční neuronové sítě se od obyčejných dopředných neuronových sítí liší v tom, že na svém vstupu explicitně předpokládají data v podobě obrazů, respektive data uspořádaná do mřížky o libovolné hloubce, čemuž odpovídá i jejich architektura, a jsou proto v úloze zpracování obrazu velmi efektivní. Obzvláště nápadný je v porovnání s obyčejnou dopřednou neuronovou sítí způsob, jakým jsou organizovány neurony v samotných vrstvách konvoluční neuronové sítě. Vrstvy totiž již nejsou definovány pouze počtem neuronů v nich obsažených. Místo toho jsou vrstvy konvoluční sítě definovány jejich šířkou, výškou a hloubkou a jsou tak tvořeny trojrozměrnou soustavou předem uspořádaných neuronů (viz Obr. 23) [ibid.][36].



Obr. 23: Porovnání uspořádání neuronů obyčejné dopředné neuronové sítě (vlevo) s konvoluční neuronovou sítí (vpravo) – převzato z [36]

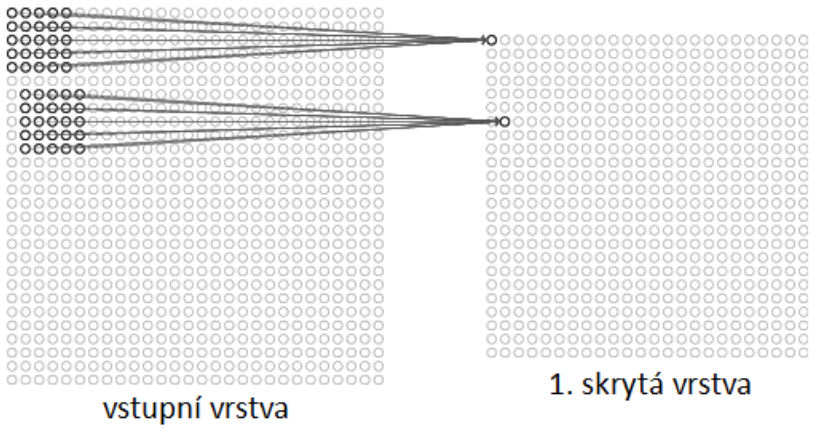
Dalšími stěžejními vlastnostmi konvolučních neuronových sítí jsou *lokální konektivita*, *sdílení parametrů* a takzvaný *pooling*. Každý z těchto pojmů bude postupně přiblížen v následujících podkapitolách.

2.2.4.1 LOKÁLNÍ KONEKTIVITA

Jednou z hlavních charakteristik konvoluční neuronové sítě je způsob, jakým jsou propojeny neurony mezi dvěma sousedními vrstvami. Namísto toho, aby neurony byly spojeny každý s každým, jako tomu bylo u vícevrstvých perceptronových sítí, jsou neurony propojeny jen na

lokální bázi podle hyperparametru známého pod názvem velikost filtru[6]. Velikost filtru je dána jeho šířkou a výškou, zpravidla je však pro oba tyto rozměry volena stejná hodnota a filtr tak nejčastěji tvoří čtverec. S filtrem je pak dále operováno stejně, jako je tomu u mechanismu klouzavého okénka s tím, že každá jedna pozice filtru odpovídá plnému propojení právě jednoho neuronu z následující vrstvy s neurony aktuální vrstvy, jejichž počet je v rámci plochy dán právě velikostí filtru (viz Obr. 24). Pokud je navíc aktuální vrstva, na které je filtr aplikován, tvořena soustavou neuronů o hloubce větší než jedna, je neuron následující vrstvy v rámci oblasti filtru plně propojen i se všemi neurony hlubších úrovní [39].

Propojení neuronů v rámci lokální konektivity lze shrnout i na následujícím příkladu. Je-li vstupní vrstva konvoluční neuronové sítě tvořena obrázkem o rozměrech 32x32x3 (obraz o 3 barevných kanálech) a velikost filtru je zvolena jako 5x5, pak každý neuron následující vrstvy bude se vstupní vrstvou propojen právě $5 * 5 * 3 = 75$ spojeními.



Obr. 24: Příklad propojení dvou neuronů 1. skryté vrstvy se vstupní vrstvou pomocí filtru o velikosti 5x5 (obě vrstvy mají pro zjednodušení hloubku o hodnotě 1) – převzato z [39]

Na obrázku výše si lze všimnout, že první skrytá vrstva je rozměrem menší než vrstva vstupní. Tato skutečnost je dána přirozenou vlastností mechanismu klouzavého okénka, neboť filtr o velikosti větší než jedna nelze posunovat po stejný počet kroků jako je počet neuronů v daném směru, aniž by nedošlo ke kolizi s okraji prostoru, nad kterým je filtr posouván. Stejné pravidlo platí i pro posouvání filtru o více než jeden *krok* (angl. stride), jehož hodnota tvoří další hyperparametr [39]. V praktických úlohách je však někdy plošná velikost (nikoliv hloubka) následující vrstvy uměle udržována na stejných hodnotách jako předchozí pomocí takzvaného *zero-padding*. Zero-padding, jak již název napovídá, spočívá v tom, že je aktuální vrstva sítě obehnána nulovými hodnotami tak, aby ji zvětšila natolik, že následující vrstva

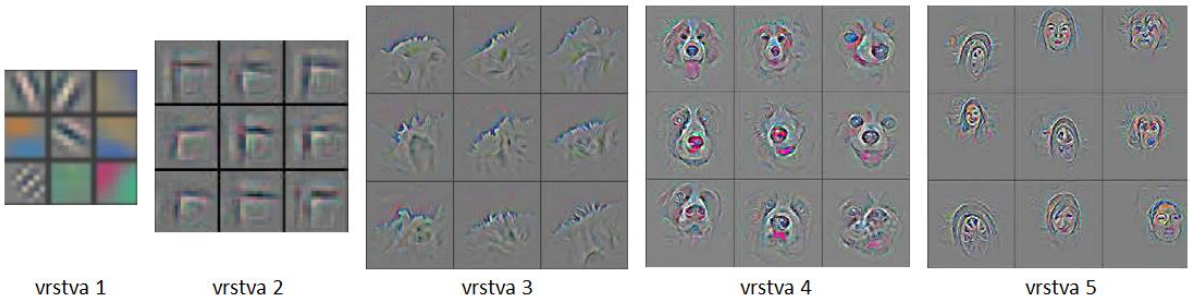
[6] V anglických textech je velikost filtru (filter size) známa také pod souslovím *receptive field* nebo *kernel size*

bude mít po aplikaci mechanismu klouzavého okénka požadované rozměry. Hodnota, která určuje počet těchto nulových ohraničení je taktéž hyperparametrem [36].

2.2.4.2 SDÍLENÍ PARAMETRŮ

Další klíčovou vlastností, kterou jsou konvoluční neuronové sítě charakteristické a díky které jsou ve zpracování obrazových dat tak efektivní, je sdílení parametrů mezi některými z jejich neuronů. Konkrétně jsou parametry sdíleny mezi všemi neurony obsaženými v každé jednotlivé hloubce jedné vrstvy neuronové sítě a počet unikátních parametrů v rámci jedné vrstvy je tak roven její hloubce. Jako příklad lze uvést, že na obrázku 23 vpravo je v kótované skryté vrstvě přítomno pouze pět jedinečných sad vah a prahů a na obrázku 24, který vyobrazuje skrytou vrstvu mající hloubku o velikosti jedna, všechny neurony sdílí stejné váhy a stejný, jediný práh. Hloubka vrstvy konvoluční neuronové sítě je v literatuře označována také jako *počet filtrů*, které vrstva obsahuje.

Každá sada takto sdílených parametrů slouží k detekci jednoho určitého příznaku ve vstupních obrazových datech. Tento příznak (angl. nazýván *feature*) odpovídá v prvních skrytých vrstvách sítě zpravidla různě orientovaným hranám nebo barevným shlukům, kdežto parametry hlubších vrstev konvoluční neuronové sítě jsou již zaměřeny na složitější příznaky jako kola automobilu, části obličeje a podobně (viz Obr. 25). Díky sdílení parametrů je pak síť tyto příznaky schopna detekovat na kterémkoliv místě ve vstupním obraze a nemusí tak pro každé jejich možné umístění podnikat žádná další opatření [36].

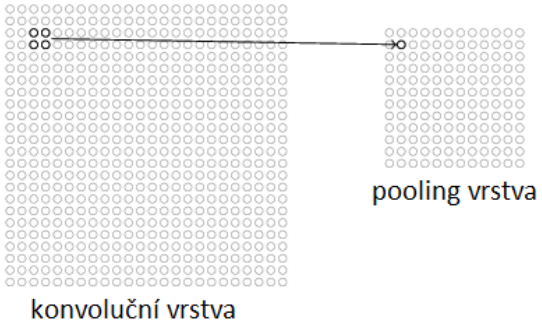


Obr. 25: Vizualizace náhodně vybraných příznaků napříč vrstvami plně naučené konvoluční neuronové sítě (převzato z [48])

Protože hloubka každé z vrstev konvoluční neuronové sítě přímo určuje, kolik různých lokálních příznaků dokáže v obraze rozpoznat, a s přibývajícími vrstvami roste i jejich zaměření (viz Obr. 25), je zvykem hloubku vrstev směrem od vstupní vrstvy k výstupní postupně navyšovat. V praxi se je proto možné setkat například s hloubkou vrstev navyšovanou od hodnoty 64, přes 128, 256 až po hloubku 512 [49].

2.2.4.3 POOLING VRSTVA

Konvoluční neuronová síť je typicky složena ze třech různých typů vrstev. Ta nejzákladnější vrstva, vrstva *konvoluční*, je založena na výpočtu aktivace neuronů na základě jejich lokálního propojení, jehož výsledek je přiložen na vstup aktivační funkce. Konvoluční vrstva byla popsána v předešlých kapitolách. Další druh vrstvy se vyskytuje na samém konci konvoluční sítě. Jedná se o vrstvu plně propojených neuronů v běžném rozložení v řadě a bez přítomnosti nelineární aktivační funkce, sloužící jako *výstupní* vrstva, která je typická pro všechny typy dopředných neuronových sítí. Poslední typ vrstvy, který je v konvolučních neuronových sítích běžně využíván, je *pooling* vrstva.

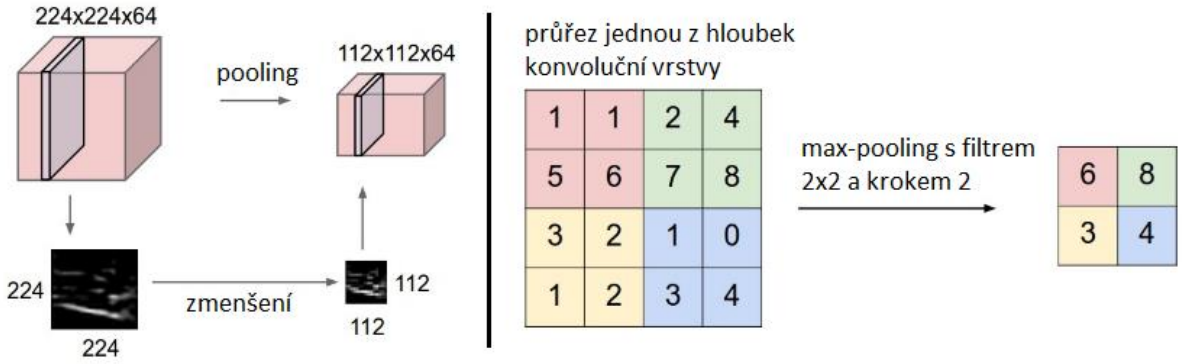


Obr. 26: Typické nastavení filtru (velikost 2x2 a krok o hodnotě 2) pooling vrstvy (převzato z [39])

Úloha pooling vrstvy spočívá v redukování množství informací z předešlých vrstev konvoluční neuronové sítě. Podobně jako konvoluční vrstva používá ke své funkci mechanismus klouzavého okénka, ovšem s krokem vždy vyšším než jedna, čímž redukuje množství neuronů na svém výstupu (viz Obr. 26). Na rozdíl od konvoluční vrstvy však pooling vrstva pro výpočet výstupní hodnoty používá mnohem jednodušší funkci, nejčastěji založenou na výběru maxima z výčtu hodnot daným aktuální pozicí filtru (viz Obr. 27 vpravo). Oproti konvoluční vrstvě pooling vrstva navíc pracuje s každou hloubkovou dimenzí vrstvy na vstupu zvlášť a pozice filtru tak na vstup pooling funkce, za předpokladu filtru o velikosti 2x2, předkládá pouze 4 hodnoty.

Způsob, jakým pooling vrstva redukuje počet neuronů vzhledem ke konvoluční vrstvě na jejím vstupu lze analogicky přirovnat ke zmenšování běžných digitálních obrazů pomocí redukce počtu jejich pixelů. Ve skutečnosti lze stejným způsobem zobrazit i pooling operaci, a to tak, že hodnoty vstupů a výstupů jsou převedeny na stupnici šedi, pomocí které je lze prezentovat jako obrazová data (viz Obr. 27 vlevo). Kromě skutečnosti, že s klesajícím počtem neuronů klesá i výpočetní náročnost operací napříč celou konvoluční neuronovou sítí, je

redukce počtu neuronů prospěšná i hlubším vrstvám sítě, které díky ní dokážou nahlížet při zachování stejné velikosti filtru na širší rozsah vstupního obrazu (viz Obr. 25).

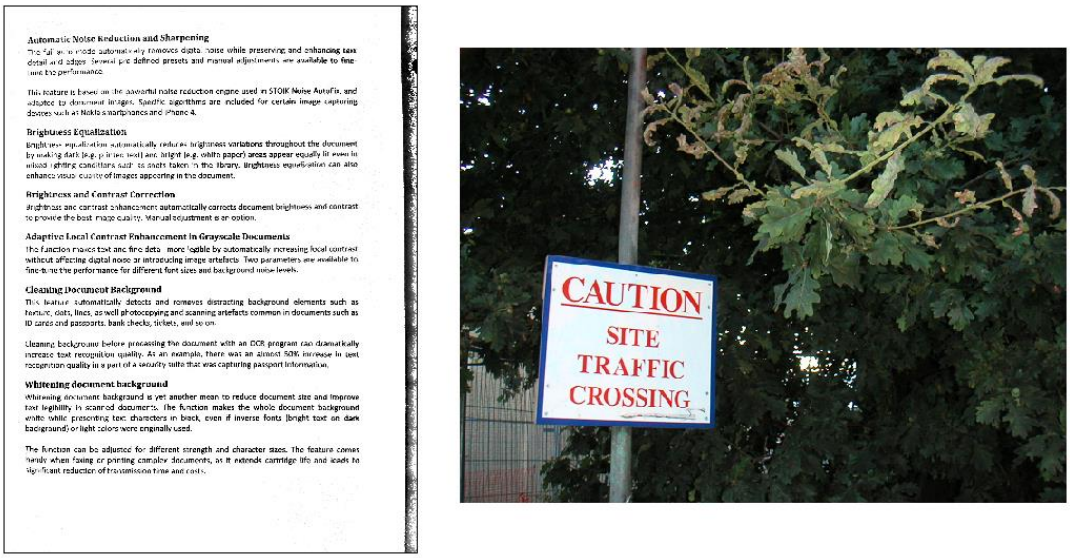


Obr. 27: Vizualizace funkce pooling vrstvy pomocí převedení výstupních hodnot neuronů na obrazová data (vlevo) a příklad operace max-pooling vrstvy (vpravo) – převzato z [36]

3 STAV POZNÁNÍ

Lokalizace a rozpoznávání textu v obrazových datech jsou v současnosti, díky neustálým pokrokům nejen v oblasti umělých neuronových sítí, stále aktivní a bádaná témata. Jakékoliv poznatky, objevy a přístupy k řešení s nimi spojené přitom mohly pro tuto práci mít nemalý přínos, a bude jim proto věnována tato kapitola.

Historicky byl výzkum věnován automatickému rozpoznávání textu v tištěných dokumentech, od kterého se postupem času pozornost přesunula na obdobnou problematiku v rámci obrazů reálných scén (viz Obr. 28). Právě ta byla inspirací i pro tuto práci. Přestože se totiž text, který se na osobním dokladu vyskytuje, vyznačuje pravidelnou strukturou, která budí dojem typického tištěného dokumentu, jeho čtení je vzhledem k vlastnostem snímku pořízeného fotoaparátem mobilního telefonu, ze kterého je text vyčítán, více spjato právě se zpracováním textu v obrazech reálných scén. Tuto příbuznost podporuje např. nepřesná ortogonalita a pokřivená perspektiva fotografie, nerovnoměrné osvětlení a odleskové plochy dokladu a v neposlední řadě rozmanité textury pozadí, na kterém je text vytištěný (viz kapitola 4). Méně použitelné se naopak jeví techniky používané pro zpracování snímků neskenovaných dokumentů, které často spoléhají na jednoduché rozložení textu a jednotné pozadí.



Obr. 28 Porovnání problematiky rozpoznávání texu v tištěných dokumentech (vlevo) a obrazech reálných scén (vpravo)

Protože je čtení textu z obrazových dat velmi komplexní problematika, bývá v literatuře rozdělena na lokalizační část, která je věnována nalezení a extrakci textu přítomného v obraze, a rozpoznávací část, věnující se samotnému vyčítání informace v něm

obsažené. Tohoto zavedeného rozdělení se budou držet i následující podkapitoly. V závěrečné podkapitole bude navíc pozornost věnována i vývoji v oblasti konvolučních neuronových sítí, které na sebe v současnosti díky jejich výkonům strhávají nebývalé množství pozornosti a jeví se tak jako slibná technologie i pro účely čtení identifikačních údajů z osobních dokladů.

3.1 Lokalizace textu v obrazech

Liu a Sarkar [50] hledají kandidáty obrazových oblastí s texty pomocí Niblackova algoritmu pro lokální adaptivní prahování [25]. Shlukování a filtraci spojených komponent pak provádějí na základě jejich geometrických vlastností, histogramů intenzity a tvarů. Liu et al. [51] používají pro adaptivní prahování klouzavé okénko o variabilních rozměrech, kterým lokalizují texty všech velikostí. Nalezené oblasti jsou pak filtrovány na základě vlastností spojených komponent a jsou následně seskupeny do slov použitím grafu, respektive hledáním maximálně kolineárních shluků v něm obsažených. Detekce textu pomocí shlukování na základě informace o barvě bylo použito v práci Kasar a Ramakrishnan [52]. Pro rozlišení textu od pozadí používají kombinaci vektorového stroje (z angl. support vector machine) a neuronové sítě, na jejichž vstupy přivádějí dvanáct různých příznaků dané spojené komponenty na základě její geometrie, okrajů, šířky tahu a hran.

Mnoho prací zaměřených na lokalizaci textu v obraze je založených na detektoru takzvaných maximálně stabilních extremálních oblastí (z angl. maximally stable extremal regions – MSER), který poprvé představili Matas et al. [53] jako metodu pro hledání shod mezi dvěma obrazy. Neumann a Matas [54] používají MSER v kombinaci s jejich topologickou informací pro tvorbu vícero hypotéz o řádcích textu, na základě kterých provádějí podrobné hledání sekvencí znaků následované krokem seskupování a validací na základě vektorového stroje. Učení vektorového stroje bylo založeno na osmi velikostně invariantních příznacích MSER ručně anotovaných trénovacích vzorů obrazů reálných scén z webového portálu Flickr. González et al. [55] kombinují MSER s lokálně adaptivním prahováním a rovněž používají klasifikátor znaků na základě vektorového stroje. Huang et al. [56] k rozlišení textu od pozadí kombinují detektor MSER s klasifikátorem v podobě konvoluční neuronové sítě, pomocí nichž jsou schopni rozdělit i takové MSER, jež odpovídají několika znakům textu naráz.

Epshtein et al. [22] představili pro lokalizaci textu operátor zvaný SWT (z angl. Stroke Width Transform), kterým hledají komponenty s konstantní šířkou tahu, pro text v obraze charakteristickou (viz kapitola 2.1.1.3). Mishra et al. [57] hledají znaky textu na základě HOG

deskriptoru [58] (z angl. histograms of gradient) a vektorového stroje uvnitř klouzavého okénka.

Gupta et al. [59] používají pro lokalizaci textu plně konvoluční regresní neuronovou síť inspirovanou technikou You Only Look Once (YOLO) [60] používanou pro detekci objektů v obraze. Obraz na vstupu je rozdělen mřížkou do mnoha buněk (např. po 16 pixelech), jež jsou asociovány s celkem sedmi hodnotami, které přímo určují pozici a pravděpodobnost výskytu potenciálního textu. Hodnoty jsou vypočteny sedmi lokálními predikátory, jež jsou připojeny na devíti vrstvou konvoluční neuronovou síť inspirovanou VGG architekturou [49]. Neuronová síť je trénována na syntetických datech.

Jaderberg et al. [61] naučili pro účely lokalizace textu konvoluční neuronovou síť, jež rozlišuje text od pozadí nad výřezy klouzavého okénka o rozměrech 24 x 24 pixelů. Vstupní obraz je zpracován v celkem 16 různých velikostech a výstupy konvoluční sítě pak tvoří takzvané salientní mapy (z angl. saliency map, viz Obr. 29), které pro každý pixel vstupního obrazu určují, s jakou pravděpodobností obsahuje text. Salientní mapy jsou následně použity pro určení ohraničení textu pomocí RLSA algoritmu (z angl. Run Length Smoothing Algorithm).

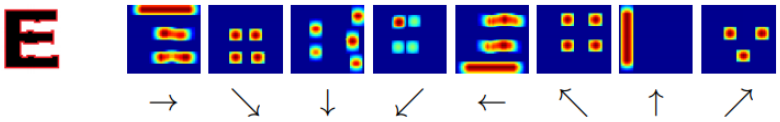
Obr. 29: Vstupní obraz (vlevo) a salientní mapa vytvořená konvoluční neuronovou sítí (vpravo) – převzato z [61]

3.2 Rozpoznávání textu v obrazech

Úloha rozpoznávání textu v obrazech přímo navazuje na jeho lokalizaci a práce, jež se jí zabývají, plně spoléhají na to, že obraz na vstupu již netvoří nic jiného, než text samotný. Termín rozpoznávání pak označuje proces vyčtení a vypsání textu z obrazu stejně, jako to při pohledu na obraz s textem umí člověk.

Neumann a Matas [54] rozpoznávají text po jednotlivých znacích pomocí vektorového stroje, na jehož vstup přikládají 200 různých příznaků založených na osmi směrovém chain-

code obvodových pixelů MSER daného znaku (viz Obr. 30). Rozlišení problematických malých a velkých písmen je následně řešeno pomocí typografického modelu zaměřeného na metriku dané řádky a jazykový model, využívající pro svou činnost pravděpodobnostní model Markovova řetězce 2. řádu, pak rozhoduje o konečné volbě nejpravděpodobnější hypotézy o zkoumaném textu. Použitý vektorový stroj byl naučen na 40 různých fontech, přičemž každé písmeno z podporované abecedy bylo využito k výpočtu příznaků bez dalších augmentací.



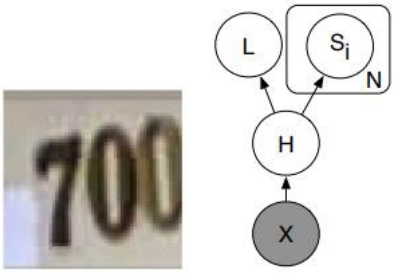
Obr. 30: Příznaky na bázi osmi směrového chain-code použité Neumannem a Matasem [54] pro rozpoznávání znaků (převzato z [ibid.])

Yao et al. [62] použili k rozpoznávání písmen příznaky z HOG deskriptoru spolu s takzvanými strokelety (z angl. strokelets), které popisují dílčí charakteristiku tvaru daného znaku. Příznaky z obou metod následně tvoří vstup pro klasifikátor typu Random Forest [63].

Wang et al. [64], Alsharif a Pineau [65], Coates et al. [66] a Saidane a Garcia [67] rozpoznávají text pomocí klasifikátoru znaků založeném na konvoluční neuronové síti, jež je schopna se příznaky, potřebné k rozeznání tříd jednotlivých znaků, naučit automatickou cestou přímo z obrazových dat. Wang et al. [64] pomocí konvoluční neuronové sítě rozeznávají celkem 62 tříd znaků, odpovídajících 26 velkým a 26 malým písmenům anglické abecedy a 10 číslicím. Tento klasifikátor, očekávající na svém vstupu obrazová data o rozměrech 32x32 pixelů, je pak metodou klouzavého okénka postupně aplikován na všechny možné pozice obrazu slova, které má být rozpoznáno. Velikost obrazu slova je vždy upravena tak, aby při zachování poměru stran jeho výška odpovídala právě 32 pixelům a výstup této operace tak tvoří matice o rozměrech Nx62, kde N odpovídá počtu všech pozic klouzavého okénka a 62 je počet rozpoznávaných znaků. Výsledná posloupnost znaků je pak vyhodnocena pomocí vyřazení takových sloupcových hodnot, které nejsou maximální (mechanismus známý pod angl. souslovím *non-maximal suppression*), a je porovnána se slovníkem, jež omezuje výstup jen na sadu podporovaných slov, která pak tvoří základ i pro konečné korekce výstupu celého systému.

Jinak se k této problematice postavili Goodfellow et al. [47], kteří pomocí konvoluční neuronové sítě vyčítají víceciferná domovní čísla v obrazech služby Google Street View bez jakékoliv předchozí segmentace a na rozdíl od předešlých prací tak nechávají síť rozpoznávat celou posloupnost znaků zcela autonomně. Výhradně dopředný model (viz Obr. 31) podporuje

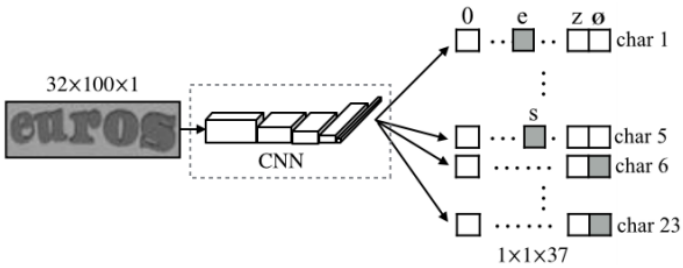
až pěticiferná domovní čísla ($N = 5$), jejichž skutečná délka je rozpoznávána paralelně se samotnými ciframi pomocí specializovaného klasifikátoru délky jejich sekvence, L, jež je součástí výstupní vrstvy neuronové sítě. Výstup neuronové sítě je tak kromě klasifikátoru délky sekvence, L, tvořen i pěti paralelně připojenými klasifikátory číslic, $S_1 \rightarrow S_5$, jež každý odpovídá jedné z pěti sekvenčních pozic vyčítaného čísla. Celý model je učen obvyklou metodou Stochastic Gradient Descent s učitelem s tím, že zpětná propagace chyby (mechanismus Back-Propagation) je prováděna pouze pro klasifikátor délky sekvence a ty klasifikátory čísel, které odpovídají délce sekvence v obrázku na vstupu. Pro výstupní neurony korespondující cifrám na pozicích, které nejsou v obraze přítomny (v případě Obr. 31 $S_4 \rightarrow S_5$), tak není zpětně propagována žádná chyba. Během experimentů bylo zjištěno, že čím je konvoluční neuronová síť hlubší, tím lepší přináší výsledky a konečný model, který se za použití pravděpodobnostních prahů (z angl. confidence thresholding), vedoucích k 95,64% pokrytí množiny všech dat, vyrovná rozpoznávací schopnosti člověka, se proto skládal z 11 skrytých vrstev. Model byl následně v téže práci rozšířen i pro rozpoznávání CAPTCHA sekvencí systému reCAPTCHA o délce až osmi znaků skládajících se z velkých a malých písmen anglické abecedy, v jejichž rozpoznávání dosáhl působivé přesnosti 99,8%.



Obr. 31: Příklad vstupního obrazu (vlevo) a zjednodušený model konvoluční neuronové sítě (vpravo) použitý v práci Goodfellow et al. [47] (převzato z [ibid.])

Stejný princip pro rozpoznávání textu v obrazech byl následně využit i v práci Jaderberg et al. [68] s tím, že model byl navržen tak, aby mohl rozpoznávat slova délky až 23 znaků složených z 26 písmen anglické abecedy a 10 číslic (model nerozeznává velká a malá písmena). Namísto klasifikátoru délky sekvence byl však představen takzvaný null znak, rozšiřující abecedu z celkových možných 36 znaků na 37, symbolizující, že se na dané pozici znak ve vstupním obraze již nevyskytuje. Učení konvoluční neuronové sítě tak není komplikováno podmíněným zpětným šířením chyby a text lze vynecháním všech null znaků na jejím výstupu vyčítat přímo. Použitá konvoluční neuronová síť je složena ze čtyř konvolučních vrstev s počty filtrů v pořadí od vstupu 64, 128, 256 a 512 a jedné plně

propojené vrstvy o 4096 neuronech, na jejíž výstup je paralelně napojeno všech 23 výstupních klasifikačních vrstev (viz Obr. 32).



Obr. 32: Konvoluční neuronová síť pro rozpoznávání jakékoliv kombinace znaků textu ve vstupním obraze použitá v práci Jaderberg et al. [68] (převzato z [ibid.])

Ve stejné práci Jaderberg et al. [68] navrhují i další přístup k rozpoznávání textu v obrazech bez nutnosti jeho segmentace na jednotlivé znaky, který spočívá ve vyčítání celých slov podle předdefinovaného slovníku. Konvoluční neuronová síť je tak omezena pouze na takovou sadu slov, která jí byla předložena při procesu učení. V práci je zároveň předvedeno, že se neuronová síť je tímto způsobem schopna naučit většinu používaných slov anglického jazyka, což je dokázáno použitím slovníku o velikosti 90 tisíc slov. Použitá architektura neuronové sítě je shodná s předešlou, jen její výstup tentokrát tvoří obvyklá jediná výstupní vrstva, v níž každý neuron odpovídá právě jednomu slovu. Pro účel učení byl však počet neuronů ve výstupní vrstvě navyšován postupně, protože jejich kompletní obsazení nevedlo vzhledem k omezeným paměťovým prostředkům a tím i omezené velikosti dávky (angl. batch size) ke konvergenci. Jejich počet byl proto navyšován po inkrementech o velikosti 5000 až do celkového počtu 20 tisíc neuronů, po nichž byl inkrement navýšen na 10 tisíc až do konečných 90 tisíc výstupních neuronů. Jak tento model, tak ten předešlý (rozpoznávání slova o max. 23 znacích), byl učen pomocí syntetických dat, imitujících text v obrazech reálných scén, pro něž byl sestrojen i patřičný generátor.

3.3 Vývoj konvolučních neuronových sítí

Vzhledem k předchozím kapitolám se zdá, že konvoluční neuronové sítě v úloze zpracování obrazu před ostatními přístupy vynikají a jsou tak v tomto oboru klíčem k dosažení těch nejlepších výsledků. Tuto domněnku potvrzují i evidované výsledky v rozpoznávání obrazů ručně psaných číslic v rámci MNIST databáze [69], pomocí které jsou různé metody vyčítání porovnávány již od roku 1998 a kategorie konvolučních neuronových sítí zde má před ostatními metodami viditelný náskok. Konvoluční neuronové sítě vedou žebříčky i dalších soutěží, jako například lokalizace a rozpoznávání objektů nad databází ImageNet [1], a je jim

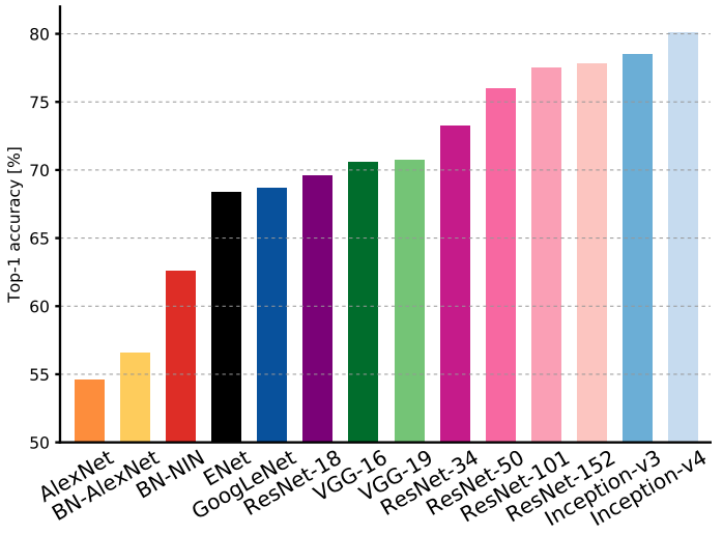
proto v poslední době věnována značná míra pozornosti, která vede k jejich neustálému vývoji.

Věří se, že za úspěchy konvolučních neuronových sítí stojí skutečnost, že jsou schopny se patřičné příznaky potřebné pro řešení dané úlohy naučit samostatně na základě trénovacích dat. Přístupy, které konvolučním neuronových sítím předcházely, byly totiž pokaždé založeny na ručně zvolených příznacích například typu SIFT, HOG, LBP nebo MSER, které byly až následně přiloženy na vstup klasifikátorů typu vícevrstvé perceptronové sítě, vektorového stroje (angl. Support Vector Machine) nebo náhodného lesa (angl. Random Forest [63]) [70].

Konvoluční neuronové sítě však ke svému učení potřebují, na rozdíl od ostatních metod založených na ruční volbě příznaků, obrovské množství trénovacích dat a výpočetního výkonu a tak bylo jejich použití od jejich prvních úspěšných aplikací (např. v práci LeCun et al. [71] z r. 1998) opožděno, protože ani jeden z těchto prostředků nebyl svého času dostupný. To se změnilo v roce 2012 s prací Krizhevsky et al. [38], ve které byla v rámci soutěže ILSVRC[7] [1], jež zpřístupnila množinu dat o velikosti 1,4 milionu anotovaných obrazů reálných scén, poprvé a s velkým úspěchem použita konvoluční neuronová síť. Krizhevsky et al. [38] tehdy doslova převálcovali konkurenci rozdílem v chybě celých 10,9% (15,3% vůči 26,2%) a odstartovali tak éru konvolučních neuronových sítí, která trvá dodnes.

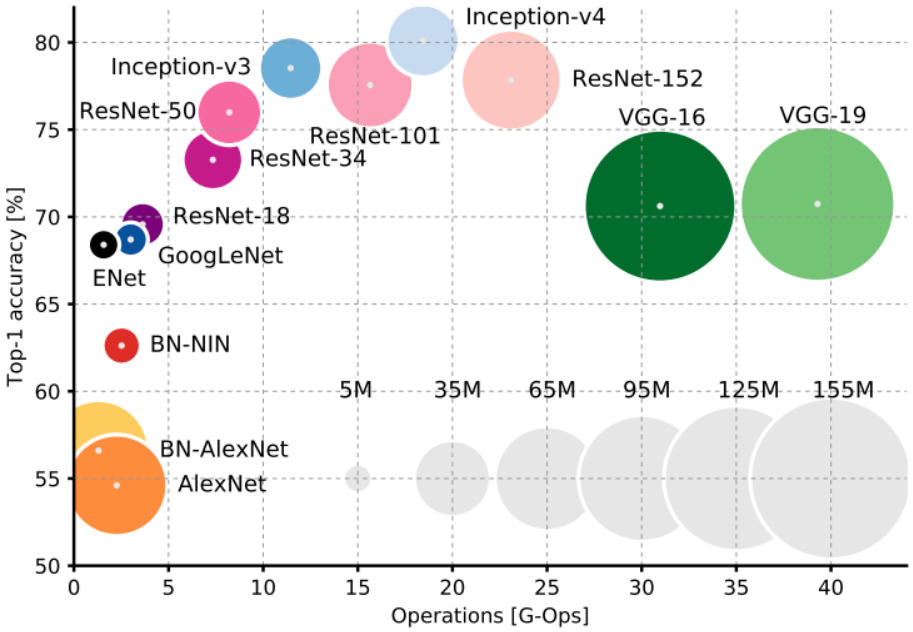
Canziani et al. [72] porovnávají na základě účasti v soutěži ILSVRC [1] všechny konvoluční neuronové sítě, které od roku 2012 nějakým způsobem posunuly stav poznání. Na obrázku 33 je proto možné pozorovat pokrok v přesnosti jednotlivých sítí sahajících od práce Krizhevsky et al. [38] z roku 2012 (na obr. označena jako AlexNet), přes GoogLeNet (Szegedy et al. [73]) a VGG (Simonyan a Zisserman [74]) z roku 2014, varianty ResNet (He et al. [75]) z roku 2015, až po Inception-v4 (Szegedy et al. [76]) z roku 2016.

[7] Z anglického sousloví ImageNet Large-Scale Visual Recognition Challenge



Obr. 33: Porovnání přesnosti rozpoznávání objektu v obraze různých architektur konvolučních neuronových sítí v rámci ILSVRC soutěže [1] (převzato z [72])

Ve stejné práci Canziani et al. [72] porovnávají modely konvolučních neuronových sítí i vzhledem k výpočetní náročnosti jejich dopředného průchodu (osa x) a počtu jejich parametrů (průměr kružnic), viz obrázek 34. V následujících podkapitolách budou některé ze zobrazených architektur krátce představeny.

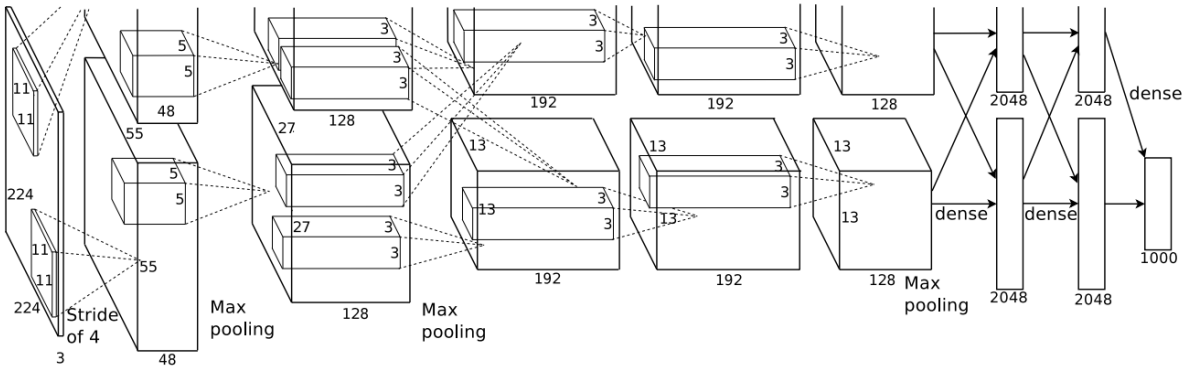


Obr. 34: Porovnání přesnosti, výpočetní náročnosti a počtu parametrů různých architektur konvolučních neuronových sítí v rámci ILSVRC soutěže [3] (převzato z [72])

3.3.1 AlexNet (2012)

Konvoluční neuronová síť představena v práci Alex Krizhevsky et al. [38], kterou bylo dokázáno, že při dostatečném množství trénovacích dat a s použitím grafických karet jako výpočetních jednotek při procesu učení, lze konvoluční neuronové sítě efektivně využívat k řešení úloh klasifikace obrazu.

Model je složen z osmi vrstev, z toho pět je konvolučních a tři jsou plně propojené. Mezi 1. a 2., 2. a 3. a 5. a 6. vrstvou je použita max-pooling vrstva s filtrem o velikosti 3x3 a krokem o hodnotě 2 a při každé z pozic klouzavého okénka tak dochází při pooling operaci k překryvu. První konvoluční vrstva obsahuje 96 filtrů o velikosti 11x11 s krokem o hodnotě 4, zatímco druhá vrstva je tvořena 256 filtry o velikosti 5x5 a krokem 1, jež zůstává napříč následujícími konvolučními vrstvami neměnný. Třetí a čtvrtá vrstva je tvořena 384 filtry o velikosti 3x3. Stejnou velikost filtru má i poslední, pátá konvoluční vrstva s 256 filtry. Výstup páté vrstvy je následně napojen na dvě po sobě jdoucí, plně propojené vrstvy o celkovém počtu 4096 neuronů, jejichž spojení podléhají dropout regularizaci. Výstupní plně propojená vrstva pak čítá 1000 neuronů, které odpovídají každé z rozpoznávaných tříd klasifikačního problému.



Obr. 35: Ilustrace modelu konvoluční neuronové sítě AlexNet, rozděleného vedví pro rozdělení učení mezi dva grafické procesory (převzato z [38])

Jako aktivační funkce je použita ReLU funkce, díky které síť konverguje několikanásobně rychleji, než při použití hyperbolického tangens. Celý model byl učen na dvou grafických kartách NVIDIA GTX 580 3GB po dobu šesti dnů.

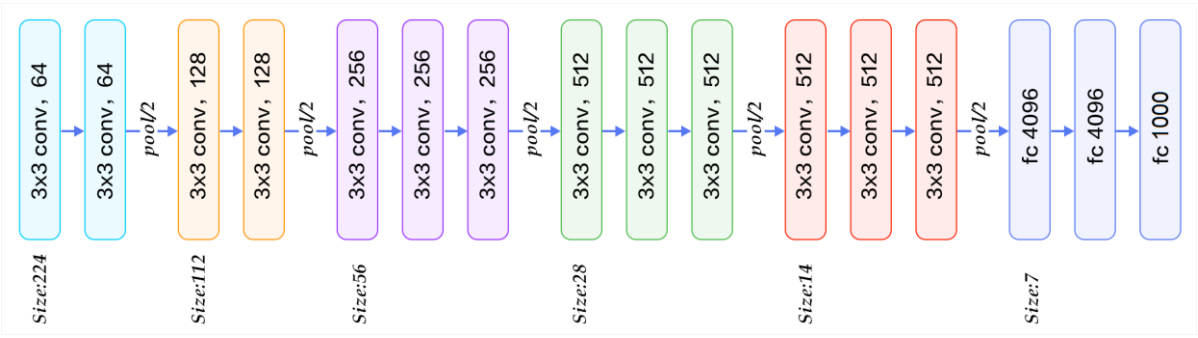
3.3.2 VGG (2014)

Simonyan a Zisserman [74], jež VGG konvoluční neuronové sítě představili, si dali za úkol prověřit, jaký vliv má hloubka modelu na jeho výslednou přesnost. Následkem toho byly

vytvořeny dvě konvoluční neuronové sítě o tehdy neslýchaných 16 a 19 vrstvách, jež se s úspěchem umístili na prvním a druhém místě lokalizace, respektive rozpoznávání objektů v obrazech tehdejšího ročníku soutěže ILSVRC [1].

Revoluční však obě VGG sítě nebyly jen pro svou hloubku, ale i pro jejich neobvykle jednoduchou architekturu, tvořenou výhradně filtry o velikosti 3x3 s krokem o hodnotě 1. Počet filtrů je napříč sítí postupně zvyšován o násobky dvou, 64 filtry začínaje a 512 filtry konče. Mezi konvolučními vrstvami, jež navyšují počet svých filtrů, jsou použity max-pooling vrstvy, snižující svou velikostí filtru 2x2 a krokem 2 rozlišení průchozích dat na polovinu. Použitím zero-padding-u je docíleno toho, že jiným způsobem se plošné rozlišení dat již nemění. Poslední tři plně propojené vrstvy VGG modelů zůstávají stejné, jako tomu bylo u AlexNet sítě, včetně využití dropout regularizace. Pro zabránění přeučení byla navíc použita i regularizace L2.

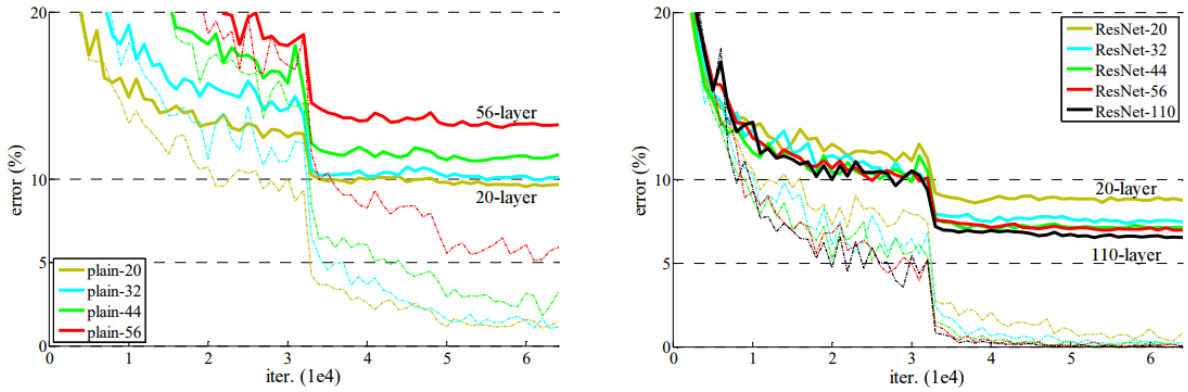
Proces učení byl založen na minimalizaci chybové funkce standardní metodou Stochastic Gradient Descent se zpětným šířením chyby a setrvačností a na čtyřech grafických kartách NVIDIA Titan Black 6GB trval v závislosti na modelu 2 až 3 týdny.



Obr. 36: Ilustrace modelu konvoluční neuronové sítě VGG-16 (převzato z [77])

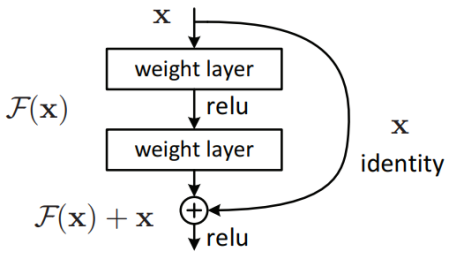
3.3.3 ResNet (2015)

V čase, kdy vznikal ResNet [75], bylo již vzhledem k předchozím úspěchům souvisejících s navyšováním počtu skrytých vrstev jasné, že budoucnost patří velmi hlubokým neuronovým sítím. He et al. [75] si proto na základě tohoto trendu dali za cíl prověřit, zda je opravdu možné současné modely jako VGG [74] jednoduše upravit přidáním dalších vrstev tak, aby podávaly lepší výsledky. Empirickými metodami bylo však následně zjištěno, že od určitého počtu skrytých vrstev se již výsledky konvoluční neuronové sítě dále nezlepšují, a s dalším přidáváním vrstev se naopak více a více zhoršují (viz Obr. 37 vlevo). Toto zhoršení přitom průkazně není způsobováno přeučením, protože se týká i trénovacích dat.



Obr. 37: Srovnání chyby při učení běžné konvoluční neuronové sítě (vlevo) a residuální sítě (vpravo) vzhledem k počtu jejich skrytých vrstev – plná křivka odpovídá validační množině, čerchovaná křivka trénovací množině (převzato z [75])

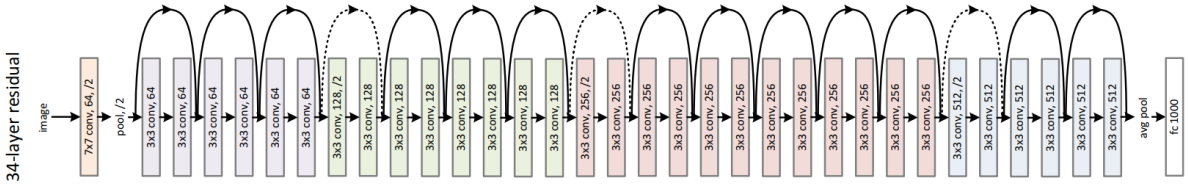
Na základě tohoto pozorování představují He et al. [75] nový koncept residuálních bloků pro hluboké neuronové sítě (viz Obr. 38), jež se snaží přiblížit původní teoretické myšlence zvyšování počtu jejich skrytých vrstev založené na tom, že pokud výkon n-vrstvé neuronové sítě dosahuje určité přesnosti, pak přidáním dalších vrstev by měla síť vykazovat přinejhorším stejné výsledky, jako před její úpravou. Neuronovou síť, jež vykazuje stejnou míru přesnosti jako její mělčí varianta, si lze představit tak, že všechny přidané vrstvy jsou pouhými identitami těch předchozích, čímž zákonitě musí tvrzení o jejich ekvivalentní přesnosti platit. Residuální blok proto zavádí do architektury neuronové sítě takzvané *identity shortcut* spojení, které do výpočtů dopředného šíření zavádí i hodnoty z předchozích vrstev a fakticky tak upravuje funkci v určitém bodě sítě z $y = \mathrm{F}(x)$ na $y = \mathrm{F}(x) + x$, přičemž x odpovídá výstupu některé z předešlých vrstev.



Obr. 38: Residuální blok představený v práci He et al. [75] (převzato z [ibid.])

Hypotéza, na které je residuální blok založen, je taková, že pokud by se identita blížila optimu, bude pro blok v případě vah a prahů blízkých nule, což je při použití L2 regularizace pravděpodobné, jednoduché potlačit $\mathrm{F}(x)$ část rovnice a ponechat tak za předpokladu ReLU aktivační funkce průchod hodnot rovným identitě z předchozích vrstev.

To, že residuální blok skutečně napomáhá k dosažení lepších výsledků v závislosti na počtu vrstev neuronové sítě He et al. [75] dokazují na pokusech s modely o 20, 32, 56, 110 (viz Obr. 37 vpravo), až o 1202 vrstvách, přičemž s modely o hloubce 101 a 152 vrstev vyhrávají hned několik významných soutěží. Nespornou výhodou ResNet modelu je navíc kromě jeho vysoké přesnosti i skutečnost, že je díky úvodní konvoluční vrstvě o kroku 2 následované pooling vrstvou (viz Obr. 39) výpočetně i paměťově o mnoho méně náročný než VGG model (viz Obr. 34). Jeho učení je přitom možno provést bez jakýchkoliv úprav standardní metodou Stochastic Gradient Descent se zpětnou propagací chyby.



Obr. 39: Ilustrace modelu residuální konvoluční neuronové sítě ResNet-34 (převzato z [75])

4 ANALÝZA PROBLEMATIKY ČTENÍ DOKLADŮ

Zatímco předchozí kapitoly byly věnovány tématům zpracování obrazu, lokalizaci a rozpoznávání textu v obrazech obecně, tato část práce bude již přímo zaměřena na analýzu problematiky čtení identifikačních údajů z osobních dokladů, a to tak, aby její výsledky mohly tvořit praktický základ pro návrh a implementaci připravovaného systému.

4.1 ANALÝZA OSOBNÍCH DOKLADŮ

První část analýzy je zaměřena na samotné osobní doklady. V následujících podkapitolách budou proto shrnuty veškeré informace, které byly o jednotlivých typech osobních dokladů nashromážděny, přičemž pozornost bude věnována pouze českým občanským průkazům a cestovním pasům, které jsou aktuálně v platnosti. Na ostatní typy osobních dokladů, jako je například řidičský průkaz nebo cizokrajné doklady, se tato práce nezaměřuje.

K tomu, aby byla analýza dané problematiky a později i implementace celého systému proveditelná, bylo nutné pracovat s poměrně velkým počtem vzorových fotografií dokladů. Množina vzorů, s níž je napříč touto prací operováno, proto čítá přes 550 fotografií občanských průkazů a cestovních pasů pořízených jak pomocí fotoaparátu, tak pomocí stolního skeneru. Kvalita fotografií přitom osciluje od perfektně ostrých snímků až po ty natolik rozmazané, že doklad, který je na nich zachycen, není čitelný ani člověkem a množina je tak velmi pestrá.

Veškeré obrázky osobních dokladů reálných osob, které budou v následujících kapitolách této práce zobrazeny, budou v rámci ochrany osobních údajů důsledně anonymizovány tak, aby na základě údajů v nich obsažených, nebylo možné přímo ani nepřímo identifikovat vlastníka daného dokladu.

4.1.1 OBČANSKÝ PRŮKAZ STARŠÍHO TYPU

Občanský průkaz staršího typu (viz Obr. 40), vydávaný od roku 2005 do roku 2011 včetně, sice již pomalu mizí z oběhu, ale protože je standardní platnost občanských průkazů stanovena na 10 let [78], jeho poslední exempláře budou platné ještě v roce 2021. Průkaz je vyroben ve formě laminované karty o rozměrech 105 mm na 74 mm, jež odpovídají mezinárodnímu formátu identifikačních karet ID2, vyjádřeného normou ISO/IEC 7810 [79].

Obr. 40: Občanský průkaz staršího typu – vlevo přední strana, vpravo zadní strana (převzato z [80])

Přední strana občanského průkazu obsahuje základní údaje o držiteli (jméno, příjmení, rodné číslo, datum narození, pohlaví, státní občanství), podobu a podpis držitele, dobu platnosti a číslo dokladu a strojově čitelné údaje.

Zadní strana obsahuje další informace o držiteli (místo narození, trvalý pobyt, rodné příjmení) volitelně s doplňujícími, nepovinnými údaji (rodinný stav, titul, informace o dětech a o manželovi / manželce či partnerovi), datum vydání dokladu a označení úřadu, který jej vydal. Názvy položek na přední straně a některé názvy položek na zadní straně jsou vyobrazeny v českém, anglickém a francouzském jazyce.

Pole příjmení, rodné příjmení, pohlaví, místo narození a trvalý pobyt jsou vytištěna velkými písmeny, s výjimkou zkratek „okr.“, označující v místě narození a trvalém pobytu okres, a „č.p.“, která je v trvalém pobytu příležitostně použita k označení čísla popisného. Výhradně velkými písmeny je vytištěno i označení města úřadu, který doklad vydal. Velkými i malými písmeny je vytištěno jméno, případný titul držitele a typ úřadu, který doklad vydal. Výhradně malá písmena jsou využita pro označení rodinného stavu držitele.

V případě, že se držitel občanského průkazu narodil na území České republiky, je pole místo narození složeno ze dvou řádek tak, jak je tomu na obrázku 40. V opačném případě tvoří místo narození pouze jeden řádek s označením země, ve které se držitel narodil, a to v třípísmenném formátu definovaném standardem ISO 3166 [81]. Trvalý pobyt je tvořen dvěma až třemi řádky.

Písmo, které je napříč občanským průkazem použito a jež není zákonem nijak definováno, se nepodařilo identifikovat[8] a je tedy patrně proprietárního charakteru. Z veřejně dostupných písem jsou mu nejvíce podobná „Quitador Sans“ (kromě znaků „9“ a „b“), „Calibri Light“ (kromě znaků „1“ a „9“) a „Avenir-Light“. Strojově čitelná oblast je vytištěna písmem „OCR-B“, upravovaným mezinárodní normou ISO 1073-2 [82].

Strojově čitelná oblast v dolní části přední strany dokladu je typu TD2, jehož strukturu popisuje dokument ICAO 9303-6 [83]. Je složena ze dvou řádek o 36 znacích, jež v sobě nesou informace o typu, datu platnosti a číslu dokladu, o zemi, která doklad vydala a o jméně, příjmení, státním občanství, datu narození, pohlaví a rodném čísle držitele (viz Obr. 41). Jako oddělovač mezi některými údaji je použit znak „menší než“ (<) a kompletní abeceda strojově čitelné oblasti je tak tvořena 37 znaky složených z 10 číslic, 26 písmen anglické abecedy a jednoho oddělovače. Jméno a příjmení držitele je zaznamenáno bez diakritiky. Všechny údaje kromě typu dokladu, jména a příjmení držitele a údaje o zemi, která doklad vydala, jsou opatřeny i kontrolní číslicí pro ověření jejich správného vyčtení[9].

Obr. 41: Údaje ve strojově čitelné oblasti občanského průkazu staršího typu

Další charakteristikou občanského průkazu staršího typu je jeho pozadí, jehož poloha není vůči textu zcela stabilizována a každý průkaz má tak obraz na pozadí různě odchýlený. Odchylka je přítomna pravděpodobně pouze ve formě posunutí obrazu. Součástí pozadí je i vyobrazení lipového stromu, které je umístěno pod údajem o pohlaví držitele, pomocí něhož lze odchylku v pozadí pozorovat nejlépe. Na obrázku 42 vlevo jsou porovnány dva různé průkazy s výrazně odlišnými odchylkami.

Součástí občanského průkazu je rovněž mnoho ochranných prvků, které zabraňují jeho padělání. Jeden z takových prvků je i hologram umístěný na pravém okraji podobizny držitele, jež z části zasahuje i do přilehlého bloku textu a může tak svou přítomností při pořizování fotografie dokladu vytvářet odlesky nebo text zčásti překrývat (viz Obr. 42 vpravo). Výraznost

[8] K určování písem byla použita služba Identifont, dostupná na adrese http://www.identifont.com
[9] Postup, jak provést ověření pomocí kontrolní číslice je uveden v dokumentu ICAO 9303-3 [83]

hologramu na výsledné fotografii je závislá na úhlu objektivu fotoaparátu a na okolním osvětlení.

Obr. 42: Různé posunutí pozadí mezi průkazy (vlevo) a rušivý vliv ochranného hologramu (vpravo)

4.1.2 OBČANSKÝ PRŮKAZ NOVÉHO TYPU

Občanský průkaz nového typu (viz Obr. 43), běžně platný po dobu 10 let, je vydáván od roku 2012 až po současnost. Průkaz je vyroben ve formě plastové karty o rozměrech 85,6 mm na 53,98 mm, jež odpovídají mezinárodnímu formátu identifikačních karet ID1, vyjádřeného normou ISO/IEC 7810 [79].

Obr. 43: Občanský průkaz nového typu – vlevo přední strana, vpravo zadní strana (převzato z [80])

Přední strana občanského průkazu obsahuje základní údaje o držiteli (jméno, příjmení, datum a místo narození, pohlaví, státní občanství), podobu a podpis držitele a dobu platnosti, datum vydání a číslo dokladu.

Zadní strana obsahuje další informace o držiteli (trvalý pobyt, rodné číslo) volitelně s doplňujícími, nepovinnými údaji (rodinný stav, titul), označení úřadu, který jej vydal,

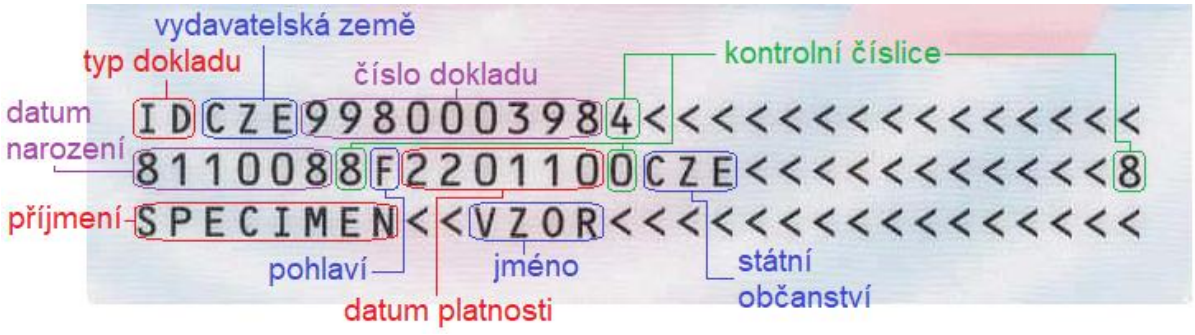
a strojově čitelné údaje. Na rozdíl od starší verze průkazu nelze již na nový občanský průkaz zapsat údaje o dětech a o manželovi / manželce či partnerovi. Názvy položek na přední straně a některé názvy položek na zadní straně jsou vyobrazeny v českém a anglickém jazyce.

Použití velkých a malých písmen je shodné s občanským průkazem staršího typu s výjimkou údaje o jméně držitele, který je nyní vytištěn, stejně jako příjmení, výhradně velkými písmeny.

Pro pole místo narození a trvalý pobyt platí stejná pravidla, jako tomu bylo u starší verze průkazu a nabývají tak 1-2, respektive 2-3 řádek textu.

Písmo, které je napříč občanským průkazem použito a jež není zákonem nijak definováno, se nepodařilo identifikovat[10] a je tedy patrně proprietárního charakteru. Z veřejně dostupných písem je mu nejvíce podobné „Lucida Sans Unicode“. Strojově čitelná oblast je vytištěna písmem „OCR-B“, upravovaným mezinárodní normou ISO 1073-2 [82].

Strojově čitelná oblast v dolní části zadní strany dokladu je typu TD1, jehož strukturu popisuje dokument ICAO 9303-5 [83]. Je složena ze třech řádek o 30 znacích, jež v sobě nesou informace o typu, datu platnosti a číslu dokladu, o zemi, která doklad vydala a o jméně, příjmení, státním občanství, datu narození a pohlaví držitele (viz Obr. 44). Na rozdíl od občanského průkazu staršího typu nelze ze strojově čitelné oblasti odvodit rodné číslo držitele. Abeceda strojově čitelné zóny zůstává stejná a jméno a příjmení držitele je tak nadále zaznamenáno bez diakritiky. Všechny údaje kromě typu dokladu, jména a příjmení držitele a údaje o zemi, která doklad vydala, jsou opatřeny i kontrolní číslicí pro ověření jejich správného vyčtení[11].



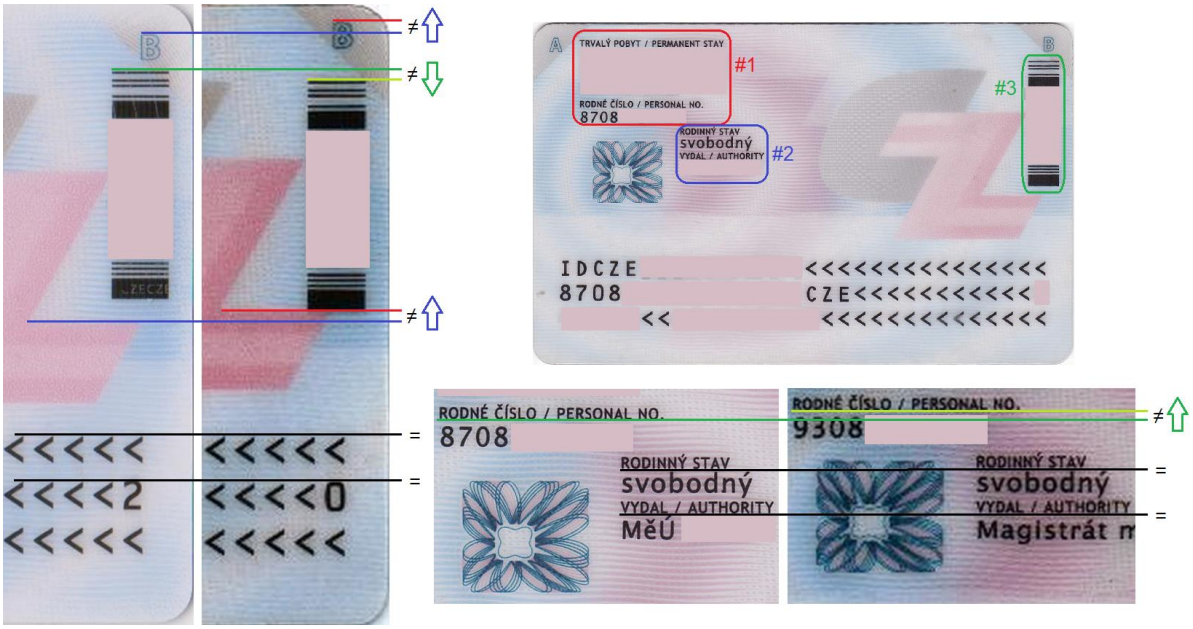
Obr. 44: Údaje ve strojově čitelné oblasti občanského průkazu nového typu

Kromě strojově čitelné oblasti je průkaz na zadní straně opatřen ještě dvourozměrným čárovým kódem typu PDF417 [85] a volitelně i elektronickým kontaktním čipem, jež v sobě

[10] K určování písem byla použita služba Identifont, dostupná na adrese http://www.identifont.com
[11] Postup, jak provést ověření pomocí kontrolní číslice je uveden v dokumentu ICAO 9303-3 [83]

nesou údaj o číslu dokladu. Do elektronického čipu lze navíc zapsat data pro vytváření, ověřování a užívání elektronických podpisů [86].

Zatímco obraz na pozadí přední strany dokladu je napříč všemi zkoumanými vzory perfektně stabilní, poloha pozadí zadní strany dokladu není vůči kartě stabilizována (viz červené a modré vodící čáry na Obr. 45 vlevo) a každý průkaz má tak obraz na pozadí různě odchýlený, stejně jako tomu je u občanského průkazu staršího typu. Na rozdíl od své starší varianty však na zadní straně nového průkazu není pevně dána ani pozice ostatních jeho prvků (viz zelené vodící čáry na Obr. 45 vlevo a vpravo dole) a jedinou stabilní oblastí tak zůstává strojově čitelná zóna v jeho spodní části, jejíž pozice podléhá striktním pravidlům dle dokumentu ICAO 9303 [83]. Po dalším zkoumání bylo zjištěno, že prvky v popředí jsou navíc na kartě průkazu sdruženy do třech různých bloků (viz Obr. 45 vpravo nahoře), jejichž pozice nejsou vzájemně závislé a mohou tak napříč doklady vůči sobě různě oscilovat. V rámci jednoho bloku je již relativní pozice prvků vůči sobě stabilní a napříč průkazy se nemění. Tyto bloky lze proto přirovnat k otiskům třech různých inkoustových razítek, jež byla při kompletaci průkazu aplikována nezávisle.



Obr. 45: Různé posunutí pozadí a popředí mezi průkazy (vlevo a vpravo dole) a uspořádání prvků popředí do bloků (vpravo nahoře)

Stejně jako tomu bylo u starší varianty dokladu, je na novém občanském průkazu přítomen hologram, umístěný na pravém okraji podobizny držitele, jež z části zasahuje i do přilehlého bloku textu a může tak svou přítomností při pořizování fotografie dokladu vytvářet odlesky nebo text zčásti překrývat (viz Obr. 46). Hologramy jsou vyhotoveny ve dvou různých verzích, přičemž první verze, vydávaná od prvních průkazů z roku 2012 do dubna roku 2014,

zasahuje výškově do menší oblasti, než je tomu u jeho druhé verze, která tu první nahradila od května roku 2014. Výraznost hologramu na výsledné fotografii je závislá na úhlu objektivu fotoaparátu a na okolním osvětlení.

Obr. 46: Rušivý vliv prvotní verze (vlevo a uprostřed) a stávající verze ochranného hologramu (vpravo)

4.1.3 CESTOVNÍ PAS

Cestovní pas na obrázku 47, běžně platný po dobu 10 let, je vydáván od roku 2006 až po současnost. Průkaz je vyroben ve formě knížky, v rámci níž se identifikační údaje nacházejí na datové stránce v podobě polykarbonátové karty o rozměrech 125 mm na 88 mm, jež odpovídají mezinárodnímu formátu identifikačních karet ID3, vyjádřeného normou ISO/IEC 7810 [79].

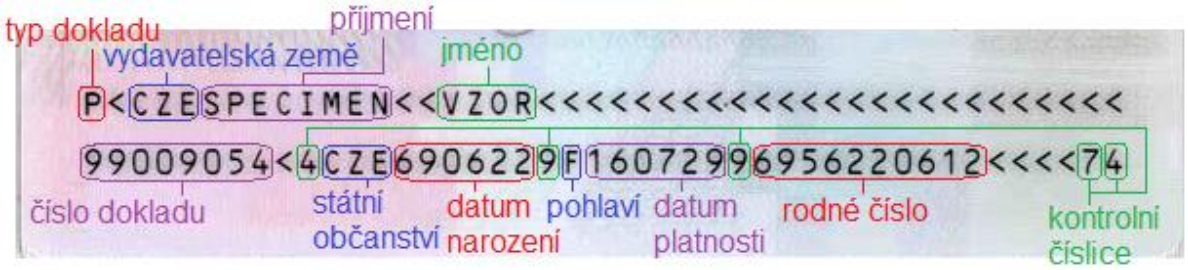
Obr. 47: Cestovní pas (převzato z [84])

Cestovní pas obsahuje základní údaje o držiteli (jméno, příjmení, datum a místo narození, rodné číslo, pohlaví, státní občanství), podobu a podpis držitele, dobu platnosti,

datum vydání, typ a číslo dokladu, označení země a úřadu, který jej vydal, a strojově čitelné údaje. Názvy položek jsou vyobrazeny v českém, anglickém a francouzském jazyce.

Použití velkých a malých písmen je shodné s občanským průkazem nového typu. Písmo, které je napříč cestovním pasem použito a jež není zákonem nijak definováno, se nepodařilo identifikovat[12] a je tedy patrně proprietárního charakteru. Z veřejně dostupných písem je mu nejvíce podobné „URW Classico“. Strojově čitelná oblast je vytištěna písmem „OCR-B“, upravovaným mezinárodní normou ISO 1073-2 [82].

Strojově čitelná oblast v dolní části dokladu je typu TD3, jehož strukturu popisuje dokument ICAO 9303-4 [83]. Je složena ze dvou řádek o 44 znacích, jež v sobě nesou informace o typu, datu platnosti a číslu dokladu, o zemi, která doklad vydala a o jméně, příjmení, státním občanství, datu narození, pohlaví a rodném čísle držitele (viz Obr. 48). Abeceda strojově čitelné zóny je shodná s občanskými průkazy a jméno a příjmení držitele je tak zaznamenáno bez diakritiky. Všechny údaje kromě typu dokladu, jména a příjmení držitele a údaje o zemi, která doklad vydala, jsou opatřeny i kontrolní číslicí pro ověření jejich správného vyčtení[13].



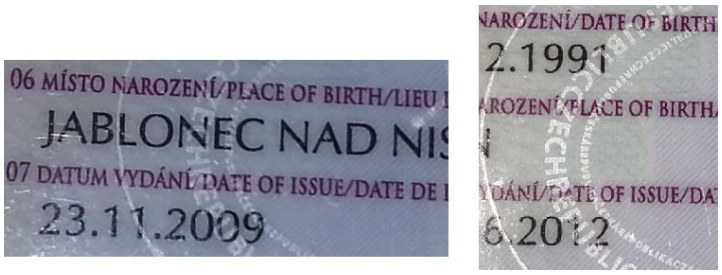
Obr. 48: Údaje ve strojově čitelné oblasti cestovního pasu

Kromě strojově čitelné oblasti je cestovní pas opatřen ještě nosičem dat v podobě elektronického bezkontaktního čipu, jež v sobě nese údaje o zobrazení obličeje a o otiscích prstů rukou držitele a údaje uvedené na datové stránce cestovního dokladu [87].

Poloha pozadí cestovního pasu není vůči kartě zcela stabilizována a každý průkaz má tak obraz na pozadí různě odchýlený, podobně jako tomu je u občanských průkazů. Poloha prvků v popředí však již mezi různými doklady zůstává neměnná.

Ochranné prvky, které by při čtení identifikačních údajů mohly působit rušivě, lze pozorovat na obrázku 49. Výraznost hologramu na výsledné fotografii je závislá na úhlu objektivu fotoaparátu, na použití blesku fotoaparátu a na okolním osvětlení.

[12] K určování písem byla použita služba Identifont, dostupná na adrese http://www.identifont.com
[13] Postup, jak provést ověření pomocí kontrolní číslice je uveden v dokumentu ICAO 9303-3 [83]



Obr. 49: Rušivý vliv ochranných prvků cestovního pasu

4.2 PROJEKTY S PODOBNOU TÉMATIKOU

Další část analýzy je zaměřena na analýzu již existujících projektů zabývajících se podobnou tématikou jako tato práce. Pozornost bude proto věnována takovým projektům, které rozpoznávají identifikační údaje z osobních dokladů na základě jejich optického záznamu pořízeného pomocí fotoaparátu mobilního telefonu. Předmětem zkoumání bude zejména spolehlivost rozpoznávání a rozsah rozpoznávaných identifikačních údajů, a zda stávající řešení podporují plný rozsah abecedy znaků, které se na českých dokladech vyskytují (např. písmena s diakritickými znaménky). Na projekty, které pro svou funkci vyžadují dedikovaný hardware, ať už stacionárního nebo mobilního charakteru, se tato kapitola nezaměřuje.

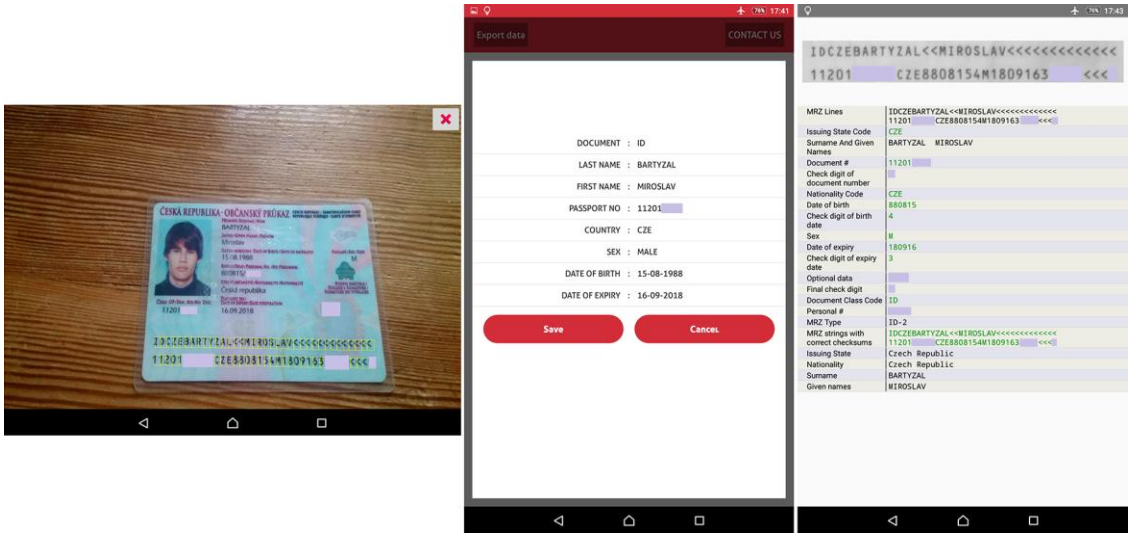
4.2.1 PROJEKTY ZAMĚŘENÉ NA STROJOVĚ ČITELNOU OBLAST

Mnoho projektů, které se zabývají čtením identifikačních údajů z osobních dokladů, je založeno výhradně na zpracování jejich strojově čitelné oblasti (z angl. Machine Readable Zone). Motivace, která se za tímto úzkým zaměřením skrývá, spočívá v tom, že prvky strojově čitelné oblasti jsou v obraze svou pravidelnou strukturou lehce rozeznatelné a jejich následné rozpoznávání může být díky jejich vlastnostem, které byly speciálně navrženy pro strojové zpracování, velice spolehlivé. K vlastnostem, které dosahování spolehlivých výsledků značně napomáhají, patří například velké rozestupy mezi jednotlivými znaky, omezená abeceda znaků, použité písmo „OCR-B“ [82], které bylo speciálně navrženo tak, aby od sebe jednotlivé znaky byly co nejvíce odlišitelné a přítomnost kontrolních čísel, pomocí nichž je možné správnost vyčtených údajů ověřit.

Všechny testované projekty, k nimž se řadí aplikace MRZ Recognition [88], Accura Scan [89] a Regula Document Reader [90], jsou volně dostupné pro mobilní zařízení jak se systémy Android, tak se systémy iOS. Rozpoznávání identifikačních údajů probíhá přímo na samotném mobilním zařízení a ke svému chodu tak ani jedno z řešení nepotřebuje připojení k internetu. Zpracování strojově čitelné zóny je ve všech případech velmi rychlé a celý proces nezabere více než dvě vteřiny. Přesnost rozpoznávání je takřka bezchybná, z čehož se dá

usuzovat, že je pravděpodobně výsledek zobrazen až tehdy, když je s úspěchem provedeno ověření údajů pomocí kontrolních číslic. Celkově je práce se všemi třemi aplikacemi intuitivní a to, k čemu jsou určeny, vykonávají velmi dobře a nedá se jim proto minimálně z technického hlediska prakticky nic vytknout.

Nevýhody těchto projektů však pramenní z jejich zaměření na strojově čitelnou oblast, která sama o sobě obsahuje jen zlomek informací, které se na osobním dokladu ve skutečnosti nacházejí. V údajích o jméně a příjmení držitele tak chybí diakritika, zatímco ostatní údaje, jako je místo narození, trvalý pobyt, podoba, podpis, u občanského průkazu nového typu i rodné číslo držitele, označení úřadu, který doklad vydal a další, chybí úplně. Některá jména a příjmení, která v sobě obsahují například přehlásku, jsou navíc ve strojově čitelné zóně dle převodní tabulky dokumentu ICAO 9303 [83] zaznamenány v alternativní podobě, která může následně vést ke zkomolení skutečného jména držitele. Například příjmení Hüber je ve strojově čitelné oblasti zaznamenáno jako HUEBER a nejsou-li dostupné další informace o držiteli, nelze rozeznat, zda došlo k záměně přehlásky, nebo se skutečně jedná o osobu s příjmením Hueber (obě varianty jsou platná příjmení).



Obr. 50: Snímání dokladu aplikací MRZ Recognition [88] (vlevo) a zobrazení výsledků v aplikaci Accura Scan [89] (uprostřed) a Regula Document Reader [90] (vpravo)

Z výše uvedeného vyplývá, že na získávání identifikačních údajů ze strojově čitelné zóny není vzhledem ke kvalitě existujících řešení již co vylepšovat, avšak pokud má být vyčítání informací kompletní, měla by být pozornost zaměřena i na oblasti dokladu mimo jeho strojově čitelné údaje.

4.2.2 PROJEKTY VYČÍTAJÍCÍ ÚDAJE MIMO STROJOVĚ ČITELNOU OBLAST

Další skupinou projektů, jejichž analýza je pro tuto práci důležitá, jsou projekty, které se při rozpoznávání identifikačních údajů zabývají informacemi i mimo strojově čitelnou oblast dokladu.

Do této kategorie patří řada komerčních produktů jako je IDscan Mobile App [91] od společnosti IDscan Biometrics, AcuFill [92] od společnosti Acuant, Netverify [93] od společnosti Jumio nebo Smart ID Reader [94] od společnosti Smart Engines. Všechny tyto produkty jsou však zaměřeny výhradně na komerční sféru, a zatímco demo aplikace IDscan není pro jednotlivce k dostání vůbec, ukázkové aplikace společností Acuant [95] a Jumio [96], které jsou pro Android volně k nalezení v rámci vyhledávání služby Google Play, nebyly v době testování na ani jednom z použitých zařízení funkční[14]. Díky spolupráci se společností GoPay, která se společností Jumio navázala kontakt a plánuje aktivně využívat jejích služeb, bylo však z těchto třech řešení možné zanalyzovat alespoň produkt Netverify. Dále je do analýzy zahrnut i produkt Smart ID Reader, který je již ve své ukázkové verzi [97] plně funkční.

Společnosti IDscan Biometrics a Acuant byly pro nedostupnost jejich ukázkových aplikací kontaktovány alespoň přes online formulář na jejich webových stránkách. Na zprávu, ve které byly dotazovány schopnosti jejich řešení s ohledem na české osobní doklady, však nereagovaly.

4.2.2.1 NETVERIFY

Řešení od společnosti Jumio [93] nabízí mimo čtení identifikačních údajů z osobních dokladů i další služby, jako je ověření, zda snímaný doklad není padělkem, a ověření identity držitele porovnáním jeho obličeje s podobiznou vyobrazenou na dokladu. Celý tento systém je zaštítěn nejen umělou inteligencí, ale i lidským faktorem a lze od něj proto očekávat přesvědčivé výsledky, což se potvrzuje minimálně na takřka bezchybném rozpoznávání identifikačních údajů. Rozsah těchto údajů je však pro občanské průkazy českého typu velmi omezen a prakticky odpovídá pouze informacím uchovaným v jejich strojově čitelné oblasti. Ve vyčtených datech proto chybí diakritika a některé další cenné informace, jako například trvalý pobyt držitele. Po úmyslné manipulaci s fotografií dokladu a záměně například jména držitele ve strojově čitelné zóně však navrácené výsledky obsahovaly jméno z oblasti mimo strojově čitelnou zónu a vzhledem k tomu, že do rozpoznávání identifikačních údajů může zasáhnout

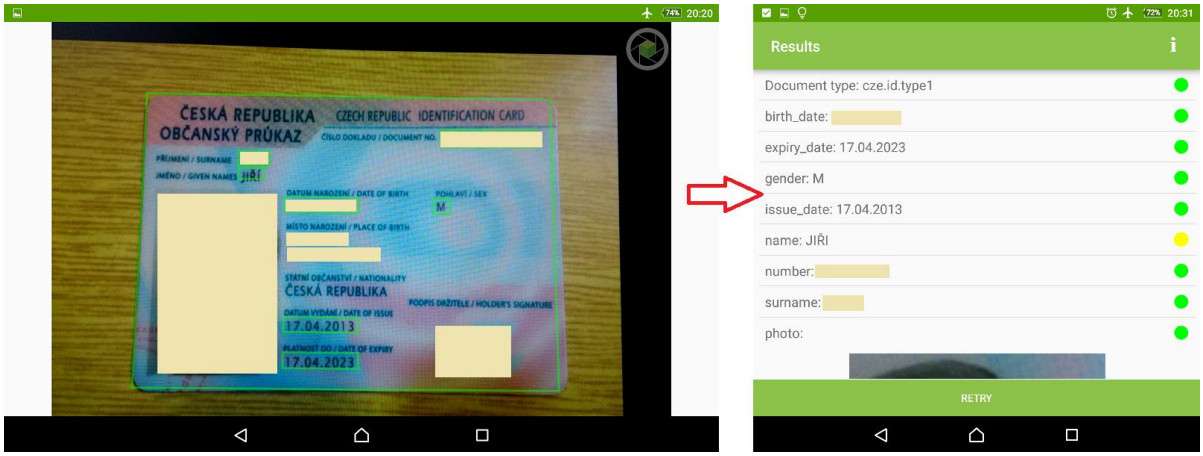
[14] V několika obdobích testováno na zařízeních Sony Xperia Z3 Tablet Compact (SGP621) a Samsung Galaxy Trend 2 Lite (SM-G318H)

i člověk, tak nelze jednoznačně určit, odkud jsou informace pro automatizovaný systém skutečně čerpány.

Protože je Netverify řešení výhradně serverového typu, je jeho použití podmíněno připojením k internetu a mobilní nebo webová aplikace tak sama údaje z dokladu nerozpoznává. Protože je vyčítání identifikačních údajů ověřováno člověkem, odpovídá tomu i doba zpracování jednoho dokladu, která ve většině případů odpovídá 2 a maximálně 5 minutám. V případě, že je doklad na fotografii nečitelný, může však být informace o této skutečnosti navrácena i později.

4.2.2.2 SMART ID READER

Řešení od společnosti Smart Engines [94] rozpoznává identifikační údaje z osobních dokladů přímo v mobilním zařízení, pomocí něhož je doklad snímán, a nepotřebuje tak ke své činnosti připojení k internetu. Podporovány jsou doklady mnoha zemí, pro Českou republiku je však výběr omezen jen na občanský průkaz nového typu, přičemž se starším typem průkazu a cestovním pasem si aplikace již neporadí. Na výběr je však i možnost čtení strojově čitelných oblastí, pomocí kterých lze tento nedostatek v případě potřeby alespoň částečně řešit.



Obr. 51: Rozpoznávání identifikačních údajů pomocí aplikace Smart ID Reader

Při rozpoznávání údajů aplikace využívá kontinuálního snímání obrazu, v rámci kterého shromažďuje hned několik snímků dokladu v řadě (viz Obr. 51 vlevo). Jakmile je tímto způsobem nashromážděno dostatek informací, což netrvá déle než 3 vteřiny, je uživateli zobrazen výstup s rozpoznanými identifikačními údaji (viz Obr. 51 vpravo). Aplikace je v případě občanského průkazu nového typu schopna rozeznat pouze část údajů z jeho přední strany (chybí místo narození a státní občanství držitele), přičemž údaje ze zadní strany již nepokrývá vůbec (označení úřadu, který doklad vydal a rodné číslo, trvalý pobyt a případný titul a rodinný stav držitele). Co se podporované sady znaků týká, aplikace je schopna

rozpoznávat základní česká diakritická znaménka, dlouze přehlasované U, například v příjmení „Műller“, však již nepodporuje. Přesnost rozpoznávání působí pro číselné údaje spolehlivě, pro jména a příjmení, která obsahují diakritiku, však její výkon klesá.

Společně s každým z rozpoznaných identifikačních údajů je navíc v pravé části jeho kolonky vyobrazena pomocí barvy puntíku i jistota, s jakou se jej podařilo rozpoznat. Zelený puntík značí vysokou jistotu, žlutý pak údaj, který nemusel být vyčten zcela správně.

4.3 Možnosti lokalizace dokladu

Jedno z posledních témat, kterému se bude analýza problematiky čtení osobních dokladů věnovat, je ověření možností, jakými je možné ve vstupní fotografii lokalizovat doklad.

Lokalizace dokladu v obraze je velmi důležitá úloha. Až na jejím základě je totiž možné určit, kde se v obraze nacházejí informace, na které se má zaměřit další zpracování. Stejně tak i znalost významu těchto informací vychází z jejich umístění v lokalizovaném dokladu. Zpracování fotografie osobního dokladu a porozumění informaci v ní obsažené však není triviální proces. Je potřeba brát v potaz, že snímek dokladu bude mít v důsledku nezávislosti na pořizovacím zařízení pokaždé jinou kvalitu obrazu. Různá bude i perspektiva foceného dokladu. Dále je nutné počítat s nerovnoměrným osvětlením a s přítomností šumu, proměnlivým kontrastem pozadí vůči textu, s neostrými fotografiemi a v neposlední řadě i se světlo odrazivým charakterem plochy dokladu. V následujících podkapitolách bude proto na různě kvalitních fotografiích zanalyzováno několik známých postupů, pomocí kterých lze k lokalizaci dokladu přistupovat, aby pak bylo možné určit, jaké z nich budou v budoucím řešení užitečné a funkční, a kterým se naopak již nevyplatí dále věnovat.

4.3.1 Lokalizace karty dokladu

Jedna z možností, jak k lokalizaci dokladu ve vstupním obraze přistupovat, je pokusit se v něm nalézt přímo samotnou kartu dokladu. Jejím ohraničením společně se znalostmi typu dokladu by pak bylo možné nalézt i patřičné identifikační údaje k rozpoznání.

4.3.1.1 Lokalizace dokladu pomocí hran

Pravděpodobně nejintuitivnější přístup k této problematice je pokusit se doklad v obraze segmentovat na základě jeho hran. Na motiv této myšlenky byl ve skutečnosti proveden i první pokus o implementaci lokalizace dokladu, jež bude nyní v rámci analýzy stručně popsán.

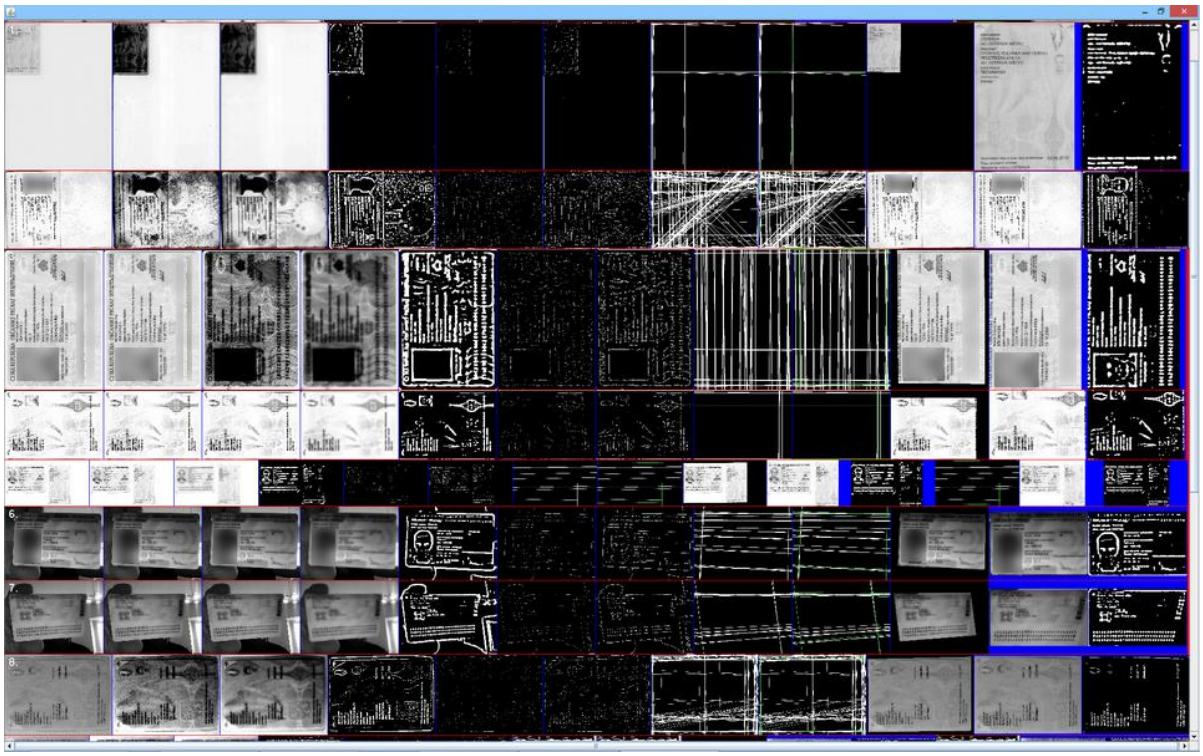
Algoritmus byl implementován v programovacím jazyce Java pomocí knihovny OpenCV [13]. V rámci jeho realizace bylo experimentováno s mnoha dílčími algoritmy jako

je manipulace s kontrastem, adaptivní prahování, Cannyho detektor hran [12], detekce kontur, detekce rohů, Houghova transformace, morfologické operátory a další. Pro dosažení výsledného ohraničení, respektive ořezu karty dokladu byl nakonec využit následující postup:

1. Normalizace histogramu
2. Rozostření (pro potlačení šumu)
3. Adaptivní prahování (OpenCV implementace)
4. Houghova transformace (upravená implementace poskytující i váhu jednotlivých nalezených přímek)
5. Nalezení rohů průkazu pomocí průsečíků Houghových přímek
6. Ořez a náprava perspektivy

V následných testech se 100 vzory obrazů dokladů dosahoval však algoritmus pouze 82% úspěšně ořezaných exemplářů. Během testování se ukázalo, že čím rozmanitější byla kvalita vstupních fotografií, tím složitější bylo předem nebo automatizovaně odhadnout správné nastavení parametrů jednotlivých dílčích algoritmů, a řešení, které by si v této úloze vedlo lépe, se již nalézt nepodařilo. Přístup založený na hledání hran byl proto následně z úvah o možnostech lokalizace dokladu vyloučen.

Pro testovací účely bylo vybudováno i grafické rozhraní (viz Obr. 52).

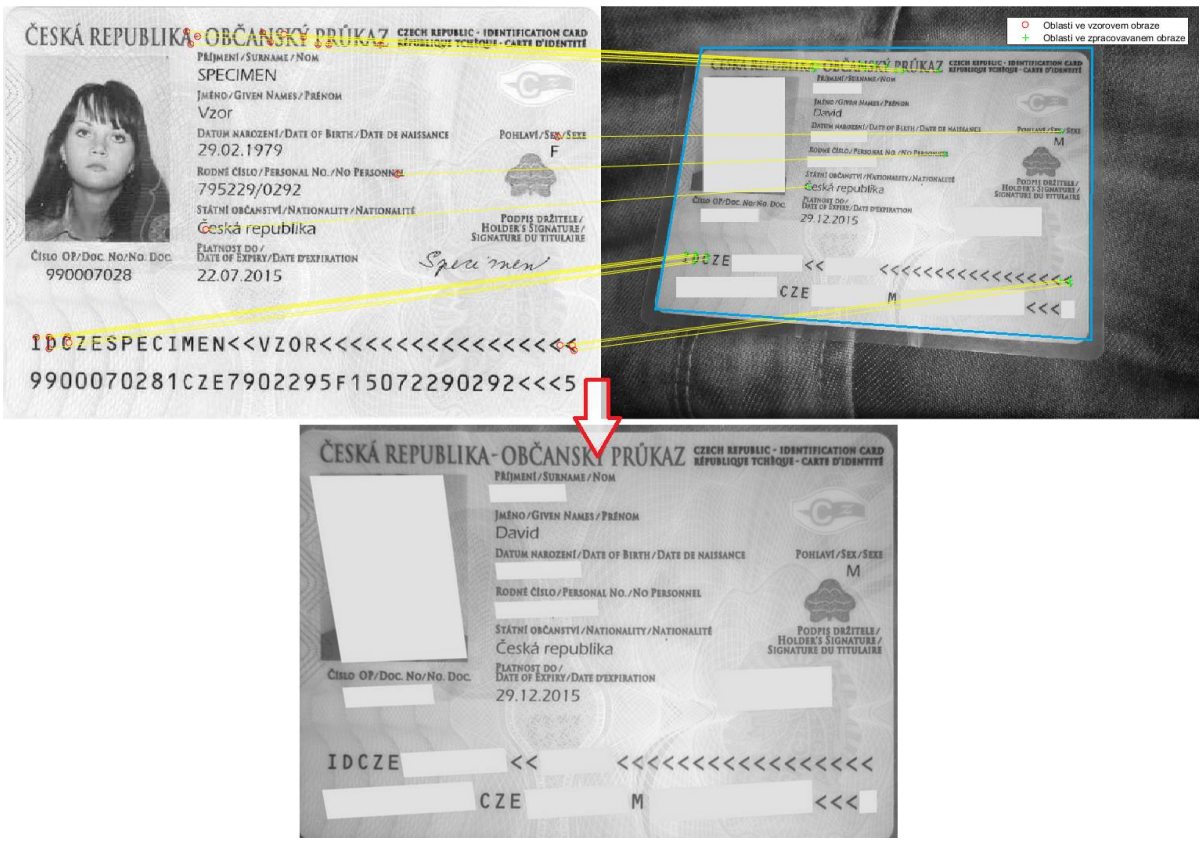


Obr. 52: Grafické rozhraní algoritmu pro detekci hran dokladu

4.3.1.2 LOKALIZACE DOKLADU POMOCÍ ZÁJMOVÝCH OBLASTÍ

Další možností, jak k lokalizaci karty dokladu přistoupit, je využít mechanismů zabývajících se problematikou hledání podobných zájmových oblastí (z angl. *features*) mezi dvěma různými obrazy. Tyto mechanismy jsou v anglické literatuře označovány souslovím *feature matching*.

Feature matching je pro lokalizaci dokladu možné použít tak, že jako první z obrazů je algoritmu předložen obraz vzorového dokladu a jako druhý z obrazů fotografie dokladu zpracovávaného. Za předpokladu, že je v obou obrazech přítomen stejný typ dokladu, může být jeho karta ve druhém z obrazů lokalizována pomocí transformační matice, která je sestrojena na základě shodných zájmových oblastí obou obrazů (viz Obr. 53 nahoře). Je-li pak tato transformační matice aplikována na zpracovávaný obraz, je jeho obrazová informace upravena tak, aby odpovídala dokladu ve vzorovému obrazu (viz Obr. 53 dole).



Obr. 53: Lokalizace dokladu vyhledáním podobných zájmových oblastí (nahoře) a aplikace následně sestrojené transformační matice (dole)

Feature matching potřebuje pro svou funkci operovat se dvěma aparáty. První z nich se nazývá detektor a druhý deskriptor. Detektor má za úkol ve vstupním obraze najít co nejvýznamnější body, podle kterých jej lze co nejlépe popsat, a deskriptor se pak stará již

o samotný popis oblasti, která daný bod obklopuje. Detektorů a deskriptorů přitom existuje celá řada a je možné je různě kombinovat.

Pro účel analýzy použitelnosti lokalizace dokladu pomocí zájmových oblastí byly v různých kombinacích testovány detektory SURF [98], ORB [99], BRISK [100], MSER [53], HARRIS [101] a FAST [102] a deskriptory SURF [98], FREAK [103], ORB [99] a BRISK [100]. Po jejich důkladné analýze bylo zjištěno, že pro fotografie dokladů s různou kvalitou obrazu je použitelný pouze deskriptor typu SURF. Ten pracoval nejlépe se svým párovým SURF detektorem, bylo by jej ale patrně možné použít i s detektorem typu MSER, se kterým rovněž dosahoval dobrých výsledků. Ostatní kombinace detektorů s deskriptory nenalézaly shodné zájmové oblasti v obrazech vůbec, nebo jich označily příliš nízký počet a nebylo na jejich základě možné sestrojit dostatečně přesnou transformační matici.

SURF deskriptor však společně se svým detektorem podléhají platnému patentu [104], který je znemožňuje volně použít v komerčních produktech a pro tuto práci tak přestávají být vhodnou technologií.

Diskuze

Několik měsíců až let po tom, co proběhla tato analýza, byl v knihovnách pro práci s obrazy implementován nový algoritmus detektoru a deskriptoru, nazvaný KAZE [105]. Tato kombinace detektoru s deskriptorem si vede v předešlých testech podobně jako SURF, KAZE však není zatížen patentem. Škoda, že tu nebyl dříve.

4.3.2 Lokalizace textu

Další možností, jak k lokalizaci dokladu přistoupit, je na rozdíl od předchozí kapitoly nejprve v obraze vyhledat text a až na základě jeho rozložení označit doklad, který jej obsahuje. Informace by také mohlo být možné číst i přímo z lokalizovaného textu a jejich význam určovat například podle nadpisů jednotlivých identifikačních údajů. Za účelem analýzy možností lokalizace textu v obraze bylo proto vyzkoušeno několik řešení třetích stran, které jsou na toto téma zaměřeny.

Ač je lokalizaci textu v obraze věnováno poměrně velké úsilí (viz kapitola 3.1), jen málo prací poskytuje výsledek svých řešení volně k vyzkoušení. Testována byla proto hlavně metoda, se kterou ve své práci přišel Neumann a Matas [106], a která je dostupná v knihovně OpenCV [107] a metoda od Gupta et al. [59][15]. Dále byly vyzkoušeny i práce podobné

[15] Metoda je k vyzkoušení volně dostupná na adrese http://zeus.robots.ox.ac.uk/textspot/

tématiky volně dostupné z projektové databáze Matlab [108], ty se však požadované kvalitě lokalizace řádek textu nepřibližovaly či obsahovaly chyby.

Pro testovací účely byly zvoleny dvě fotografie dokladu. První z nich představuje ostrý, kvalitní snímek, na němž je očekávána úspěšnost detekce blízká 100%. Druhý ze snímků představuje vzorek, který již tak kvalitní není, ale vzhledem k tomu, že je stále ještě člověkem bez problémů čitelný, by lokalizace textu v něm měla být co nejúplnější.

Obr. 54: Výsledky detekce textu pomocí OpenCV [107]

Z výsledků lokalizace textu metodou implementovanou v knihovně OpenCV [107] (viz Obr. 54) je patrné, že pro ostrý snímek dokáže nalézt většinu z přítomných řádek textu, méně kvalitní snímek však již zpracovat tak dobře nedokáže. Řádky jsou často nalezené jen z části, přičemž některé, jako například řádek „Jan“, ve výstupu chybí úplně. Podobně je na tom i metoda od Gupta et al. [59] (viz Obr. 55).

Obr. 55: Výsledky detekce textu pomocí metody od Gupta et al. [59]

Po dalším testování obou metod s fotografiemi horších kvalit nebylo v detekci textu zaznamenáno zlepšení, výsledky se naopak s klesající kvalitou obrazu zhoršují až do nepoužitelných mezí. Testována byla i volně dostupná implementace operátoru SWT [22] (viz

kapitola 2.1.1.3), výsledky však byly pro méně kvalitní fotografie dokladů rovněž neuspokojivé.[16]

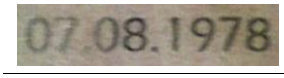
4.4 Možnosti rozpoznávání textu

Poslední položkou analýzy problematiky čtení osobních dokladů je ověření možností, jakými je možné provést již samotné rozpoznávání lokalizovaného textu. Cílem je přitom určit, jaké ze známých řešení by mohlo být v budoucím systému použitelné, a kterým se naopak nevyplatí věnovat další pozornost.

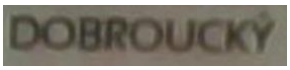
Po detailnějším ohledání dostupných vzorových fotografií bylo zjištěno, že nadpisy jednotlivých identifikačních údajů, které se na dokladech vyskytují, často nelze považovat za čitelné ani člověkem, a byly proto z analýzy vyřazeny. Znamená to také, že se na jejich vyčítání nelze v budoucím systému spoléhat. Se zbylými řádky textu byly testovány řešení Tesseract [109], JavaOCR [110], Ocrad [111] a OCRopus [112].

Z testování vyplynulo, že si s řádky textu tak, jak se vyskytují na fotografiích osobních dokladů, tedy bez úprav, nedokáže spolehlivě poradit ani jedno z řešení. Jednotlivé řádky textu byly proto před dalšími pokusy o jejich rozpoznávání předzpracovány prahováním, což vedlo ke znatelnému zlepšení a výstupy z jednotlivých řešení tak začaly produkovat srozumitelné výsledky. Nejlépe si z testovaných řešení vedl Tesseract, jehož výsledky lze pro některé vstupy pozorovat v tabulce 1, ani ten však nedokázal korektně rozpoznat všechny předložené řádky textu a budoucí řešení by tak pro rozpoznávání identifikačních údajů mohlo profitovat z vlastního, specializovaného klasifikátoru.

Tab. 1: Některé vstupní a výstupní hodnoty z testování rozpoznávání řádek textu pomocí Tesseract [109]

Vstupní obraz	Tesseract výstup
	07.08. 1918.
	07.08.1978
	am
	WASH ER
	okr. Lnoumcz

[16] Jak si se stejnými fotografiemi poradí výsledné řešení, je možné vidět na Obr. 64 na straně 66.

	okr. LITOMĚŘICE
	WAV
	CUS l'AV
	DOUIOUCKÝ
	DOBROUCKÝ

4.5 SHRNUTÍ ANALÝZY

Z analýzy problematiky čtení osobních dokladů vyplynulo, že zpracovávání fotografií o různé kvalitě obrazu může pro budoucí systém představovat nemalou výzvu, což je potvrzeno i nestabilními výsledky jednotlivých testovaných algoritmů. Zanalyzované algoritmy zaměřené jak na úlohu lokalizace, tak na úlohu rozpoznávání textu, pro fotografie s nižší kvalitou obrazu nedosahují takových výsledků, na které by mohlo jít přímo navázat, a jejich použití by proto muselo být závislé na dalším zpracování. Bylo proto vhodné zamyslet se nad návrhem a implementací vlastního řešení, které by díky úzkému zaměření na fotografie s osobními doklady mohlo podávat lepší výsledky.

Z analýzy projektů s podobnou tématikou vyplynulo, že je v poli existujících řešení rozpoznávání identifikačních údajů z osobních dokladů co vylepšovat. Žádný z projektů totiž nedokáže poskytnout pro české doklady kompletní výčet údajů, které obsahují, a často nepodporují českou abecedu znaků.

Analýza jednotlivých typů osobních dokladů pak odhalila, že, aby bylo budoucí řešení rozpoznávání identifikačních údajů úspěšné, mělo by počítat s pohyblivými bloky textu, které se vyskytují na zadní straně občanského průkazu nového typu, a s odlesky ochranných prvků, které jsou na dokladech přítomny.

4.5.1 STANOVENÍ POŽADAVKŮ

Na základě provedené analýzy byly pro řešení rozpoznávání identifikačních údajů z osobních dokladů stanoveny tyto požadavky:

- Řešení by mělo být schopné zpracovat oba typy platného českého občanského průkazu a český cestovní pas
- Řešení by mělo být schopné z fotografie dokladu vyčíst jakýkoliv identifikační údaj, zvláště pak informaci o rodném číslu a trvalém pobytu držitele

- Řešení by mělo podporovat plnou škálu znaků české abecedy a navíc i znaky, které se běžně v českých dokladech vyskytují (přehlásky ve jménech apod.)
- Řešení by mělo být schopné zpracovat fotografie dokladů s horší kvalitou obrazu
- Pokud to bude možné, řešení by nemělo být závislé na technologiích, které by jej následně neumožňovaly bezplatně použít i komerčně

5 POPIS ŘEŠENÍ

V návaznosti na předchozí analýzu vznikalo v průběhu řešení této práce mnoho návrhů, jak k problematice čtení identifikačních údajů z dokladů přistupovat, řada z nich se ale v následné implementační fázi ukázala jako neúčinná[17] a bylo od nich proto upuštěno ve prospěch jiných alternativ. Vzhledem k této skutečnosti, rozsahu práce a tomu, že je systém stále ještě aktivně vylepšován, se namísto návrhu řešení tato kapitola bude věnovat spíše jeho popisu ve stavu, v jakém se aktuálně nachází s tím, že jakékoliv rozsáhlejší implementační pasáže budou pro zachování přehlednosti popsány až v následující kapitole.

Popis řešení je členěn do třech částí v podobě lokalizace textu, rozpoznávání textu a korekce rozpoznaného textu tak, aby jejich sled odpovídal i skutečné návaznosti operací v popisovaném systému. V jednotlivých částech přitom bude zvlášť přistupováno k problematice týkající se strojově čitelné oblasti a k problematice ostatních údajů na dokladu. Každá z těchto oblastí je totiž vzhledem k různým požadavkům a časovému rozestupu mezi jejich řešeními, který činí dva a půl roku, založena na odlišných postupech.

5.1 LOKALIZACE TEXTU

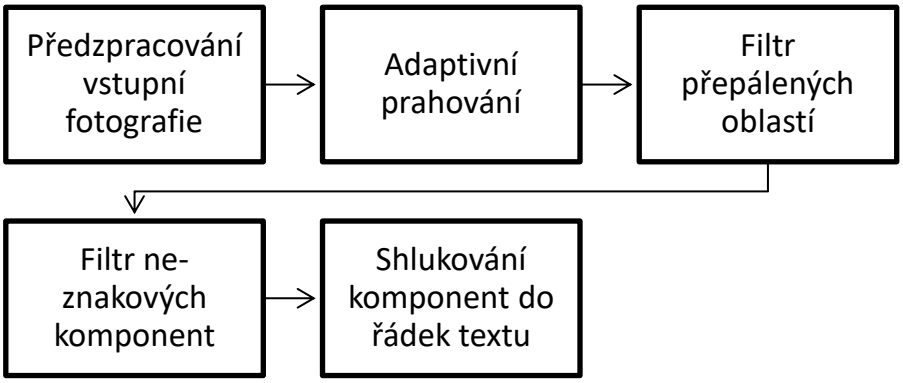
Lokalizace textu je vstupní branou celého systému. Jejím cílem je ve vstupním obraze najít všechny řádky textu, porozumět jejich významu a korektně je z obrazu extrahovat tak, aby je bylo možné v další fázi co nejúspěšněji rozpoznat.

Veškeré algoritmy související s lokalizací textu jsou implementovány v jazyce Matlab od společnosti MathWorks [14], a to zejména pro jeho bohatou nabídku předpřipravených funkcí a procedur pro práci s obrazovými daty.

5.1.1 LOKALIZACE ŘÁDEK TEXTU

Lokalizace řádek textu je založena na principu vyhledání spojených komponent v binarizovaném obrazu. Spojené komponenty jsou následně filtrovány až do té doby, než zůstanou jen ty komponenty, které náleží textu. Náčrt algoritmu je znázorněn na obrázku 56.

[17] Některé z těchto návrhů byly pokryty již v předešlé kapitole v podobě možností, jak k dané problematice přistupovat.



Obr. 56: Jednotlivé kroky algoritmu pro lokalizaci řádek textu

5.1.1.1 PŘEDZPRACOVÁNÍ VSTUPNÍ FOTOGRAFIE

Na vstupu je fotografie převedena z barevného formátu na černobílý. Je totiž předpokládáno, že barevná informace nenese v případě fotografie osobního dokladu žádnou cennou informaci. Veškerý text je totiž ve složkách intenzity obrazu podobně čitelný jako s jeho barevnými kanály. Tímto krokem je problém zjednodušen z 3 dimenzí barev na jedinou a navazující výpočty tak mohou být znatelně urychleny.

Následně je zkontrolována velikost fotografie. Je-li její delší strana menší než 1000 bodů, je obraz při zachování poměru stran zvětšen na 1280 bodů. Jako interpolační algoritmus byl použit lanczos3, protože oproti ostatním zkoumaným algoritmům (bilineární, bikubický, nejbližší soused) nemá tendence zvětšovaný obraz rozostřovat.

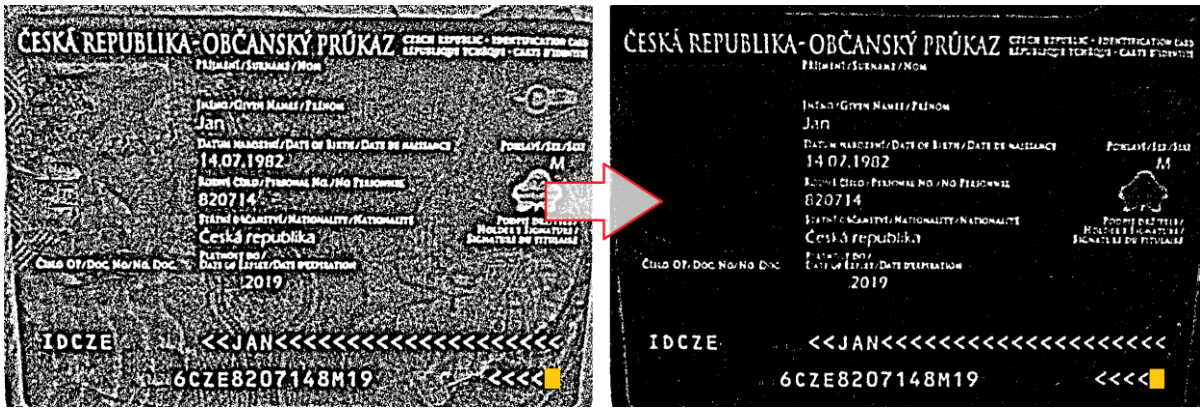
5.1.1.2 ADAPTIVNÍ PRAHOVÁNÍ

Adaptivní prahování je nejdůležitější krok celého procesu detekce řádek textu. Na jeho výkonu totiž závisí zbytek operací a je tedy nutné, aby byl dostatečně přesný nezávisle na kvalitě vstupní fotografie. Zároveň je pro jeho automatický chod nezbytné, aby jeho operace nebyla podmíněna jakýmikoli vstupními parametry, které by nebylo možné odvodit přímo na základě poskytnuté fotografie.

Testováno bylo hned několik algoritmů pro adaptivní prahování: Niblack [25], Bradley [26], Feng [28], Sauvola [29], Bernsen [113], Nick [114], Wolf [115] a běžné prahování klouzavým oknem [27], jež se v obdobné formě vyskytuje i v knihovně OpenCV [13]. Nejlépe si po stránce čitelnosti binární podoby fotografie vedl algoritmus Bradley, jehož vstupní parametry však musejí být přizpůsobeny vlastnostem vstupního obrazu a jejich výběr se nepodařilo zautomatizovat. Stejným nedostatkem se vyznačují i ostatní algoritmy, a bylo proto rozhodnuto provést úpravu běžného prahování klouzavým oknem [27] tak, aby pro svoji

činnost v rámci binarizace fotografií osobních dokladů nepotřebovalo žádné další vstupní parametry.

Původní algoritmus má celkem dva vstupy. První určuje velikost klouzavého okna a druhý konstantu C. První parametr lze odvodit od celkové velikosti fotografie respektive od očekávané velikosti řádek textu, druhý parametr má být odstraněn. Princip algoritmu spočívá v tom, že každý pixel promění buď na bílý, nebo černý podle toho, jak vypadá jeho okolí v klouzavém okně. Je-li průměr hodnot odstínu šedi okolních bodů nižší než proměňovaný, je nastaven na černou barvu, protože je považován za pozadí (text je na osobních dokladech vytištěn tmavými barvami). V opačném případě je pixel obarven na bílo. Vstupní konstanta C je pak při rozhodování o výstupní barvě odečtena od průměrné hodnoty okolí a zabraňuje tak nerozhodnému počínání v místech jednolitého pozadí (viz Obr. 57).

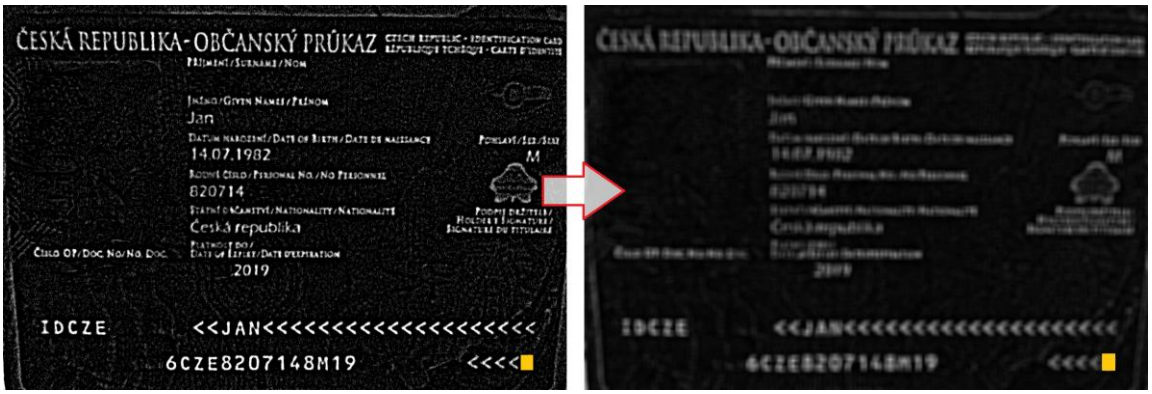


Obr. 57: Význam vstupní konstanty původního algoritmu pro adaptivní prahování (viz [27])

Metoda pro odstranění druhého vstupního parametru spočívá v tom, že je hodnota každého bodu ve vstupním obrazu přepsána na výsledek rozdílu mezi průměrnou hodnotou jeho okolí a hodnotou jeho samého. Na tento obrázek je následně aplikována Otsuova metoda [24] pro nalezení ideální hodnoty pro prahování. Výsledek operace je použit stejným způsobem jako konstanta C v předchozím řešení, tentokrát již ale algoritmus nepotřebuje ke své činnosti jiný vstupní parametr, než je velikost klouzavého okna – konstanta je vypočtena automaticky.

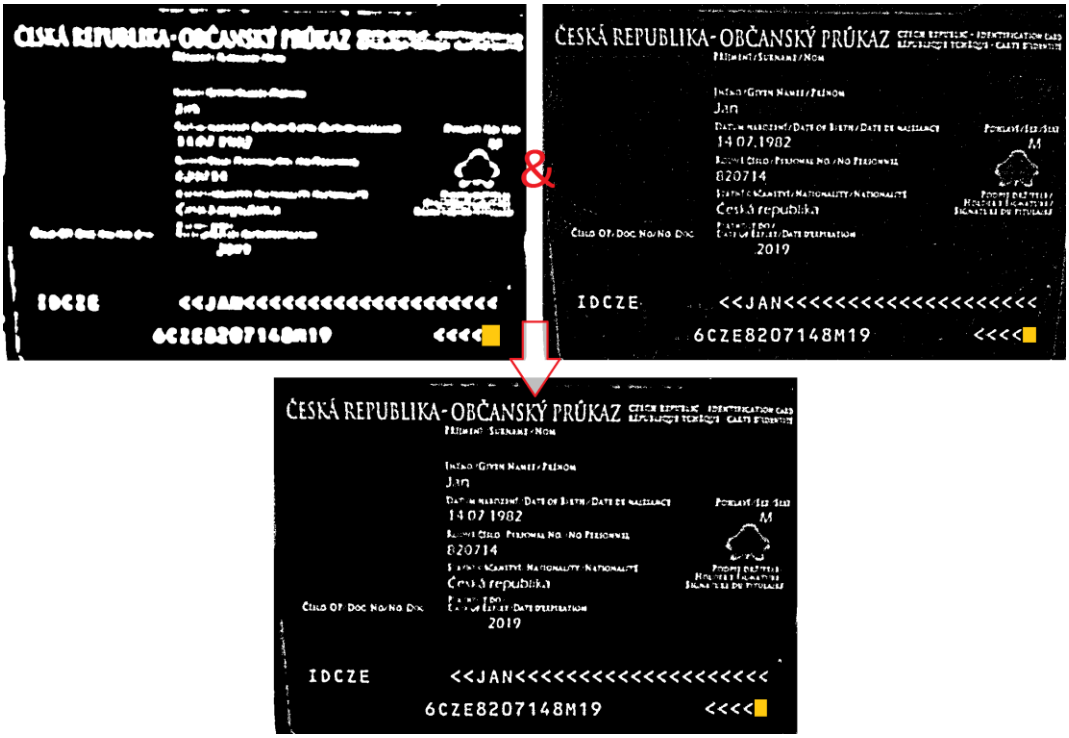
Výsledný obraz je takřka shodný jako snímek na obrázku 57 vpravo. Po pečlivém zkoumání je však patrné, že výstup obsahuje společně s binarizovaným textem i nemalé množství šumu, který s klesající kvalitou fotografie stoupá. Algoritmus byl proto dále zdokonalen.

Pro odstranění šumu je znovu použit obraz, který vznikl rozdílem průměrné hodnoty okolí a bodu uprostřed klouzavého okna, jehož podobu lze pozorovat na obrázku 58 vlevo (intenzita odstínů šedi byla upravena tak, aby byl obraz pozorovatelný lidským okem). Na tento obraz je pak aplikován rozostřovací filtr s oknem o velikosti původního klouzavého okénka (viz Obr. 58 vpravo).



Obr. 58: Rozdílový obraz (vlevo) a jeho rozostření (vpravo)

Následně je pomocí Otsuovy metody rozostřený rozdílový obraz oprahován. Díky rozostření však výsledek již neobsahuje šum a lze jej použít jako masku pro původní výstup upraveného algoritmu (viz Obr. 59 nahoře), čímž vzniká finální produkt této funkce (viz Obr. 59 dole). Za konečný výstup nelze považovat samotnou zmiňovanou masku, protože kvůli rozostření neobsahuje znaky textu patřičné kvality.



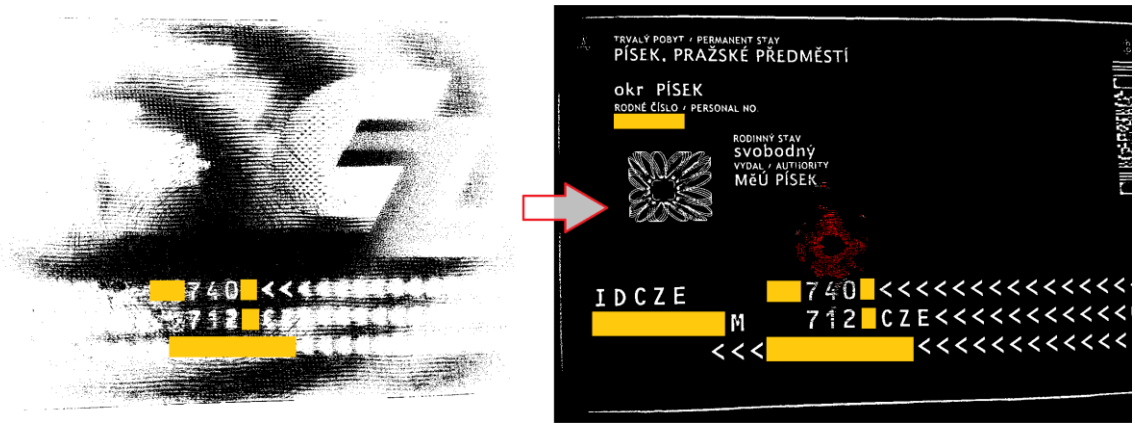
Obr. 59: Filtrace šumu a konečný výstup upraveného algoritmu pro adaptivní prahování

5.1.1.3 FILTR PŘEPÁLENÝCH OBLASTÍ

Obr. 60: Vstupní fotografie

Filtr přepálených oblastí vznikl pro účel odstranění takových bílých bodů z výstupního obrazu adaptivního prahování, které odpovídají ne-znakům v podobě odleskové plochy dokladu. Komponenty odleskových ploch jsou přítomny například tehdy, byl-li při pořizování fotografie použit blesk fotoaparátu (viz Obr. 60).

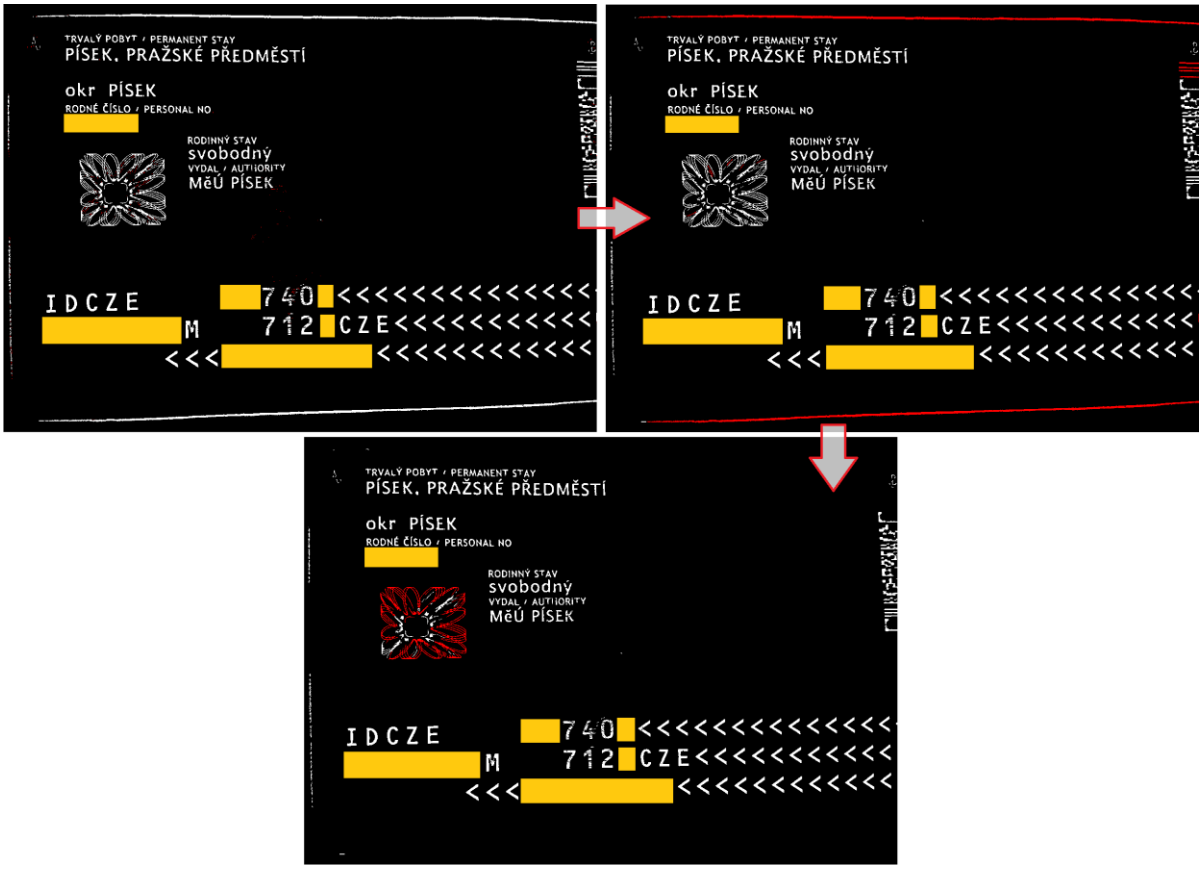
Pro nápravu takové fotografie je spektrum intenzity původního snímku normalizováno tak, aby maximální hodnota odpovídala bílé a minimální hodnota černé barvě. Tímto způsobem pak lze snímek oprahovat pomocí konstantní hodnoty odpovídající odstínu šedi intenzity 190. Tato hodnota byla vyvozena empiricky a odděluje přesvětlené plochy dokladu od těch ostatních. Výsledkem je pak maska, kterou lze použít jako filtr pro odstranění přesvětlených ploch (viz Obr. 61).



Obr. 61: Maska přepálení a její aplikace (červené komponenty jsou odfiltrovány)

5.1.1.4 Filtr ne-znakových komponent

V následující fázi algoritmus nalezne všechny spojené komponenty současného obrazu a pro každou z nich vypočítá vlastnosti v podobě její velikosti, výstřednosti a konvexnosti (viz kapitola 2.1.3). Pomocí empiricky vyvozených prahů jsou následně komponenty neodpovídající znakům odstraněny (viz Obr. 62).



Obr. 62: Filtr velikosti (vlevo nahoře), výstřednosti (vpravo nahoře) a konvexnosti (dole) spojených komponent (červené komponenty jsou odfiltrovány)

5.1.1.5 Shlukování komponent do řádek textu

V tuto chvíli je již binarizovaný snímek dokladu zbaven všech jednoduše odhalitelných spojených komponent, které nepředstavují znaky textu. Shlukování komponent do řádek textu tvoří poslední krok algoritmu, na jehož výstupu jsou očekávány již samotné souřadnice textových řádek. Princip, na kterém tento krok funguje, je následující.

Algoritmus uchopí komponentu a hledá, jestli v horizontální vzdálenosti její výšky existuje komponenta sousední. Pokud ano a leží přibližně ve stejné vertikální poloze, jsou komponenty označeny jako příbuzné. Příbuzné komponenty postupně utvoří řadu komponent, která odpovídá vlastnostem řádky textu. Algoritmus kromě vzdálenosti zahrnuje do výpočtů

ještě velikost komponent a spodní linku řádky textu. Prahy jsou určeny empiricky a jsou do jisté míry odolné vůči drobným výkyvům hodnot. Komponenty, které nelze sdružit, jsou považovány za ne-znaky. Tímto způsobem jsou odfiltrovány i poslední komponenty ne-znaků v obraze a na výstupu jsou již zbylé komponenty seřazeny do kolekcí, které odpovídají řádkám textu (viz Obr. 63).

Obr. 63: Filtrace spojených komponent na základě shlukování (vlevo) a sdružené komponenty (vpravo)

Aby bylo možné popsané řešení lokalizace řádek textu porovnat s metodami třetích stran, kterými se zabývala analýza problematiky čtení dokladů v kapitole 4.3.2 na straně 56, byl algoritmus aplikován i na původní fotografie, s nimiž bylo v rámci analýzy operováno. Výsledek je možné pozorovat na obrázku 64.

Obr. 64: Aplikace popsaného řešení na fotografie dokladů použitých v předchozí analýze

5.1.2 Určení významu a korekce nalezených řádek textu

Po tom, co byly ve vstupním obrazu řádky textu lokalizovány, je nyní algoritmus zaměřen na určení jejich významu tak, aby bylo jednoznačné, jaké identifikační údaje představují.

Obr. 65: Šablona rozložení řádek textu občanského průkazu nového typu

Algoritmus pracuje s předdefinovanými šablonami dokladů, respektive se souřadnicemi řádek textu v nich obsažených (viz Obr. 65). Rozeznávány jsou tři typy řádků: řádek, jehož proporce se napříč doklady nemění (červené orámování), řádek, který má variabilní délku (jeho počátek označen modře) a řádek, který nemusí být v obraze lokalizován (označen fialovou barvou). Pro každou stranu všech podporovaných typů dokladů je definována jedna taková šablona s výjimkou zadní strany občanského průkazu nového typu, pro který byly vyhotoveny celkem tři varianty rozložení řádek textu tak, aby pokrývaly co nejširší škálu jejich různých umístění (viz kapitola 4.1.2 a Obr. 45).

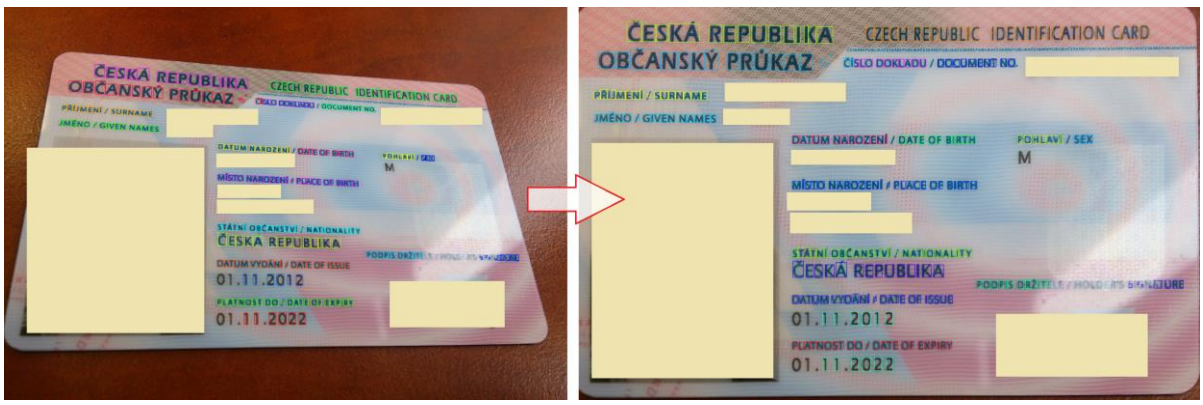
Kromě souřadnic šablona obsahuje ještě definice vztahů mezi všemi možnými kombinacemi dvojic řádek textu, jejichž proporce se nemění (červené orámování na Obr. 65). Mezi tyto vztahy patří poměr jejich délek a dvojice úhlů, které řádky mezi sebou vzhledem ke svým počátkům a koncům svírají.

Na základě této definice je pak v kolekci dříve lokalizovaných řádek textu vyhledána taková dvojice řádek, která svými vzájemnými vztahy odpovídá vztahům některé dvojice řádek ze šablony. Jsou-li dvě takové dvojice řádek nalezeny (zelené šipky na Obr. 66), je možné, že jsou ekvivalentní a že je jejich význam shodný. Pro to, aby byla tato hypotéza ověřena, je na základě rozdílů souřadnic těchto dvou párů řádek vypočtena transformační matice, pomocí které jsou všechny řádky dokladu ve vstupním obraze „přiloženy“ k řádkám dokladu v šabloně (viz Obr. 66). Následně algoritmus na základě překryvu souřadnic ověřuje, jestli pro řádky textu ze šablony existují i řádky textu ve vstupním obraze. Pokud součet překrývajících se řádek překoná určitý práh (určen empiricky), je původní hypotéza označena za správnou. Není-li počet překrývajících se řádek dosažen, pokračuje algoritmus v hledání nových hypotéz.

Obr. 66: Hledání významu řádek textu na základě jejich překryvu s šablonou

Překrytí řádek textu je ověřováno jak ve vertikální, tak v horizontální ose. V případě, že se jedna řádka šablony překrývá v horizontální úrovni vstupního obrazu s více řádkami, jsou tyto řádky spojeny do jedné (nadpis „podpis držitele“ na Obr. 66).

Je-li hypotéza o významu řádek potvrzena, pokračuje algoritmus tak, že pomocí dříve vypočtené transformační matice transformuje nyní již celý vstupní obraz, čímž jeho pozici v prostoru srovná s pozicí, v jaké se nachází šablona (viz Obr. 67). Tímto krokem jsou eliminovány jakékoliv výchylky v posunutí, rotaci, zvětšení a perspektivě vstupního snímku.



Obr. 67: Transformace vstupního obrazu

Jednotlivá ohraničení řádek textu šablony jsou pak na základě původně lokalizovaných řádek a nového prahování, které je nyní zaměřeno přímo na pozici daného řádku, zpřesněny

tak, aby mohly být pro další zpracování z obrazu s co nejmenší chybou extrahovány (viz přechod z červeného ohraničení na zelené vpravo na Obr. 68). Algoritmus je zároveň schopen na základě šablony vyloučit ty řádky, které nenesou užitečné informace a naopak zpětně dohledat ty, které předchozí proces nebyl schopen lokalizovat (modré ohraničení na Obr. 68 vpravo).

Obr. 68: Vylučování řádek neodpovídajících textu, dohledávání chybějících řádek a zpřesňování jejich ohraničení

5.1.3 STROJOVĚ ČITELNÁ OBLAST DOKLADU

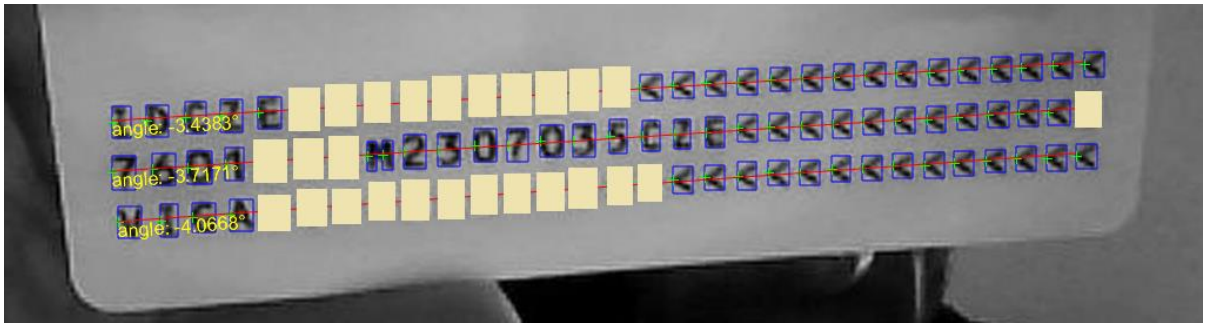
Ačkoliv by lokalizace strojově čitelné oblasti mohla být provedena algoritmem pro korekci a určování významu řádek textu z předešlé kapitoly, v současném řešení tomu tak prozatím není a zpracování strojově čitelné zóny tak navazuje přímo na lokalizaci řádek textu z kapitoly 5.1.1. Díky této skutečnosti není čtení strojově čitelné oblasti vázáno na konkrétní typ dokladu a je tedy možné je aplikovat na jakýkoliv dokument obsahující strojově čitelnou zónu dle dokumentu ICAO 9303 [83] (např. cizokrajné doklady).

Strojově čitelné oblasti se podle dokumentu ICAO 9303 [83] dělí na tři typy. Dva z nich mají 2 řádky, třetí má řádky 3. Na základě jejich definic pracuje algoritmus pro nalezení řádek strojově čitelné zóny (SČZ) s následujícími předpoklady:

- Řádky SČZ patří svou délkou mezi nejdelší lokalizované řádky
- Řádky SČZ mají přibližně stejnou délku
- Existují minimálně 2 a maximálně 3 řádky SČZ
- Řádka SČZ obsahuje minimálně 30 znaků
- Řádky SČZ na sebe vertikálně přímo navazují

V souladu s výše uvedenými předpoklady jsou mezi lokalizovanými řádky textu identifikovány takové řádky, které náleží strojově čitelné oblasti dokladu.

Algoritmus pokračuje výpočtem úhlu sklonu těchto řádek. Ten je proveden za pomoci souřadnic spojených komponent, kterými jsou řádky definovány ještě z kroku obecné detekce řádek textu. Pro každý řádek strojově čitelné zóny je sestavena kolekce bodů, kterou tvoří polovina výšky levé stěny ohraničení všech spojených komponent, které řádek utvářejí (viz zelené body na Obr. 69). Těmito body je následně proložena přímka, jejíž úhel odpovídá sklonu odpovídajícího řádku (viz Obr. 69).



Obr. 69: Výpočet úhlu sklonu řádek strojově čitelné oblasti dokladu

Tvoří-li absolutní hodnota úhlu sklonu řádku hodnotu větší než 0.5 stupňů, je řádek rotován v opačném směru tak, aby úhel po operaci činil 0 stupňů. Řádky jsou rotovány jednotlivě, protože rozdíl úhlu sklonu mezi různými řádky může být z důvodu pokřivené perspektivy snímku větší než 0.5 stupňů (viz Obr. 69). Řádky se sklonem menším jsou v tomto stavu ponechány. Řádky strojově čitelné zóny jsou nakonec, stejně jako ostatní řádky, z obrazu pro účely dalšího zpracování extrahovány.

5.2 Rozpoznávání textu

Úloha rozpoznávání textu přímo navazuje na jeho lokalizaci a plně spoléhá na to, že obraz na vstupu již netvoří nic jiného, než samotná řádka textu. Termín rozpoznávání pak označuje proces vyčtení textu z obrazu stejně, jako to při pohledu na obraz s textem umí člověk. Vstup je proto tvořen obrazem, zatímco výstup má již podobu textového řetězce.

5.2.1 Rozpoznávání textu mimo strojově čitelnou oblast

Pro rozpoznávání textu mimo strojově čitelnou oblast je využívána residuální konvoluční neuronová síť (viz kapitola 3.3.3). Na rozdíl od běžných řešení typu OCR (z angl. Optical Character Recognition), které text rozpoznávají po jednotlivých znacích a jsou tak závislé na jejich předchozí separaci, je však zmíněná neuronová síť sestavena takovým způsobem, aby dokázala rozpoznávat celé řádky textu bez jakéhokoliv jejich předchozího zpracování. Jednotlivé znaky a slova je tak síť schopna separovat zcela autonomně.

Veškeré algoritmy pro rozpoznávání textu mimo strojově čitelnou oblast dokladu jsou implementovány v jazyce Python a pomocí softwarové knihovny TensorFlow [116] (viz kapitola 6.1.3).

5.2.1.1 ARCHITEKTURA NEURONOVÉ SÍTĚ

Architekturu použité neuronové sítě, která má celkem 34 vrstev[18], je možné pozorovat na obrázku 70. Model je uveden konvoluční vrstvou o 64 filtrech velikosti 7x7 s krokem 2 a max-pooling vrstvou o velikosti filtru 3x3 s krokem 2. Tyto vrstvy společně vstupní data čtyřnásobně zmenšují a značně tak urychlují jak proces učení, tak samotný dopředný průchod sítě. Jádro neuronové sítě je tvořeno celkem čtyřmi bloky, jež pracují postupně s 96, 192, 384 a 786 filtry o shodných rozměrech 3x3 a kroky o velikosti 1. Konvoluční vrstvy jsou v těchto blocích sdruženy ještě do takzvaných stavebních bloků, které mají napříč modelem stejnou strukturu. V každém stavebním bloku jsou zapouzdřeny dvě konvoluční vrstvy s filtry o hodnotách určených rodičovským blokem s výjimkou prvních konvolučních vrstev 2., 3. a 4. bloku, které mají krok nastaven na hodnotu 2 a dochází tak k dalšímu redukování množství přenášených informací. Stavební bloky jsou napříč hlavními bloky distribuovány v počtech 3, 4, 6 a 3, čímž samy o sobě zastupují 32 z celkových 34 vrstev. Poslední skrytou vrstvu neuronové sítě tvoří max-pooling vrstva s filtrem o velikosti 2 a kroku 2, která je již napojena na vrstvu výstupní. Ta je složena z n plně propojených vrstev o $q + 1$ neuronech, kde n odpovídá maximální podporované délce řádky textu na vstupu a q počtu znaků v podporované abecedě ($+1$ pro speciální znak symbolizující prázdnou pozici v řetězci, viz dále).

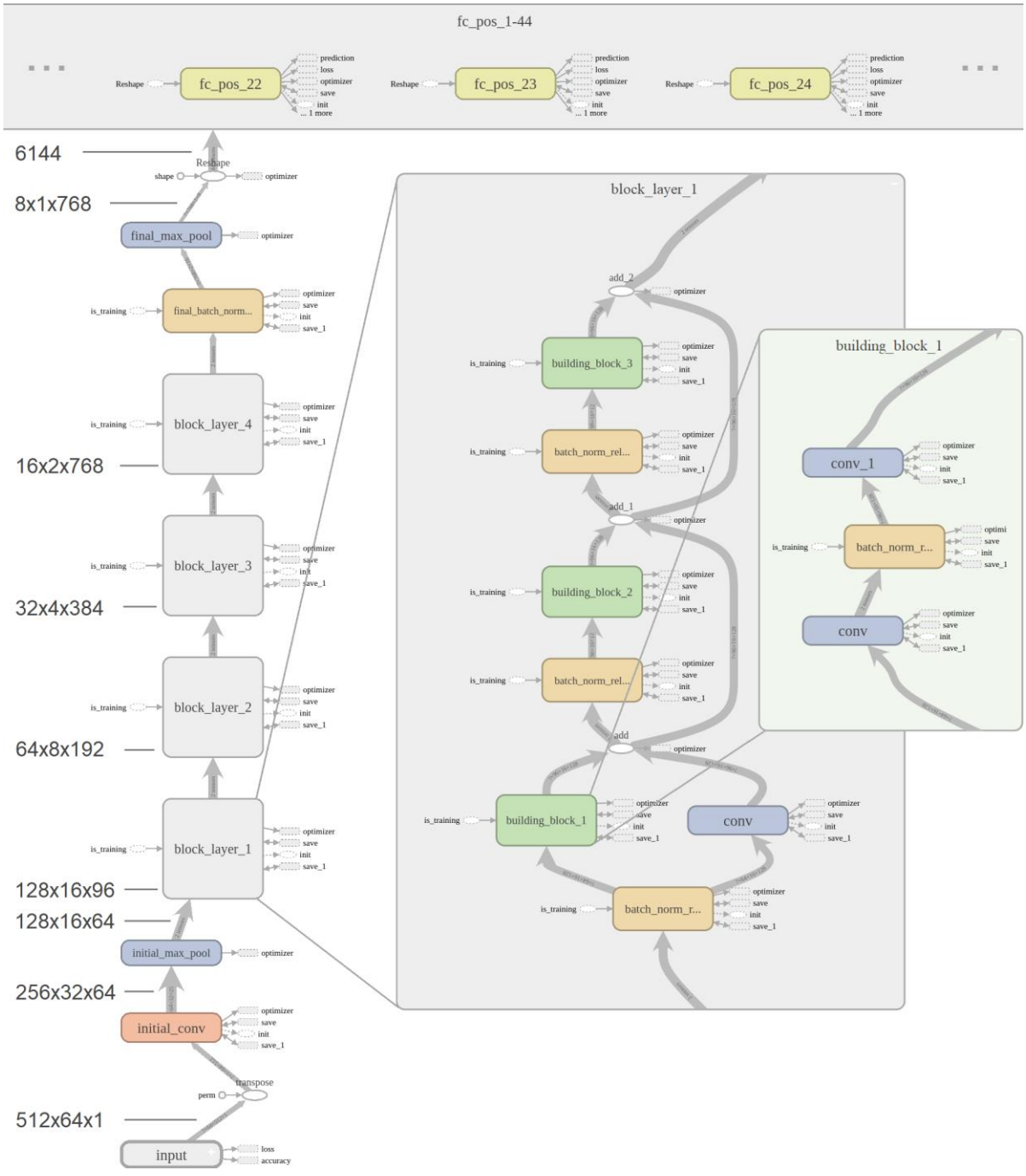
Vstupní obraz je převeden do odstínů šedi a je zvětšen nebo zmenšen bez zachování jeho proporcí tak, aby vyhověl velikosti vstupní vrstvy neuronové sítě. Obraz je poté ještě standardizován odečtem svého aritmetického průměru a podílem své standardní deviace.

Na počátku každého z bloků je přítomna v residuální části spojení jedna konvoluční vrstva navíc, která má za úkol navýšit počet filtrů na vstupu na počet filtrů odpovídající vlastnostem rodičovského bloku tak, aby pak mohla být v dalším kroku sečtena s výsledky z přilehlého stavebního bloku. Filtr této „převodní“ konvoluční vrstvy má velikost 1x1 a krok odpovídající kroku první konvoluční vrstvy daného bloku.

Napříč celým modelem je extenzivně používána batch normalizace (z angl. batch normalization, viz [117]). Jako aktivační funkce je pro všechny neurony, kromě těch ve

[18] Do toho součtu nejsou započítány paralelní vrstvy v podobě jednotlivých plně propojených vrstev na výstupu a převodních konvolučních vrstev na počátku každého z bloků. Tyto vrstvy by daný součet rozšířily o dalších až 47 položek. Dále se dle konvencí do součtu nezapočítávají ani ty vrstvy, které neobsahují váhy (např. vstupní a pooling vrstvy).

vstupní a výstupní vrstvě, použita ReLU funkce (viz kapitola 2.2.1.1) a proces učení neuronové sítě jako celku byl proveden standardní metodou Stochastic Gradient Descent (viz kapitola 6.1.3.3).



Obr. 70: Model residuální konvoluční neuronové sítě používaný pro rozpoznávání celých řádek textu

To, že je síť ve vstupním obraze sama schopna rozeznávat jednotlivé znaky textu je dáno její výstupní vrstvou. Tradičně je výstupní vrstva neuronové sítě zastoupena jedinou plně propojenou vrstvou, která představuje právě jeden klasifikátor. Model na obrázku 70 má však

výstupní vrstvu složenou z mnoha takových klasifikátorů, přičemž každý z nich je nezávisle a paralelně připojen k předešlé vrstvě sítě stejným způsobem, jako by tomu bylo v tradičním provedení s jediným klasifikátorem. Pomocí tohoto zapojení a předchozího učení je pak každý z těchto klasifikátorů v obraze zaměřen na jinou pozici textu a dohromady tak utvářejí klasifikátor, který vyjadřuje posloupnost znaků. Protože je neuronová síť složena z předem definovaného a konečného množství těchto pod-klasifikátorů, je jím omezena i maximální délka textu, kterou je síť ještě schopna vyjádřit. Pro vyjádření posloupnosti znaků, jejichž délka je menší, než množství pod-klasifikátorů, je navíc každý pod-klasifikátor schopen vyjádřit skutečnost, že se na dané pozici žádný znak již nevyskytuje. Tato skutečnost je daným pod-klasifikátorem indikována aktivací speciálního neuronu, který byl do každého z nich pro tento účel zabudován (níže označován jako znak [None]).

Diskuze

Podobné paralelní uspořádání výstupní vrstvy poprvé představili Goodfellow et al. [47] a poté Jaderberg et al. [68], kteří pomocí něj v obrazech vyčítali pěticiferná čísla, respektive slova až o 23 znacích (viz kapitola 3.2). Rozpoznávání celých řádek textu o libovolném počtu slov je však již originálním příspěvkem této práce.

5.2.1.2 Model sítě v závislosti na typu identifikačního údaje

Aby bylo rozpoznávání řádek textu co možná nejpřesnější, jsou pro vyčítání identifikačních údajů použity celkem tři různé modely neuronové sítě. Tyto modely byly speciálně navrženy pro různé typy identifikačních údajů tak, aby při jejich rozpoznávání mohly využít jejich omezené abecedy (viz Tab. 2). Architektura neuronové sítě však zůstává takřka stejná, mění se jen rozměry vstupní a výstupní vrstvy. Vstupní vrstva je závislá na rozměru obrazu, který model přijímá, počet pod-klasifikátorů v rámci výstupní vrstvy je dán maximálním podporovaným počtem znaků a počet neuronů v každém z pod-klasifikátorů je v rámci konkrétního modelu dán velikostí jeho podporované abecedy.

Tab. 2: Modely neuronové sítě používané v závislosti na typu identifikačního údaje

Název modelu	Rozměr vstupního obrazu	Maximální počet znaků	Počet podporovaných znaků	Sada podporovaných znaků (abeceda)
Velko-písmenný model	512x64x1	44	69	0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZokrčpÁÄÉÍÓÖŐÚÜŮŰÝČĎĚĽŇŘŠŠŤŽ[*Mezera*],-./[*None*]
Numerický model	128x64x1	11	13	0123456789./[*None*]
Model pro křestní jména	384x64x1	32	118	ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyzÁÂÇÉÍÖÚÜáâäçéëíîóôöúüýăĆćČčĎďĐđĚĹĺĽľŁłŃńŐőŕŘřŚśŞşŠšţŤťůŰűźżŽž[*Mezera*]'-.[*None*]

Pro všechny číselné údaje jako je číslo, datum vydání a platnost dokladu a datum narození a rodné číslo držitele je použit numerický model. Model pro křestní jména je naopak úzce zaměřen na odpovídající údaj ve starší verzi občanského průkazu, ve kterém je jméno držitele vytištěno s ohledem na malá a velká písmena. Nejobecněji je využíván velko-písmenný model, pomocí kterého jsou rozpoznávány údaje jako příjmení, rodné příjmení, pohlaví, místo narození a trvalý pobyt držitele, na občanském průkazu nového typu navíc také jméno a státní občanství a v cestovním pasu kód země. Ve velko-písmenném modelu jsou navzdory jeho názvu zahrnuty i čtyři malá písmena „okrčp“ tak, aby model mohl pokrýt i zkratky typu „okr.“ a „č.p.“ označující okres a číslo popisné v údaji o trvalém pobytu a místě narození držitele.

Údaje, které svou abecedou nespadají ani do jednoho z modelů nejsou prozatím pro rozpoznávání podporovány. Mezi tyto údaje patří označení úřadu, který doklad vydal, rodinný stav a titul držitele a na starším typu občanského průkazu navíc jeho státní občanství.

5.2.2 Rozpoznávání strojově čitelné oblasti

Protože bylo rozpoznávání strojově čitelné oblasti řešeno o dva a půl roku dříve než rozpoznávání řádek textu z minulé kapitoly a byly pro něj v tu dobu dostupné jen omezené výpočetní prostředky v podobě osobního laptopu, je k dané problematice přistupováno tradičním způsobem rozpoznávání textu po jednotlivých znacích. Vzhledem k tomu, že byla však strojově čitelná oblast pro účely rozpoznávání textu separací jejích znaků speciálně navržena, není její předzpracování nikterak náročné.

Následující algoritmy zabývající se binarizací a separací znaků řádek strojově čitelné oblasti jsou implementovány v jazyce Matlab od společnosti MathWorks [14]. Neuronová síť je pak implementována pomocí softwarové knihovny Caffe [118], respektive její grafické nadstavby DIGITS [119] od společnosti NVIDIA.

5.2.2.1 BINARIZACE ŘÁDEK STROJOVĚ ČITELNÉ OBLASTI

Aby mohly být jednotlivé znaky strojově čitelné oblasti jednoduše separovány, bude s jejími řádky operováno v jejich binarizované podobě.

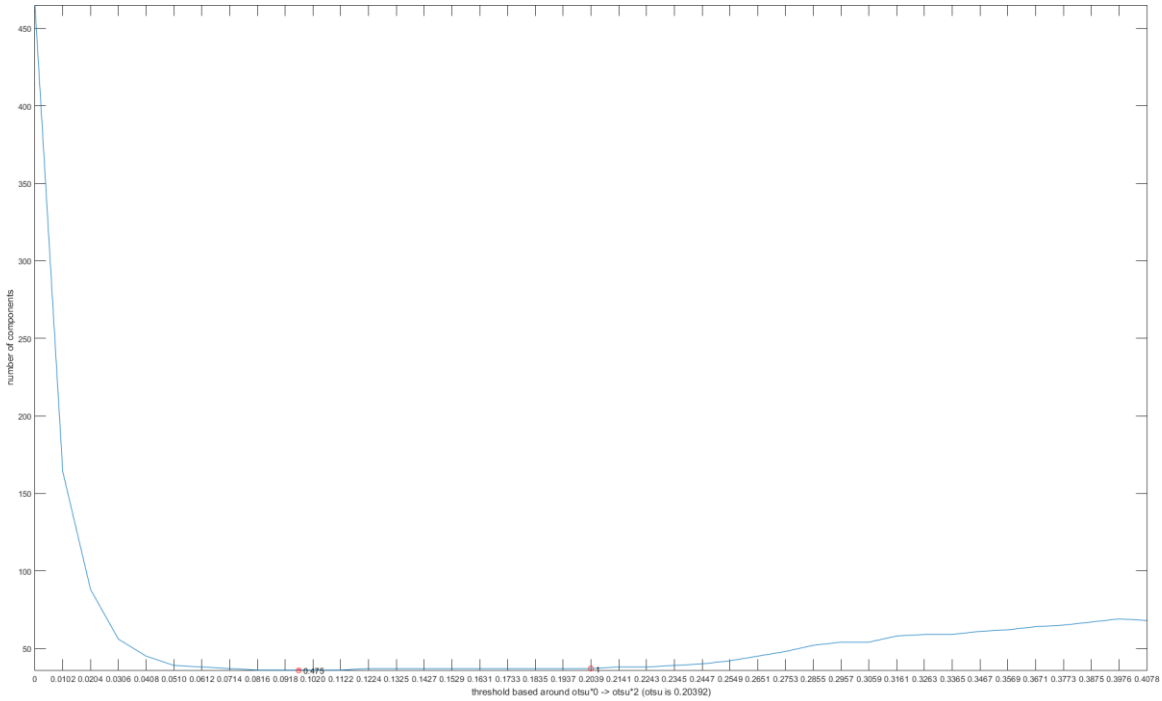
Ačkoliv by bylo možné pracovat s binarizovanými řádky pořízenými již v předešlé fázi lokalizace textu, je výhodné je oprahovat znovu a samostatně. Binarizace nad celým dokladem byla totiž ovlivněna výběrem prahu na základě kompletního snímku, je-li ale obdobnou technikou binarizován pouze samostatný řádek, je díky přesnějšímu určení prahu zaručen i o něco kvalitnější výsledek. Pro proces binarizace samostatných řádek je navíc možné využít zvětšení ořezu, které taktéž dopomáhá ke kvalitnějšímu výsledku. Každý řádek je proto uměle zvětšen pomocí lanczos3 interpolace na výšku 300 bodů při zachování poměru jeho stran.

Protože ne vždy je v procesu adaptivního prahování určení hodnoty prahu Otsuovou metodou optimálním řešením (viz kapitola 5.1.1.2), byla vynalezena technika, která se pokusí vyhledat ideální parametr pro práh automaticky na základě změn binarizovaného řádku napříč prahy různých hodnot.

Obr. 71: Vliv zvyšování prahu masky na obraz řádky textu

Postup je podobný jako v případě binarizace kompletní fotografie, pro binarizaci samostatných řádek je ale namísto aritmetického průměru pro obě průměrování použita Gaussova křivka a konstanta C je nastavena na nulovou hodnotu. Po získání rozostřené podoby rozdílového snímku je opět provedeno volání Otsuovy metody, nyní se však získaná hodnota uloží stranou. V následujících krocích algoritmus vyzkouší 20 různých hodnot prahů založených na uschované Otsuově hodnotě, počínaje nulou a dvojnásobkem uložené hodnoty konče. Po každé zkoušce je ve vzniklém obrazu sečten počet vzniklých spojených komponent, čímž lze vytvořit funkci závislosti jejich počtu na hodnotě prahu (viz Obr. 72).

Funkce zachycená na obrázku 72 má pro každý řádek textu podobný průběh. Tato podobnost je dána tím, že při hodnotě prahu nula je obraz zrnitý. To je způsobeno přirozeným chováním průměrování v klouzavém okně na pozici pozadí textové řádky (viz Obr. 57 vlevo na str. 62 a Obr. 71 nahoře). Se zvyšováním hodnoty prahu se však zpřísňuje i maska pro odstranění šumu (viz Obr. 59 vlevo nahoře na str. 63) a mění tak svou podobu z převážně bílé na převážně černou. Tím je odstraňováno čím dál tím více šumu a křivka funkce tak klesá až do takových hodnot, kdy je většina či všechen šum odstraněn (viz Obr. 71 uprostřed). Funkce se tak dostává do svého teoretického globálního minima. S dalším zvyšováním prahovací hodnoty se totiž začnou vlivem příliš přísné masky postupně rozpadat znaky textové řádky. To se v průběhu funkce projeví opětovným stoupáním počtu komponent (např. když se na Obr. 71 dole písmeno R a K rozdělí vedví). Ideálními hodnotami pro prahování lze proto prohlásit takové parametry, které tvoří v zaznamenané funkci minima (viz Obr. 72).



Obr. 72: Funkce závislosti počtu spojených komponent na hodnotě prahu (vybrané hodnoty označeny červeným zakroužkováním)

V případě velmi nekvalitního snímku řádky textu je možné, že je v průběhu funkce nalezeno více lokálních minim současně. V takové situaci algoritmus umožňuje binarizovat řádky pomocí více než jednoho řešení a na výstup pak posílá hned několik binarizačních kandidátů. Pro binarizaci jsou vybrána tato minima:

- Všechna lokální minima, která předcházela minimu globálnímu
- Globální minimum

- Je-li přítomno více globálním minim (stejné hodnoty), algoritmus zahrne i ty

Navíc je přidán i původní parametr Otsuovy metody, avšak jen tehdy, splňuje-li všechny následující podmínky:

- Tvoří vůči ostatním minimům extrém v x-ové souřadnici (ostatní minima jsou všechna na levé nebo pravé straně)
- Je vzdálený minimálně 3 kroky od nejbližší minima
- Obsahuje ve svém bodě jiný počet komponent než nejbližší minimum

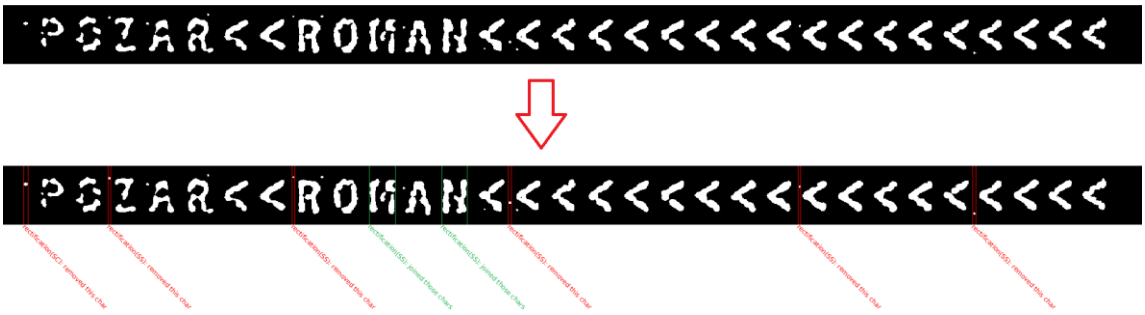
5.2.2.2 SEPAROVÁNÍ ZNAKŮ STROJOVĚ ČITELNÉ OBLASTI

Po kolekci binarizačních kandidátů řádek textu ze strojově čitelné oblasti následuje fáze jejího rozdělení na jednotlivé znaky. Pro tento účel byla s přihlédnutím na pevnou šířku písma OCR-B [82] použita metoda vertikální projekce obrazu.

Pro každý sloupec obrazu je nutné vyhodnotit, jestli obsahuje alespoň jeden bílý bod. Pokud ano, zapíše se do pomocného jednorozměrného pole o délce šířky obrazu jednička, v opačném případě je zapsána nula. Takto jsou postupně vyhodnoceny všechny sloupce obrazu řádky textu.

Probíhala-li by separace znaků na základě vertikální projekce triviálním způsobem, konečný krok by pak ohraničil místa, která obsahují nepřetržitou sérii jedniček za znaky a místa se sériemi nul jako mezery mezi nimi. Takový postup by však fungoval pouze pro vstupy bez jediné známky šumu a rozdvojených znaků, což nelze pro méně kvalitní fotografie dokladu zaručit. Byl proto navržen algoritmus, který se s případným šumem a rozdvojenými znaky dokáže vypořádat. Jeho popisu se věnují následující odstavce.

Na základě analýzy původního pomocného pole je určeno, jakou šířku mají série po sobě jdoucích jedniček a jakou šířku mají série po sobě jdoucích nul. Následně jsou pro oba druhy šířek určeny i střední hodnoty. Medián byl zvolen pro jeho odolnost vůči odlehlým hodnotám. Střední hodnota mezer (nul) je pak postupně porovnávána se všemi přítomnými mezerami. Vzhledem k tomu, že je písmo ORC-B neproporcionální, měly by mezi písmeny být mezery přibližně stejných šířkových hodnot. Nalezne-li algoritmus mezeru, která je nezvykle úzká, rozhodne se na základě porovnání okolních hodnot mediánů (předchozí a následující mezera a pravý a levý znak) pro spojení dvou sousedících znaků (viz zelené orámování na Obr. 73), nebo pro vymazání jednoho z nich (viz červené orámování na Obr. 73). Algoritmus podobným způsobem zanalyzuje i nezvykle malé znaky. Znak buď promění v mezeru, nebo jej spojí se znakem sousedícím.



Obr. 73: Oprava separace znaků na základě střední hodnoty šířky znaků a mezer

Vyskytne-li se ve zpracování řádky textu sporná situace, která nelze pomocí mediánu s jistotou správně rozuzlit, přichází na řadu jeho hlubší analýza. Jelikož je separace znaků prováděna implicitně jen na prvním z binarizačních kandidátů, zavolá funkce sama sebe s kandidátem následujícím. Ke zpracování ovšem předloží jen část řádky textu, kterou nedokázala vyřešit. Rekurzivně je tak nalezeno řešení například pro spojené znaky či pro znak napojený na ložisko šumu (viz modré orámování na Obr. 74). Metoda využívá toho, že každý další kandidát binarizace obsahuje menší množství šumu a tedy i menší rozměr znaků. Pokud se problém nepodaří vyřešit rekurzí, použije algoritmus neuronovou síť pro rozpoznávání znaků (viz následující kapitola) a rozhodne o separaci znaku na základě toho, ve kterém místě má síť pro daný ořez největší odezvu. Zbytek prostoru pak algoritmus označí jako mezeru. Testovány jsou celkem 3 různé pozice ořezu o 3 různých šířkách a dohromady je tak z rozpoznávání porovnáváno 9 výsledků.

Obr. 74: Rekurzivní řešení slitých znaků pomocí dalších binarizačních kandidátů

5.2.2.3 Rozpoznávání znaků strojově čitelné oblasti

Pro rozpoznávání jednotlivých znaků strojově čitelné oblasti dokladu je využívána konvoluční neuronová síť.

Model použité neuronové sítě, který má celkem 5 vrstev, je uveden vstupní vrstvou přijímající binarizovaný obraz o rozměrech 64x80x1.Vstupní vrstva je následně napojena na sérii třech po sobě jdoucích konvolučních vrstev s 20, 50 a 100 filtry o shodných rozměrech 5x5 a kroky o velikosti 1. Každá z těchto konvolučních vrstev je následována max-pooling vrstvou s filtrem o velikosti 2x2 a krokem 2 a postupně tak dochází k redukování množství informací, které jsou sítí přenášeny. Výstup z poslední max-pooling vrstvy je plně propojen s vrstvou o 500 neuronech, která již představuje poslední skrytou vrstvu celého modelu. Výstupní vrstvu, která je rovněž plně propojena, pak tvoří celkem 37 neuronů, přičemž každý z nich udává pravděpodobnost výskytu jednoho konkrétního znaku, který reprezentuje. Sada znaků, kterou je síť schopna na svém vstupu rozlišit, je dána abecedou strojově čitelné oblasti a obsahuje tedy velká písmena anglické abecedy, číslice a speciální oddělovací znak „<“ (menší než).

Vstupní obraz je zvětšen nebo zmenšen bez zachování jeho proporcí tak, aby vyhověl velikosti vstupní vrstvy neuronové sítě, a je poté ještě standardizován odečtem aritmetického průměru a podílem standardní deviace. Aritmetický průměr i standardní deviace byly předem vypočítány přes vzory celé trénovací množiny a obě hodnoty jsou proto nyní již konstantní.

Jako aktivační funkce je pro všechny neurony, kromě těch ve vstupní a výstupní vrstvě, použita ReLU funkce (viz kapitola 2.2.1.1) a proces učení neuronové sítě jako celku byl proveden standardní metodou Stochastic Gradient Descent s dávkami o velikosti 100 vzorů na jednu změnu vah (viz kapitola 6.2).

5.3 Korekce rozpoznaného textu

Protože pro většinu identifikačních údajů platí, že existuje jen konečné množství předvídatelných hodnot, kterých mohou nabývat, lze této skutečnosti využít v závěrečném zpracování rozpoznaného textu a provést na jejím základě jeho validaci, případně i korekturu. Algoritmy, které byly pro tento účel vytvořeny, budou popsány v následujících podkapitolách.

5.3.1 Korekce textu mimo strojově čitelnou oblast

Aby bylo možné rozpoznaný text mimo strojově čitelnou oblast dokladu validovat a případně opravovat, musí mít algoritmus přístup ke slovníku, který pro daný typ identifikačního údaje obsahuje co možná nejúplnější výčet všech hodnot, kterých může nabývat. Z webových stránek Ministerstva vnitra České republiky [120] a Katastru nemovitostí [121] byly proto shromážděny, zpracovány a uloženy informace o jménech a příjmeních osob žijících na území České republiky a o názvech českých ulic, obcí a okresů tak, aby svou formou odpovídaly

údajům v osobních dokladech. Pro údaje o jménech a příjmeních byla navíc zaznamenána i četnost jejich výskytu (viz Tab. 3).

Tab. 3: Přehled slovníků používaných k validaci a opravám rozpoznaných údajů

Obsah slovníku	**Počet údajů**	**Odpovídající řádky v osobním dokladu**	**Údaj o četnosti výskytu**
Jména	69 536	Jméno držitele	Ano
Příjmení	284 592	Příjmení držitele	Ano
Názvy okresů	92	Místo narození 2. řádek Trvalý pobyt 2. nebo 3. řádek	Ne
Názvy obcí	5 341	Místo narození 1. řádek	Ne
Názvy obcí včetně jejich částí	25 468	Trvalý pobyt 1. řádek	Ne
Názvy ulic (bez čís. označení)	27 135	Trvalý pobyt 1. nebo 2. řádek	Ne
Pohlaví	2	Pohlaví držitele	Ne
Státní občanství	544	Státní občanství držitele	Ne
Kódy zemí	273	Kód země Místo narození 1. řádek	Ne

Validace a případná korekce rozpoznaného textu pracuje tak, že je nejprve na základě typu zpracovávaného identifikačního údaje vybrán patřičný slovník (výběr je ovlivněn i rozložením řádek dokladu, viz kapitola 4.1). V něm je následně provedeno vyhledání rozpoznaného textu. Pokud je rozpoznaný text nalezen, je označen jako ověřený a další zpracování již není potřeba. V opačném případě je proveden pokus o jeho opravu tak, že je ve slovníku nalezena taková podmnožina údajů, která má k hledanému textu nejmenší editační vzdálenost. Ta však nesmí překročit maximální hodnotu v podobě poloviny délky hledaného údaje, čímž je zabráněno příliš velkým zásahům do rozpoznaného textu. Pokud taková množina údajů existuje a obsahuje jen jednu položku, je jí rozpoznaný text nahrazen. Pokud taková množina neexistuje, je rozpoznaný text označen jako neověřený a žádné další zpracování se již neprovádí. Existuje-li pro dané hledání více řešení a množina tak obsahuje více prvků, je každý její prvek porovnán s původním výstupem neuronové sítě a vybrán k nahrazení ten, který by síť označila po svém prvotním rozpoznání jako druhý nejpravděpodobnější.

Protože je validace a korekce údajů řešena pomocí předem připraveného slovníku, který může obsahovat znakovou sadu nad rámec použitého modelu neuronové sítě, mohou

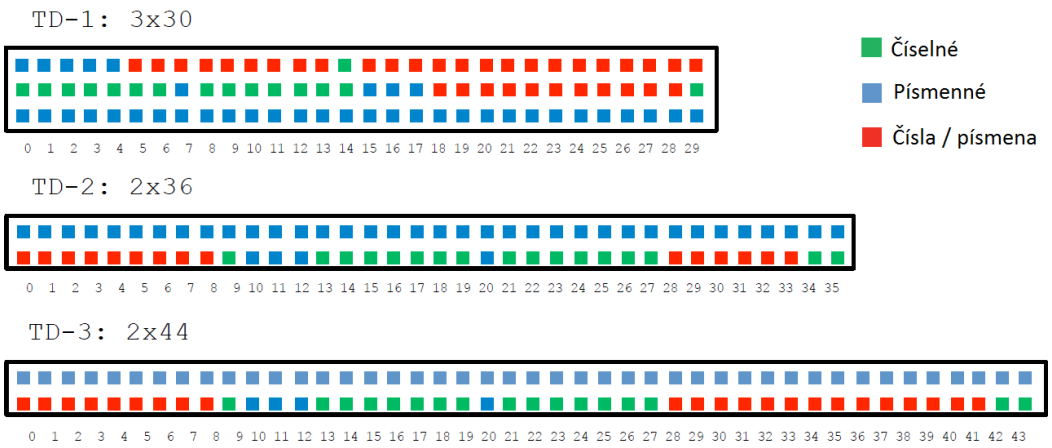
díky tomu výstupy systému obsahovat i znaky, které původně nebylo plánované podporovat. Pokud tato situace nastane a pro opravu navíc existuje více řešení, nelze je kvůli znakům nad rámec rozpoznávané abecedy porovnat na výstupu neuronové sítě tak, jak to bylo popsáno v předchozím odstavci. Algoritmus se v tomto případě o nejpravděpodobnější opravě rozhodne na základě četnosti výskytu daných údajů, jsou-li k dispozici (viz Tab. 3).

Editační vzdálenost je určena Levenshteinovou vzdáleností [122], jejíž hodnota je pro dva řetězce dána minimálním množstvím jedno-znakových úprav v podobě vložení, mazání a substituce tak, aby po jejich aplikaci byly oba řetězce shodné.

Informace ze strojově čitelné oblasti dokladu nejsou v současném stavu řešení do procesu validace a korekce rozpoznaných údajů zapojeny a jakékoliv číselné údaje, jejichž hodnoty nelze předem určit, proto do tohoto zpracování prozatím nespadají.

5.3.2 KOREKCE STROJOVĚ ČITELNÉ OBLASTI

Algoritmus validace a případné korekce strojově čitelné oblasti dokladu je úzce svázán s výstupy neuronové sítě, která byla pro rozpoznávání jejích znaků použita. Pro každou jedno-znakovou pozici ve strojově čitelné oblasti je zaznamenána jedna sada 37 výstupů, jež vypovídají o pravděpodobnostech výskytu každého ze znaků podporované abecedy. Celkový součet těchto pravděpodobností pro jednu takovou pozici je vždy roven hodnotě 1. Na různých pozicích řádky pak lze očekávat různě omezenou znakovou abecedu, proto je na výstup neuronové sítě použita maska (viz Obr. 75). Ta vynuluje ty pravděpodobnosti znaků, které nejsou pro danou pozici v řádku vhodné. Ze zbylých pravděpodobností je pak vybrán ten znak, který má nejvyšší pravděpodobnostní ohodnocení.



Obr. 75: Struktura různých typů strojově čitelných oblastí dle dokumentu ICAO 9303 [83] (převzato z [123])

Strojově čitelná oblast dokladu obsahuje kromě samotných identifikačních údajů i kontrolní číslice, na základě kterých lze ověřit, zda přečtení patřičné informace proběhlo správně. Toto ověření je implementováno i v popisovaném algoritmu s tím, že nekoresponduje-li kontrolní číslice s vyčtenými hodnotami, je algoritmus schopen situaci napravit. Napravení probíhá pomocí náhledu do záznamu pravděpodobností klasifikace jednotlivých znaků. Na místech, kde se ve výstupu neuronové sítě vyskytuje více znaků s podobnou pravděpodobností, je tato skutečnost zaznamenána. V následujícím kroku je vytvořena množina všech možných kombinací nejistých znaků a podle jejich celkové pravděpodobnosti ve zřetězení jsou postupně od nejvyšší po nejnižší ověřovány do té doby, než se shodují s požadovaným kontrolním číslem. Řetězec znaků, který této podmínce vyhověl, je použit jako náhrada původního rozpoznání. Do tohoto procesu je zahrnuta i samotná kontrolní číslice.

6 IMPLEMENTACE

Kapitola implementace přímo navazuje na popsané řešení z kapitoly předchozí tak, aby dále rozvinula implementační detaily některých prezentovaných postupů. Pozornost bude přitom stále zaměřena na samotný serverový modul pro rozpoznávání identifikačních údajů z osobních dokladů, respektive na serverovou část systému.

6.1 MODELY PRO ROZPOZNÁVÁNÍ CELÝCH ŘÁDEK TEXTU

Tato kapitola je věnována implementaci modelů pro rozpoznávání celých řádek textu mimo strojově čitelnou oblast dokladu, které byly popsány v kapitole 5.2.1, respektive v kapitole 5.2.1.2.

6.1.1 GENERÁTOR SYNTETICKÝCH DAT

Konvoluční neuronové sítě ke svému učení potřebují obrovské množství trénovacích dat. Není přitom výjimkou, že se jejich počet, nutný ke zdárnému naučení daného problému, pohybuje v řádech milionů [38, 47, 68]. Aby byl takový objem vzorů složen čistě na základě reálných dat, často není proveditelné, a ne jinak tomu bylo i při učení modelu pro rozpoznávání řádek textu osobních dokladů. Podobně jako v pracích Gupta et al. [59] a Jaderberg et al. [68] (viz kapitola 3) byl pro účel získání potřebného množství trénovacích dat vytvořen program, který je schopen je uměle vytvářet.

Úkolem generátoru syntetických dat je vytvářet obrazy s textem takovým způsobem, aby se jejich obsah co nejvíce podobal řádkám textu z osobních dokladů na reálných fotografiích. Generátor proto musí kromě samotného renderování textu umět i simulovat fotografie s nízkou kvalitou obrazu. Výsledné obrazy je poté nutné uložit do patřičného úložiště dat spolu se záznamem o jejich textovém obsahu tak, aby je následně bylo možné spárovat.

Pro implementaci generátoru syntetických dat byl zvolen programovací jazyk Matlab od společnosti MathWorks [14], a to zejména pro jeho bohatou nabídku předpřipravených funkcí a procedur pro práci s obrazovými daty.

6.1.1.1 POPIS ALGORITMU GENERÁTORU

Vstup generátoru je tvořen několika parametry, které musejí být součástí požadavku na jeho spuštění. Mezi tyto parametry patří označení adresáře, kam bude generátor vytvořené obrazy

ukládat, určení počtu obrazů k vygenerování, určení jejich požadované šířky a výšky a odkaz na funkci pro generování textových řetězců.

Aby syntetická data byla co možná nejrozmanitější, používá algoritmus generátoru na všech místech, kde je to vhodné, generátor náhodných čísel, na základě jehož výstupu rozhoduje, jakým způsobem bude výsledný obraz vytvářet.

Proces vytvoření jednoho obrazu vypadá následovně. Nejprve je náhodně vybrán jeden z pěti předem připravených obrazů, který bude sloužit jako pozadí pro renderovaný text. Všech pět obrazů přitom bylo připraveno tak, aby odpovídaly obrazům na pozadí podporovaných osobních dokladů (viz Obr. 76). Následně je provedeno volání funkce pro vygenerování textového řetězce, která byla součástí vstupních parametrů generátoru, jejíž výstup je pak zaznamenán. Funkce pro generování textového řetězce je z procesu vytváření umělého obrazu oddělena proto, aby ji bylo možné operativně měnit a aby generátor syntetických dat byl použitelný pro různé typy úloh bez jeho dalších úprav (viz kapitola 6.1.2.1). Algoritmus následně provede náhodný výběr mezi předem definovanými typy písma a jejich velikostmi. Kolekce typů písem byla sestrojena na základě analýzy podporovaných osobních dokladů (viz kapitola 4.1) a obsahuje tedy 6 různých písem. Kromě velikosti je náhodně určena i jejich tučnost a odstín šedi, kterým budou vykresleny (hodnoty 0 až 100). V dalším kroku již algoritmus renderuje zaznamenaný textový řetězec do vylosovaného obrazu pozadí. Text je samozřejmě renderován pomocí písma a jeho vlastností tak, jak bylo předem pro tento účel náhodně připraveno. Souřadnice pro jeho vykreslení jsou voleny opět náhodně. Proces poté pokračuje sérií augmentačních operací, které mají za úkol ve výsledném obrazu simulovat různé kvalitativní nedostatky. Tyto operace budou popsány v samostatné kapitole.

Obr. 76: Různá pozadí českých osobních dokladů použitá pro renderování vygenerovaného textového řetězce

Výstup generátoru je tvořen hotovým vygenerovaným obrazem řádky textu o definovaných rozměrech a řetězcovou hodnotou, která byla použita pro renderování jeho obsahu. Obraz je ve formátu PNG uložen do adresáře určeného vstupním parametrem a řetězcová hodnota spolu s informací o cestě k uloženému obrazu přidána do CSV souboru. Obrazy i CSV záznamy jsou tvořeny kontinuálně do té doby, než je dosažen požadovaný počet vygenerovaných obrazů. Soubory obrazů jsou pojmenovávány číslovkami a rozdělovány po 10 tisících záznamech do oddělených adresářů. Oddělování bylo implementováno z toho důvodu, že přístup do adresářů se sta tisíci až miliony souborů činí běžným souborovým i obrazovým prohlížečům problémy a nebyly za těchto okolností použitelné. Algoritmus byl implementován tak, aby jej bylo možné kdykoliv přerušit a při příštím spuštění automaticky navázal na svou předešlou činnost.

Protože se nároky na míru snižování kvality obrazu pro modely s různou abecedou znaků liší, byly vypracovány dvě varianty generátoru syntetických dat. První varianta je zaměřena na silnější snižování kvality obrazu a je určena pro číselný model neuronové sítě (viz Tab. 2 na str. 74), zatímco druhá varianta již takovou měrou kvalitu obrazu nesnižuje a je určena pro modely s širší abecedou obsahující znaky s diakritickými znaménky. Druhá varianta generátoru je používána proto, aby byly drobné detaily právě v podobě diakritických znamének ve výsledném obraze stále ještě rozlišitelné. Výstupy z obou variant generátoru je možné pozorovat na obrázku 77.

Obr. 77: Příklady výstupů generátoru syntetických dat

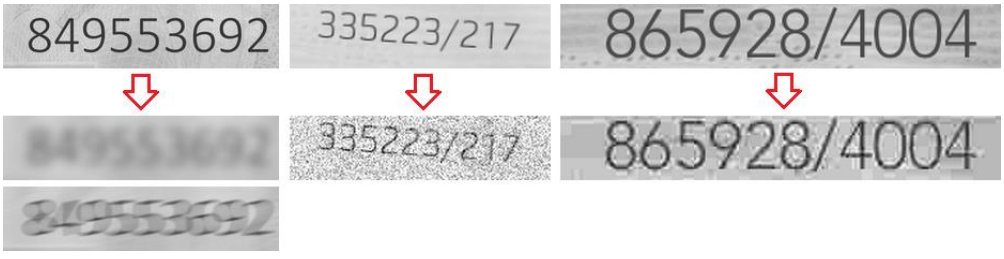
6.1.1.2 AUGMENTACE GENEROVANÝCH DAT

Aby syntetická data byla co možná nejrozmanitější a zároveň se co nejvíce podobala řádkám textu z reálných fotografií osobních dokladů, je na ně napříč jejich tvorbou aplikováno mnoho různých obrazových úprav. Rozsah těchto úprav, stejně jako jejich samotné užití je přitom ponecháno náhodě. Výsledný obraz tak může být úpravou daného typu ovlivněn markantně, nepatrně, anebo vůbec. Následující obrazové operace budou postupně uváděny v takovém pořadí, v jakém jsou skutečně prováděny.

Mezi základní použité obrazové operace patří náhodná úprava jasu obrazu na pozadí (aplikována ještě před renderováním textu), náhodná rotace vykreslené řádky textu a ořez textu s náhodnou velikostí okrajů. Rotace řádky je omezena +-4 stupni a velikost okrajů může nabývat i mínusových hodnot, přičemž je pak část textu ořezem vynechána.

Rozostření, šum a JPEG komprese

Jakmile je řádek textu vyříznut, čeká na něj série třech dalších operací. Každá z nich přitom po své aktivaci vrací na svém výstupu procentuální údaj, který naznačuje, jakou měrou přispěly ke snížení kvality vytvářeného obrazu. Tento údaj je používán pro kontrolování rozsahu snižování kvality obrazu tak, aby celkový součet po aplikaci jednotlivých operací nepřesáhl mez 140%. To má za následek to, že když například operace šumu „poškodí“ obraz na 100% svých možností, zbylé dvě operace mohou operovat již jen se zbylými 40% tak, aby výsledný obraz byl stále ještě čitelný. Aby každá z těchto třech operací mohla do vytvářeného obrazu zasáhnout za stejných podmínek, je jejich volání prováděno v náhodném pořadí. Operace rozostření je navíc na základě náhody provedena buď ve formě Gaussovského, nebo pohybového rozostření. Výstupy všech třech operací při jejich 100% zásahu do vytvářeného obrazu je možné pozorovat na obrázku 78.



Obr. 78: Operace Gaussovského rozostření (vlevo uprostřed), pohybového rozostření (vlevo dole), přidání šumu (uprostřed) a JPEG komprese (vpravo)

Úprava rozměrů obrazu

Do tohoto okamžiku se s obrazem pracovalo v takových rozměrech, v jakých vznikl pomocí renderování textu písma dané velikosti. Nyní je však již obraz upraven tak, aby odpovídal

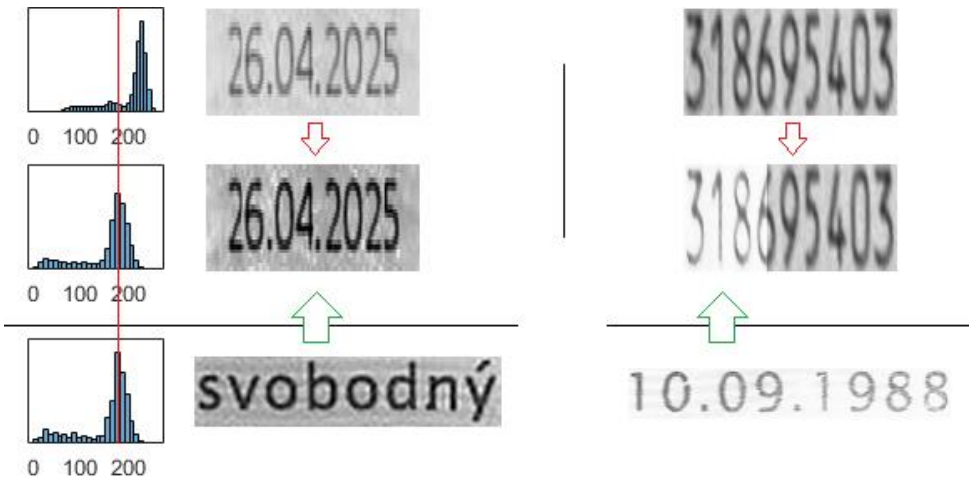
požadovaným rozměrům, které byly určeny vstupním parametrem generátoru. Algoritmus pro úpravu velikosti obrazu je volen opět náhodně, přičemž výběr je omezen na interpolaci pomocí nejbližšího souseda, Lanczos3, bikubickou a bilineární interpolaci.

Ekvalizace histogramu

Další operace augmentace generovaných dat mění kontrast vytvářeného obrazu. Změny kontrastu jsou založeny na ekvalizaci histogramů dvou obrazů. První histogram patří obrazu, který je generován, zatímco druhý histogram je odvozen od některé řádky textu z reálné fotografie osobního dokladu. Operace ekvalizace histogramu pak provádí na generovaném obrazu takové úpravy, aby histogram výsledného obrazu byl přibližně stejný, jako cílový histogram odpovídající reálné fotografii (viz Obr. 79 vlevo). Díky této úpravě je generovaný obraz svým kontrastem připodobněn obrazu reálnému.

Aby generátor syntetických dat nemusel pracovat s osobními údaji, které se na řádkách textu z reálných fotografií vyskytují, a aby nemusel výpočet jejich histogramů být prováděn opakovaně, jsou jednotlivé histogramy generátoru poskytnuty v podobě předzpracované konstantní kolekce. Histogramy jsou odvozeny z 1000 obrazů řádek textu, které utvářejí validační množinu pro učení neuronové sítě (viz kapitola 6.1.2.2) a jsou vybírány náhodně.

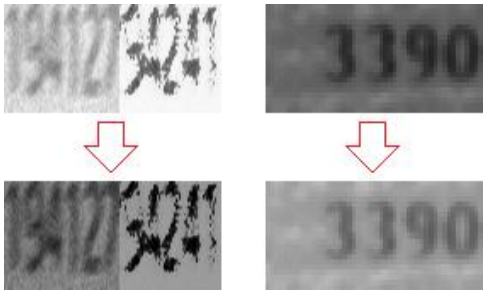
Ekvalizace histogramu je zároveň čas od času použita pouze tak, aby její výstup ovlivnil vytvářený obraz jen z části (viz Obr. 79 vpravo). Tato úprava adresuje různý jas některých částí řádek textu reálných fotografií v důsledku přítomnosti ochranného prvku v osobním dokladu (viz kapitola 4.1).



Obr. 79: Ekvalizace histogramu generovaného obrazu na základě hodnot z reálných fotografií (pod čarou)

Úprava jasu

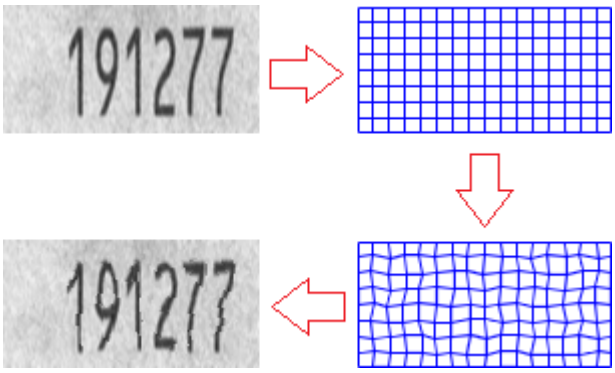
V dalším kroku je vytvářenému obrazu upraven jas. Jas je upravován plošně celému obrazu přičtením nebo odečtením náhodného čísla. Náhodné číslo je omezeno pomocí aritmetického průměru hodnot vstupního obrazu tak, aby byl výstupní obraz stále ještě čitelný. Pokud byla v předchozím kroku použita ekvalizace histogramu, je jas upravován bezpodmínečně. Je tak rozšířena omezená paleta výstupů, která je dána počtem předzpracovaných histogramů.



Obr. 80: Úprava jasu generovaného obrazu

Deformace obrazu

Poslední operaci, kterou generovaný obraz může podstoupit, je jeho deformace. Deformace obrazu byla implementována proto, aby přinesla renderovanému písmu daného typu drobné výkyvy ve vykreslení a byl tak do obrazu vnesen další náhodný prvek. Úprava obrazu deformací je implementována tak, že se na základě aktuálního obrazu vytvoří pravidelná mřížka, jejíž jednotlivé spojnicové body jsou následně posunuty v náhodném směru (viz Obr. 81). Obě mřížky jsou pak použity pro provedení takové transformační operace, která změny mezi originální a upravenou mřížkou promítne do výsledného obrazu. Rozlišení mřížky je přitom upraveno tak, aby pro každou velikost obrazu byla právě osm polí vysoká, přičemž její šířka je dána šířkou obrazu. Transformace je provedena náhodným výběrem mezi lineární metodou a metodou nejbližšího souseda.



Obr. 81: Deformace obrazu pomocí deformace mřížky

6.1.1.3 Optimalizace výkonu generátoru

Programy vytvořené v jazyce Matlab obecně nevynikají svou rychlostí běhu [124] a pro to, aby byl generátor syntetických dat použitelný, musel být jeho chod nejdříve optimalizován.

Pro optimalizaci byl využit profilovací nástroj prostředí Matlab. Po detailní analýze nejdéle trvajících částí algoritmu bylo zjištěno, že nejmarkantnějšího zlepšení je možné dosáhnout jedině úpravou interních skriptů Matlab knihoven. Kopie vybraných interních skriptů byly proto přeneseny do projektového adresáře, aby na nich byly provedeny patřičné optimalizační úpravy. Úpravy spočívaly zejména v odstranění validace vstupních parametrů některých funkcí a v přidání mezipaměti pro takové interní proměnné, které byly při procesu renderování textu zbytečně vypočítávány stále dokola. Algoritmus generátoru syntetických dat byl poté přesměrován na upravené verze funkcí, čímž se podařilo navýšit jeho výkon v rychlosti generování dat zhruba o 60%.

Dále byl ještě algoritmus paralelizován tak, aby byla do generování obrazů zapojena všechna jádra daného procesoru. Po popsaných úpravách bylo na procesu Core i5-7500 spolu s SSD diskem dosažena rychlost generování dat zhruba 500 tisíc obrazů za hodinu. Vytížení všech čtyř jader procesoru se při generování dat pohybuje v rozmezí 95-100%.

6.1.2 Příprava dat pro učení neuronové sítě

Tato kapitola bude věnována tomu, jak byly sestaveny trénovací, validační a testovací množiny pro všechny tři modely neuronových sítí pro rozpoznávání celých řádek textu.

Abeceda a podporované délky řádek textu byly předem zvoleny na základě analýzy dat slovníků, které byly popsány v kapitole 5.3.1.

6.1.2.1 Syntetická data

Syntetická data jsou stěžejním prvkem v realizaci všech třech modelů neuronové sítě. Bez nich by nikdy nebylo možné uspokojit vysoké datové nároky tak velkého konvolučního modelu.

Pomocí generátoru syntetických dat (viz kapitola 6.1.1) byla pro každý ze třech modelů vytvořena trénovací a validační množina. Trénovací množiny obsahovaly 4 miliony vzorů a validační množiny 20 tisíc. Trénovací množina velko-písmenného modelu byla později ještě rozšířena na celkových 10 milionů vzorů (viz kapitola 6.1.3.3).

Aby bylo možné generátor syntetických dat použít, bylo nejprve nutné sestrojit funkci pro generování řetězcových hodnot, kterou generátor předpokládá na svém vstupu. Tato funkce by mohla být implementována tak, aby byly textové řetězce pro každý z modelů na

základě jejich znakových sad generovány zcela náhodně. Protože se však v řádkách textu osobních dokladů vyskytují určité vzorce, které jsou zcela neměnné (např. malá písmena „k“, „o“ a „r“ jsou v trvalém pobytu použita výhradně v podobě „okr.“ a označují okres), bylo rozhodnuto této pravidelnosti ve funkci pro generování řetězců využít. Toto rozhodnutí bylo založeno na předpokladu, že pokud budou v trénovacích vzorech tato pravidla obsažena, neuronová síť si na jejich základě vytvoří předpoklady, jež bude poté moci při rozpoznávání textu využít.

Pro numerický model proto funkce pro generování textových řetězců vytváří takové hodnoty, které připomínají formát data, rodného čísla a obecné řady čísel.

Pro velko-písmenný model funkce čas od času generovaný řetězec předsadí textem „okr.“ nebo zakončí řetězcem „č.p.“, po kterém následují číslice, které jsou někdy předěleny lomítkem, aby simulovaly číselné značení ulice trvalého pobytu. Tělo generovaného textu je tvořeno řadou dílčích řetězců oddělených mezerami, pomlčkami nebo tečkami tak, aby připomínaly jednotlivá slova.

Protože je model pro křestní jména velmi specifický, byl pro něj text na rozdíl od ostatních modelů vytvářen na základě slovníku. Slovník byl složen ze stejného obsahu jako slovník pro korekci rozpoznaných křestních jmen (viz kapitola 5.3.1).

Všechny jmenované funkce pro generování textových řetězců jsou implementovány tak, aby simulovaly i případné odseknutí části textu. Funkce pro numerický model tak netvoří například datum pouze ve formátu „24.12.2017“, ale i jako „24.12.20“.

Diskuze

Jakékoliv předpoklady o formátu řádek textu v osobních dokladech musejí být konstruovány opatrně, protože je-li některé z pravidel porušeno, neuronová síť se pak chová nepředvídaným způsobem. Příkladem může být prvotní mylný předpoklad, že řádek textu velko-písmenného modelu nemůže začínat číslicí. Některé názvy českých ulic totiž číslicí začínají (např. 5. května, 5 domků, atd.). Množina dat proto musela být vygenerována znovu a neuronová síť přeučena.

6.1.2.2 REÁLNÁ DATA

Aby bylo možné ověřovat, jak se jednotlivé modely neuronové sítě vypořádávají s řádky textu z reálných fotografií, bylo nutné kromě syntetických dat pro učení neuronové sítě vytvořit i množinu dat reálných. Pro tento účel byla využita kolekce 550 fotografií občanských průkazů a cestovních pasů, jež byla pro tuto práci k dispozici (viz kapitola 4.1).

Protože ani jedna z fotografií neměla předem přiřazenou informaci o jejím obsahu, musely být řádky textu z fotografií extrahovány a klasifikovány až dodatečně. Pro účel sběru reálných dat byla proto vytvořena aplikace v jazyce Matlab (viz Obr. 82), která má za úkol sběr řádek textu alespoň z části zautomatizovat. Aplikace je schopna na základě určení cesty k adresáři s fotografiemi jimi iterovat tak, aby byly všechny postupně zpracovány. V prvním kroku je načten vstupní obraz, který je pak možné rotovat v obou směrech a v případě potřeby (např. při více dokladech na jedné fotografii) určit i čtvercovou výseč, která omezí další zpracování výhradně na její obsah. V dalším kroku pak aplikace využívá již popsaný algoritmus lokalizace textu (viz kapitola 5.1), na základě jehož výstupu umožní iterovat extrahovanými řádky textu. Uživatel má pak za úkol jednotlivé obrazy řádek textu klasifikovat (přepsat jejich obsah do textového pole), aby pak byly uloženy do předem specifikovaného výstupního adresáře. Spolu s uložením obrazu řádky textu je do patřičného CSV souboru přidán záznam o jeho umístění a provedené klasifikaci.

Obr. 82: Grafické rozhraní aplikace pro manuální klasifikaci reálných dat

Pomocí vytvořené aplikace pro sběr reálných dat bylo zpracováno celkem 2019 řádek textu. Řádky byly poté rozděleny do dvou množin. První množina obsahuje 1000 řádek a slouží jako zdroj histogramů pro generátor syntetických dat (viz kapitola 6.1.1.2). Tato množina je zároveň sestavena jako druhá validační množina, aby při procesu učení určovala vedle syntetické množiny přesnost modelu i pro reálná data. Druhá množina o zbylých 1019 vzorech pak slouží pro všechny modely jako testovací množina.

6.1.3 Učení neuronové sítě

Protože zvolená architektura neuronové sítě pro rozpoznávání celých řádek textu obsahuje takové uspořádání výstupní vrstvy, které není běžné, není k učení ani jednoho z modelů možné použít interaktivní řešení jako například DIGITS [119] od společnosti NVIDIA. Pro učení neuronové sítě pro rozpoznávání celých řádek textu byla proto vytvořena vlastní aplikace.

Aplikace pro učení neuronové sítě je naprogramována v jazyce Python s pomocí softwarové knihovny TensorFlow [116]. Je zaměřena na učení s učitelem, automaticky předzpracovává trénovací, validační a testovací množiny a podporuje učení neuronových sítí s výstupními vrstvami o libovolném počtu pod-klasifikátorů. Podporována je i libovolná abeceda znaků a libovolný rozměr vstupních obrazů. Průběh učení neuronové sítě je monitorován jak pomocí výstupů do konzole, tak vykreslováním grafů. Mezi sledované veličiny patří vývoj chybové funkce, vývoj rychlosti učení a vývoj mnoha typů přesnosti rozpoznávání trénovacích a validačních dat (viz kapitola 6.1.3.2). Přesnost rozpoznávaných dat je navíc možné sledovat pro každou délku textu zvlášť. Při procesu učení neuronové sítě aplikace odhaduje čas jeho konce a je na základě analýzy vývoje její přesnosti schopna automaticky snižovat hyperparametr rychlosti jejího učení (angl. learning rate). Mezi podporované funkce patří i takzvaný fine-tuning, tedy schopnost navázat na předchozí učení při současné možnosti modifikace některých parametrů obnovované neuronové sítě, zobrazování náhodných nebo chybných klasifikací z validační nebo testovací množiny a generování matice záměn.

6.1.3.1 Volba architektury neuronové sítě

Řešení rozpoznávání celých řádek textu pomocí modelu, jehož popisu se věnovala kapitola 5.2.1.1, předcházelo mnoho experimentů s jinými architekturami neuronových sítí, které však nevedly k uspokojivým výsledkům.

Jako první bylo experimentováno s konvoluční neuronovou sítí o 5 konvolučních vrstvách následovaných dvěma plně propojenými vrstvami podobně, jako v práci Jaderberg et al. [68]. Tato architektura si však v rozpoznávání číslic, na kterém byla testována, nevedla tak dobře, jak bylo očekáváno a bylo od ní proto opuštěno.

Další experimenty byly již inspirovány neuronovou sítí VGG (viz kapitola 3.3.2). Tato architektura byla s úspěchem použita pro rozpoznávání objektů v obraze a dalo se proto předpokládat, že bude podávat dobré výsledky i pro rozpoznávání řádek textu. To se následně potvrdilo, když síť s 13 konvolučními vrstvami a filtry o velikosti 3x3 konečně začala podávat výborné výsledky v rozpoznávání číslic. V dalším kroku byla tato síť rozšířena pro velko-

písmenný model (viz Tab. 2 na str. 74). Pro jeho plnou podobu však učením nebylo možné dospět k použitelným výsledkům. Podporovaná délka textu velko-písmenného modelu byla proto z původních 44 znaků dočasně snížena na znaků 11. Při učení upraveného modelu bylo zjištěno, že neuronová síť podává lepší výsledky, pokud je navýšena její šířka v podobě počtu filtrů v jednotlivých skrytých vrstvách. I přesto však nebylo možné délku rozpoznávaného textu navýšit na více než 16 znaků bez toho, aby to mělo negativní vliv na přesnost jeho rozpoznávání.

Výrazného zlepšení přesnosti velko-písmenného modelu v jeho plném rozsahu bylo docíleno až s implementací neuronové sítě na základě práce He et al. [75], respektive ResNet modelu (viz kapitola 3.3.3). Pomocí residuální konvoluční neuronové sítě o 34 vrstvách získal konečně model dostatečnou kapacitu a podařilo jej naučit tak, aby podával velmi dobré výsledky i pro 44 znaků dlouhé řetězcové hodnoty. Experimentováno bylo i se sítí o 152 vrstvách, ta však pro rozpoznávání textu nepodávala tak dobré výsledky. Zhoršení výsledků bylo přičteno takzvané *bottleneck architektuře* (viz He et al. [75]), která byla použita výhradně pro tuto hlubší variantu sítě.

6.1.3.2 MĚŘENÍ PŘESNOSTI ROZPOZNÁVÁNÍ

Aby bylo možné v průběhu učení neuronové sítě hodnotit, jak úspěšně si při rozpoznávání celých řádek textu vede, byly zhotoveny tři typy určení její přesnosti. Jednotlivé typy přesností budou v následujících sekcích postupně popsány na příkladech jedné řádky textu s tím, že určování přesnosti nad celou množinou je pak vypracováno pomocí aritmetického průměrování.

Přesnost na úrovni znaku

Přesnost na úrovni znaku je založena na porovnávání celých vektorů očekávaného výstupu s výstupem, který síť vyprodukovala, a to včetně speciálního [None] znaku, který symbolizuje neobsazenou pozici v textu. Pro numerický model, který je určen pro textové řetězce délky až 11 znaků, má proto vektor pro hodnotu „123456789“ podobu „123456789[None][None]“. Je-li potom na výstupu očekávána například hodnota „12345“ a síť v rozpoznání této hodnoty selže a na svém výstupu vyprodukuje hodnotu „66666“, její přesnost na úrovni znaku bude stále po zaokrouhlení rovna 55%, protože správně odhadla délku textu a zbylých 6 [None] znaků se v obou vektorech shoduje.

Přesnost na úrovni znaku v rámci délky textu

Přesnost na úrovni znaku v rámci délky textu je určena podobně, jako přesnost na úrovni znaku, s tím rozdílem, že se do konečné přesnosti nezapočítávají překrývající se [None] znaky.

Pro stejný příklad, jaký byl uveden v popisu přesnosti na úrovni znaku, by tak přesnost činila 0%. Pokud by pro stejný model byl očekáván výstup „12345“ a skutečná hodnota na výstupu měla podobu „123456“, dosahovala by přesnost po zaokrouhlení 83%.

Přesnost na úrovni sekvence znaků

Přesnost na úrovni sekvence znaků je určena na základě rovnosti očekávaného a skutečného výstupu. Pokud je tedy očekávaný výstup roven „12345“ a výstup na síti je roven „92345“, přesnost se rovná 0%. Jakákoliv odchylka v těchto dvou hodnotách má tak za následek pro danou sekvenci kompletní selhání.

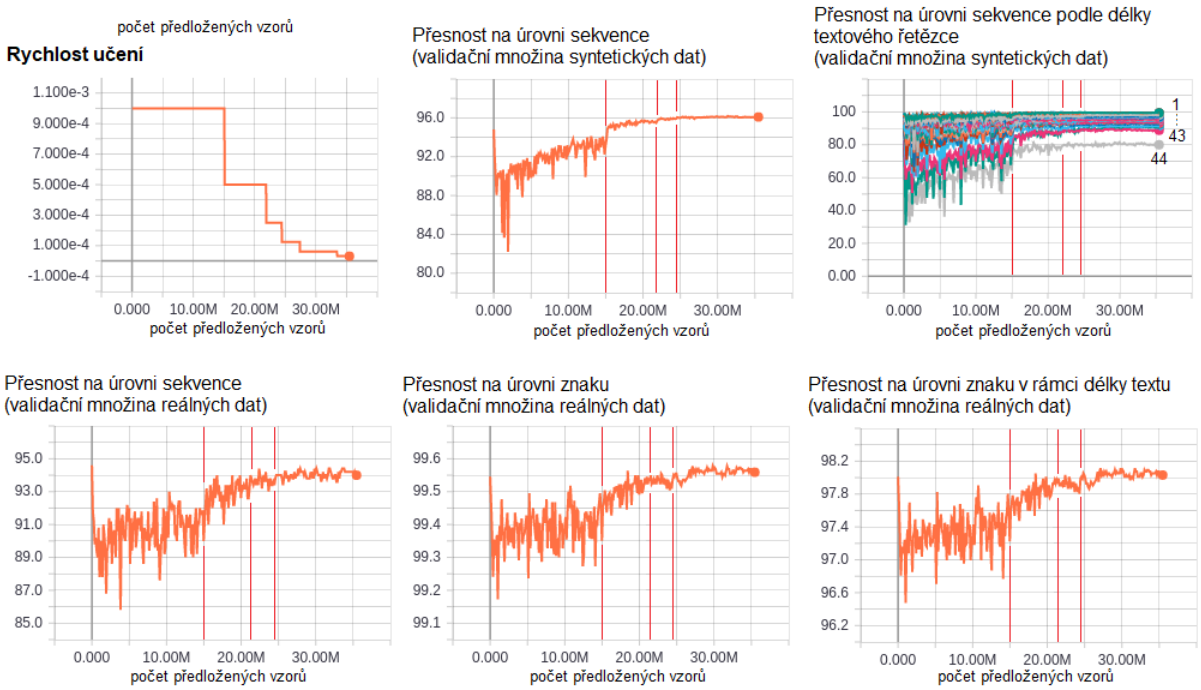
6.1.3.3 Proces učení neuronové sítě

Učení neuronové sítě bylo provedeno standardní metodou Stochastic Gradient Descent se zpětnou propagací chyby a setrvačností. Aby bylo možné učit neuronovou síť jako celek, musela pro ni však být sestrojena taková chybová funkce, která v sobě zohledňuje všechny paralelní plně propojené vrstvy v její výstupní vrstvě. Chybová funkce byla proto definována jako součet Cross Entropy chybových funkcí napříč všemi pod-klasifikátory. Inicializace vah byla provedena MSRA metodou [125].

Trénovací vzory byly během učení vybírány v rámci každé z epoch náhodně a každý vzor byl lokálně standardizován odečtem svého aritmetického průměru a podílem své standardní deviace. Tyto vzory pak již nebyly předmětem žádné další augmentační operace. Velikost dávky na jednu změnu vah (angl. batch size) činila 64 trénovacích vzorů. Během učení numerického a velko-písmenného modelu bylo experimentováno s L2 regularizací, nebyla však prozatím nalezena taková síla regularizace, aby jejím použitím modely na validační množině získaly vyšší přesnost, a tak byla z procesu učení prozatím vynechána. Vzhledem k množství vzorů v trénovací množině však všechny tři modely pracují uspokojivě i bez ní.

Prvotní rychlost učení byla stanovena na hodnotu 0.001. Ta byla pak během učení automaticky snižována o polovinu pokaždé, když nebylo po 2 miliony předložených trénovacích vzorů zaznamenáno zlepšení v kumulované přesnosti nad oběma validačními množinami. Kumulační přesnost se skládala ze součtu přesností na úrovni sekvencí a přesností na úrovni znaku (viz kapitola 6.1.3.2) pro validační množinu syntetických dat a validační množinu reálných dat (celkem 4 přesnosti). Jaký má vliv snížení rychlosti učení na přesnosti velko-písmenného modelu v závěru jeho učení (model byl učen na dvě etapy, viz dále) je možné pozorovat na obrázku 83. Analýza kumulovaných i dalších přesností byla prováděna po každých 128 tisících předložených trénovacích vzorech. Pokaždé, když bylo v rámci

analýzy zjištěno, že došlo ke zlepšení nejlepší dosažené kumulované přesnosti, byl aktuální model včetně jeho parametrů uložen pro případné pozdější použití.



Obr. 83: Vliv snížení rychlosti učení na různé typy přesností obou validačních množin při druhé etapě učení velko-písmenného modelu neuronové sítě

Během učení velko-písmenného modelu s trénovací množinou o velikosti 4 milionů vzorů bylo pomocí analýzy přesností zjištěno, že dochází k přeučování. Na trénovacích datech totiž model dosahoval takřka pro každou dávku vzorů 100% přesnosti, zatímco na validační množině již takový výkon nepodával. Učení bylo proto přerušeno a trénovací množina navýšena na 10 milionů vzorů. Po této úpravě pak síť při obnoveném učení podávala znatelně lepší výsledky.

Učení neuronové sítě probíhalo na grafické kartě NVIDIA GeForce GTX 1080 s grafickým jádrem Pascal, jejíž vytížení se během učení pohybovalo kolem 96-100%. Přehled časové náročnosti učení a počty předložených trénovacích vzorů lze pro každý ze tří modelů pozorovat v tabulce 4.

Tab. 4: Přehled časové náročnosti učení a počtu předložených vzorů pro jednotlivé modely neuronových sítí

Název modelu	Rozměr vstupního obrazu	Doba učení	Počet předložených trénovacích vzorů	Počet vzorů v epoše
Velko-písmenný model	512x64x1	3 dny a 13 hodin	60 milionů	Prvních 40% 4 miliony, zbytek 10 milionů
Model pro křestní jména	384x64x1	1 den a 17 hodin	35 milionů	4 miliony
Numerický model	128x64x1	16 hodin	26 milionů	4 miliony

6.1.4 VYHODNOCENÍ

Jakmile bylo pro každý model neuronové sítě učení ukončeno, byla pozornost zaměřena na jeho vyhodnocení. Vyhodnocení bylo prováděno určením přesnosti modelu na testovací množině, pomocí něhož byly následně porovnávány i různé varianty modelů stejného zaměření. Při vyhodnocování modelů bylo nejvíce přihlíženo na jejich přesnost na úrovni sekvence. Vyhodnocení všech třech modelů pro rozpoznávání celých řádek textu je jednotlivě shrnuto v následujících podkapitolách.

V přehledech přesností v tabulkách 5, 6 a 7 je obecně možné pozorovat nižší přesnost u testovací množiny než u validační množiny reálných dat. Je tomu tak i přesto, že jsou obě množiny tvořeny řádky textu z reálných fotografií. Vyšší přesnost validační množiny je způsobena jednak tím, že byla, na rozdíl od testovací, používána k hledání ideálních hyperparametrů neuronové sítě, a také skutečností, že byla na základě histogramů ze vzorů validační množiny reálných dat generována syntetická data pro trénovací množinu. Rozdíl mezi přesnostmi na validační množině syntetických dat a validační množině reálných dat pak signalizuje, do jaké míry se podařilo připodobnit syntetická data datům reálným, přičemž čím menší je mezi nimi rozdíl, tím jsou syntetická data zdařilejší. Tento rozdíl byl sledován hlavně při implementaci a vylepšování generátoru syntetických dat (viz kapitola 6.1.1).

Velko-písmenný model

Tab. 5: Přehled přesností finálního velko-písmenného modelu na validačních množinách a testovací množině (vzory validační množiny reálných dat a testovací množiny omezeny abecedou modelu)

Množina	Přesnost na úrovni sekvence znaků	Přesnost na úrovni znaku v rámci délky textu	Přesnost na úrovni znaku	Počet nesprávně rozpoznaných vzorů
Validační množina syntetických dat	96,15%	99,71%	99,86%	770 z 20000
Validační množina reálných dat	94,4%	98,11%	99,58%	28 z 500
Testovací množina	**91,92%**	**97,79%**	**99,47%**	**43 z 532**

Obr. 84: Příklad rozpoznání některých vzorů z testovací množiny

(Pred: rozpoznaný údaj, T: očekávaný výstup pokud se nerovná rozpoznanému)

Model pro křestní jména

Tab. 6: Přehled přesností finálního modelu pro křestní jména na validačních množinách a testovací množině (vzory validační množiny reálných dat a testovací množiny omezeny abecedou modelu)

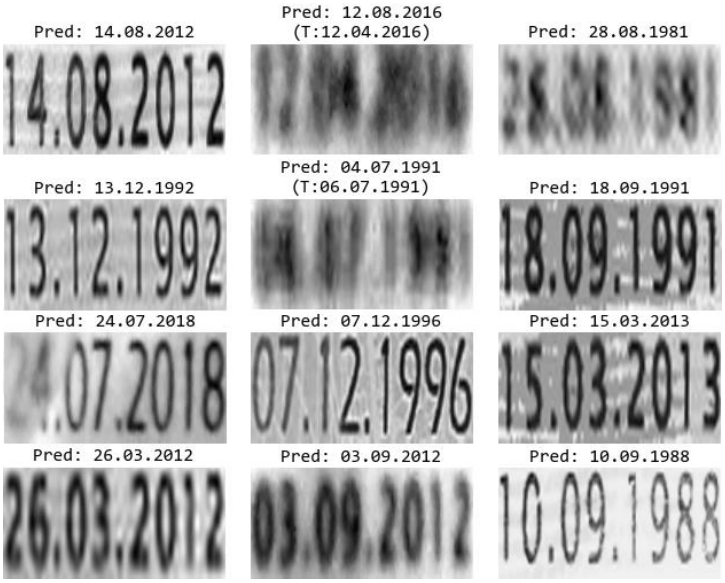
Množina	Přesnost na úrovni sekvence znaků	Přesnost na úrovni znaku v rámci délky textu	Přesnost na úrovni znaku	Počet nesprávně rozpoznaných vzorů
Validační množina syntetických dat	98,78%	99,81%	99,93%	244 z 20000
Validační množina reálných dat	90,0%	95,26%	99,22%	4 ze 40
Testovací množina	**83,33%**	**94,61%**	**99,06%**	**5 z 30**

Obr. 85: Příklad rozpoznání některých vzorů z testovací množiny
(Pred: rozpoznaný údaj, T: očekávaný výstup pokud se nerovná rozpoznanému)

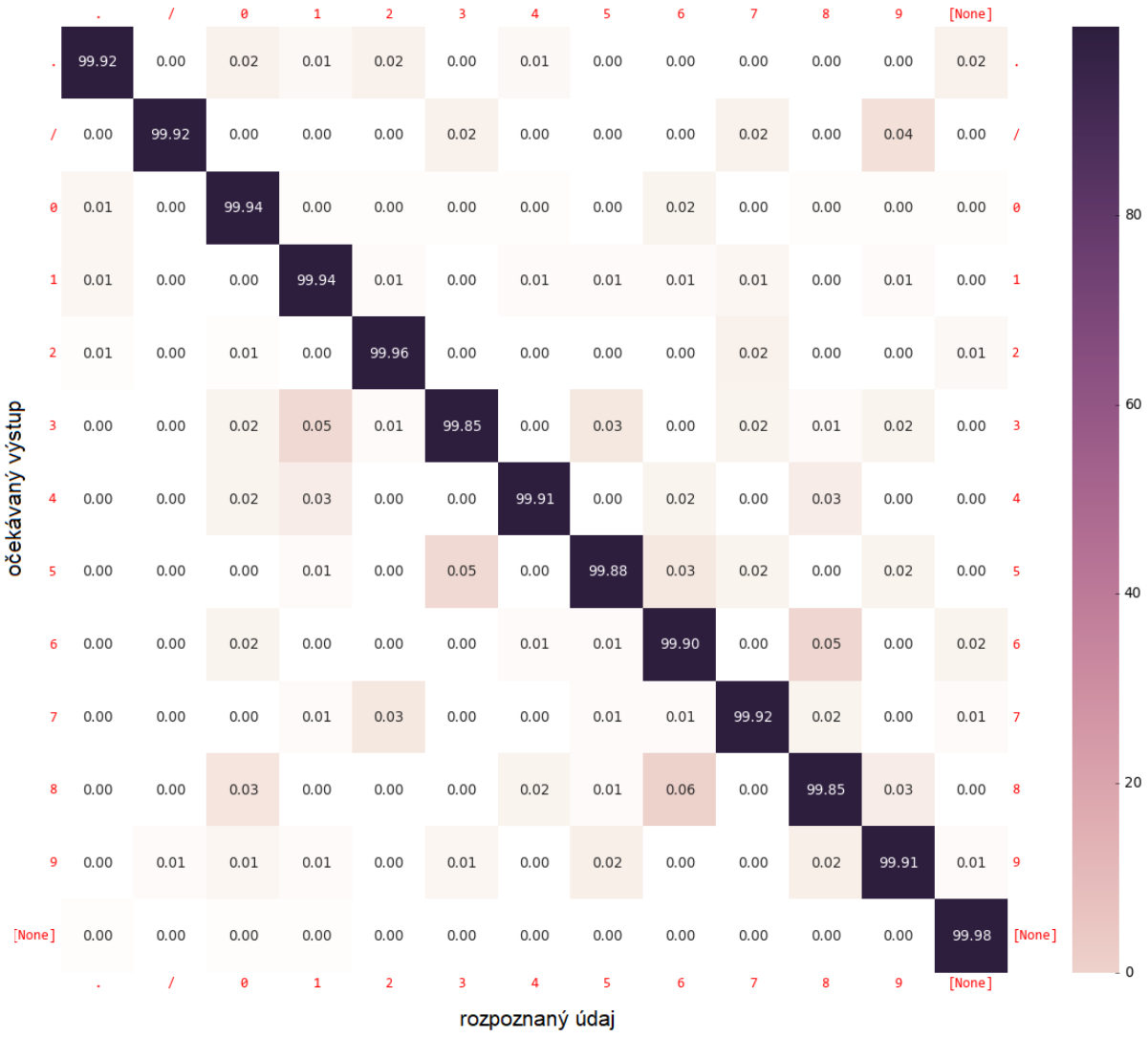
Numerický model

Tab. 7: Přehled přesností finálního numerického modelu na validačních množinách a testovací množině (vzory validační množiny reálných dat a testovací množiny omezeny abecedou modelu)

Množina	Přesnost na úrovni sekvence znaků	Přesnost na úrovni znaku v rámci délky textu	Přesnost na úrovni znaku	Počet nesprávně rozpoznaných vzorů
Validační množina syntetických dat	99,50%	99,91%	99,93%	101 z 20000
Validační množina reálných dat	99,71%	99,76%	99,79%	1 z 345
Testovací množina	**98,80%**	**99,88%**	**99,89%**	**4 z 332**



Obr. 86: Příklad rozpoznání některých vzorů z testovací množiny (Pred: rozpoznaný údaj, T: očekávaný výstup pokud se nerovná rozpoznanému)



Obr. 87: Matice záměn numerické modelu pro validační množinu syntetických dat

Diskuze

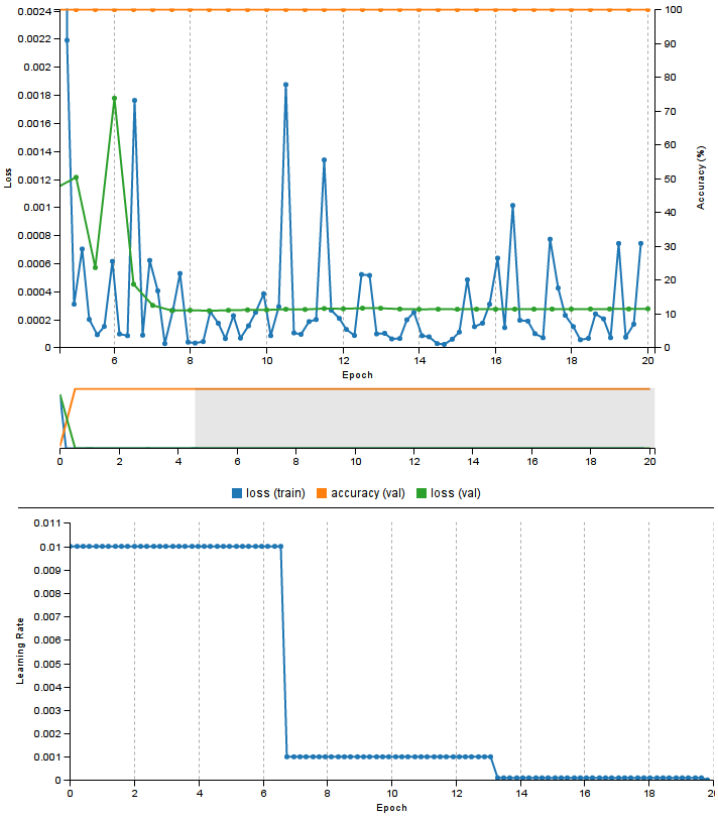
Během vyhodnocování jednotlivých modelů neuronové sítě bylo ve validační množině reálných dat a testovací množině odhaleno celkem 7 vzorů s nesprávným údajem o jejich obsahu. Tyto chyby vznikly z nepozornosti při manuální kompletaci obou množin (viz kapitola 6.1.2.2). Na základě těchto chyb lze konstatovat, že přesnost autora práce na úrovni sekvence znaků je po zaokrouhlení při 7 chybných vzorech z celkových 1779 použitých vzorů[19] 99,61%.

[19] Z původních 2019 vzorů (viz kapitola 6.1.2.2) mohlo být použito jen 1779, protože ne všechny vzory vyhovovaly abecedám jednotlivých modelů. Toto platí například pro řádky textu označující v občanských průkazech úřad, který doklad vydal. Čtení tohoto údaje není prozatím podporováno.

6.2 Model pro rozpoznávání znaků strojově čitelné oblasti

Tato kapitola je věnována implementaci konvoluční neuronové sítě pro rozpoznávání jednotlivých znaků strojově čitelné oblasti dokladu, která byla popsána v kapitole 5.2.2.3.

Protože zvolená architektura neuronové sítě neobsahuje žádné nestandardní prvky, jako tomu je u výstupních vrstev modelů pro rozpoznávání celých řádek textu, mohlo být pro její učení použito interaktivní řešení DIGITS [119] od společnosti NVIDIA. Pomocí DIGITS a základních znalostí softwarové knihovny Caffe [118], která pracuje na pozadí DIGITS a je pomocí jejího zápisu definována architektura neuronové sítě, tak bylo možné navrženou síť zhotovit velmi rychle a jednoduše a jediná komplikace tak spočívala ve sběru trénovacích dat, kterou se bude zabývat samostatná podkapitola.



Obr. 88: Průběh učení neuronové sítě pro rozpoznávání jednotlivých znaků strojově čitelné oblasti dokladu

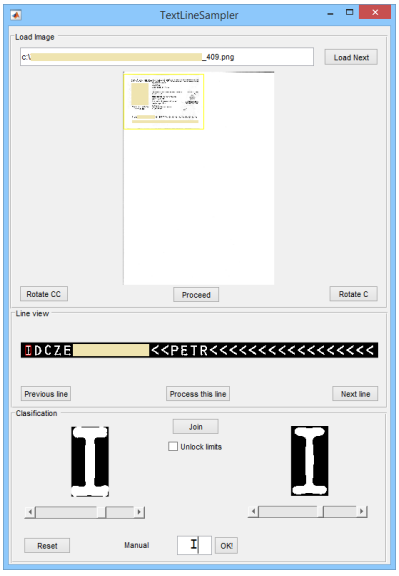
Samotné učení neuronové sítě bylo provedeno standardní metodou Stochastic Gradient Descent se zpětnou propagací chyby a setrvačností. Trénovací vzory byly během učení vybírány v rámci každé z epoch náhodně a každý vzor byl standardizován odečtem aritmetického průměru a podílem standardní deviace. Aritmetický průměr i standardní deviace

byly předem vypočítány přes vzory celé trénovací množiny. Velikost dávky na jednu změnu vah (angl. batch size) činila 100 trénovacích vzorů. Prvotní rychlost učení byla stanovena na hodnotu 0.01. Ta byla pak během učení automaticky snižována podle předem definovaných intervalů. Průběh učení spolu se snižováním jeho rychlosti je možné pozorovat na obrázku 88.

Na obrázku 88 je mimo jiné možné pozorovat i to, že vzhledem k mimořádně vysoké přesnosti na validační množině neuronová síť podává v rozpoznávání znaků strojově čitelné oblasti velmi dobré výsledky. DIGITS v době jeho používání však nepodporoval testovací množinu, a tak nebylo možné tyto výsledky ihned potvrdit. Následné pokusy s reálnými daty, které nebyly součástí validační ani trénovací množiny, však naznačovaly, že se přesnost na úrovni znaku skutečně pohybuje okolo 99%. Protože systém pro rozpoznávání strojově čitelné oblasti není prozatím ve výsledném řešení serverového modulu přímo používán, jeho vyhodnocením se tato část práce nebude nezabývat.

6.2.1 Příprava dat pro učení neuronové sítě

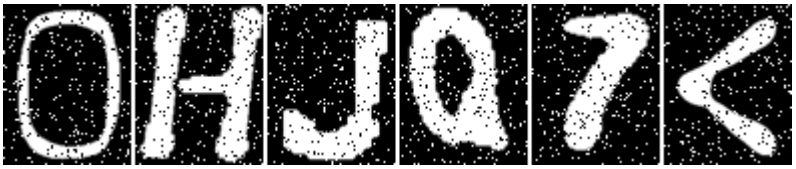
Konvoluční neuronová síť pro rozpoznávání jednotlivých znaků strojově čitelné oblasti dokladu byla učena převážně na datech z kolekce 550 reálných fotografií občanských průkazů a cestovních pasů, jež byla pro tuto práci k dispozici (viz kapitola 4.1). Pro jejich efektivní sběr byla, podobně jako pro přípravu dat modelu určeného k rozpoznávání celých řádek textu, vytvořena aplikace v jazyce Matlab (viz Obr. 89), která měla za úkol sběr znaků strojově čitelné oblasti alespoň z části zautomatizovat. V grafickém rozhraní aplikace je možné iterovat obrazy ve zvoleném adresáři, provést ořezání a rotaci dokladu, výběr řádky pro binarizaci a klasifikaci segmentovaného znaku.



Obr. 89: Grafické rozhraní aplikace pro manuální klasifikaci znaků strojově čitelné oblasti

Binarizace řádek textu byla prováděna podobně, jako to bylo popsáno v kapitole 5.2.2.1 s tím rozdílem, že byla pro navýšení počtu vzorů použita každá prahovací hodnota, při které byl znak ještě čitelný. Rozsah prahů bylo možné pomocí posuvníků i dále rozšiřovat, a tak z jednoho manuálně klasifikovaného znaku vznikalo přibližně 10 až 30 různých obrazů.

Po vyčerpání dat z kolekce reálných fotografií byly množiny jednotlivých znaků ještě augmentovány tak, aby počty vzorů mezi jednotlivými třídami znaků byly rovnoměrné. Pro jejich augmentaci byly použity operace rotace, zkosení, pixelizace a různé jejich kombinace. Některé znaky byly navíc ještě doplněny synteticky, protože jejich množství nebylo v reálných datech dostatečné (zejména písmena G, Q, W a X). Na všechny vzniklé vzory byl nakonec aplikován šum typu „sůl a pepř“ s pravděpodobností postihu pixelu 10% (viz Obr. 90).



Obr. 90: Příklad některých vzorů z validační množiny

Výsledná množina dat nakonec činila 1 110 000 unikátních obrazů znaků, z níž dvě třetiny byly přiřazeny trénovací množině a zbylá jedna třetina množině validační.

6.3 SERVEROVÁ APLIKACE

Aby byly jednotlivé prvky řešení pro rozpoznávání identifikačních údajů z osobních dokladů propojeny v jeden celistvý systém, byla vytvořena serverová aplikace. Tato aplikace zároveň poskytuje aplikační rozhraní v podobě REST API a celý systém tak lze ovládat pomocí http volání. Díky tomu lze aplikaci použít jak jako serverový modul v infrastruktuře typu microservices, tak jako samostatný server.

Serverová aplikace, která je naprogramována v jazyce Python, umožňuje pomocí definovaných koncových bodů (viz Tab. 8) provádět řadu operací, které nyní budou představeny v takovém pořadí, v jakém budou vykonávány při rozpoznávání identifikačních údajů z osobního dokladu. Operace přijmutí dokladu na svém vstupu očekává označení typu dokladu (nepovinné, systém je schopen typ dokladu odvodit) a jednu nebo dvě fotografie osobního dokladu podle toho, zda má doklad přední i zadní stranu. Jakmile jsou informace o dokladu úspěšně přijaty, server okamžitě odpovídá HTTP kódem 202 Accepted spolu s informací o číselném identifikátoru dokladu. Tento identifikátor je následně možné použít k operaci získání rozpoznaných identifikačních údajů. Pokud je doklad při volání této operace

stále ještě zpracováván, vrací server odpověď s HTTP kódem 404 Not Found. V opačném případě jsou navráceny rozpoznané identifikační údaje spolu s označením jistoty, s jakou se každý z nich podařilo rozpoznat, a označením typu dokladu. Identifikační údaje daného osobního dokladu jsou v tuto chvíli na serverové straně již uloženy v databázi a jejich záznam proto může později být ještě doplněn o případné opravy rozpoznaných údajů. Operace pro doplnění opravených údajů by měla být klientskou stranou volána i v případě, že rozpoznané identifikační údaje neobsahují chybu, přičemž tělo zprávy je pak klientem ponecháno prázdné. V tomto případě je pak jen doklad v databázi označen jako zkontrolovaný, aby mohl být později zahrnut do statistiky přesnosti rozpoznávání, pro jejíž získání slouží poslední operace (viz dále). Veškerá komunikace se serverem probíhá ve formátu JSON, přičemž binární data fotografií jsou přijímána v kódování BASE64.

Tab. 8: Přehled koncových bodů serverové aplikace

Koncový bod	**HTTP metoda**	**Stručný popis**
/documents	POST	Přijímání dokladu pro účel rozpoznávání identifikačních údajů
/document/<int:doc_id>	GET	Získání rozpoznaných identifikačních údajů z daného dokladu
	PUT	Doplnění daného dokladu o opravené identifikační údaje - označení kontroly člověkem
/statistics	GET	Získání statistiky přesnosti rozpoznávání dokladů

Údaje, které jsou součástí statistiky (druhy přesností jsou popsány v kapitole 6.1.3.2):

- Průměrná přesnost rozpoznávání celého systému na úrovni sekvence znaků (včetně korekce rozpoznaných údajů)
- Průměrná přesnost rozpoznávání celého systému na úrovni znaku v rámci délky textu (včetně korekce rozpoznaných údajů)
- Průměrná přesnost rozpoznávání na úrovni sekvence znaků přes všechny modely neuronové sítě
- Průměrná přesnost rozpoznávání na úrovni znaku v rámci délky textu přes všechny modely neuronové sítě
- Počet zkontrolovaných dokladů v databázi
- Počet zpracovaných obrazů zkontrolovaných dokladů (většina dokladů má dvě strany)
- Počet znaků napříč zkontrolovanými doklady

- Počet řádek napříč zkontrolovanými doklady
- Počet nesprávně rozpoznaných řádek
- Počet obrazů dokladů, pro které selhala lokalizace textu
- Poměr mezi počtem obrazů dokladů, pro které selhala lokalizace textu a počtem všech zpracovávaných obrazů dokladů

Pro propojení aplikace naprogramované v jazyce Python s algoritmy naprogramovanými v jazyce Matlab bylo použito MATLAB API pro Python [126], přičemž, je-li serverovou aplikací zpracováván doklad o dvou stranách, je lokalizace textu v obou fotografiích prováděna paralelně.

6.3.1 Určení jistoty rozpoznání řádky textu

Pro určení jistoty, s jakou se daný řádek textu podařilo rozpoznat, používá serverová aplikace dvou enumeračních hodnot: HIGH a UNSURE. První z nich značí vysokou jistotu správnosti rozpoznání a druhá označuje případ, ve kterém je rozpoznanému údaji potřeba při případné kontrole věnovat zvýšenou pozornost.

Jistotou HIGH označí server takové řádky textu, jejichž každý znak neuronová síť rozpoznala s jistotou větší než 99%. Pakliže existuje pro daný typ řádky slovník (viz Tab. 3 na str. 80) a je v rámci jeho prohledání rozpoznaný údaj potvrzen jediným platným nálezem, je práh jistoty rozpoznání znaků snížen na 90%. Ve všech opačných případech je jistota nastavena na UNSURE.

7 KLIENTSKÁ APLIKACE

Zatímco předmětem všech předchozích částí práce byl serverový modul pro rozpoznávání identifikačních údajů z osobních dokladů, tato kapitola bude věnována tématům klientské aplikace.

Aby bylo možné systém pro rozpoznávání identifikačních údajů z osobních dokladů jednoduše vyzkoušet jako celek včetně interakce s uživatelem, bylo nutné kromě serverového modulu vytvořit i klientskou aplikaci. Serverová aplikace poskytuje aplikační rozhraní v podobě REST API a možnosti pro formu klientské aplikace byly proto prakticky neomezené. Protože je však práce zaměřena na zpracování fotografií zejména z fotoaparátů mobilních telefonů, bylo rozhodnuto vytvořit aplikaci pro operační systém Android.

Klientská aplikace je naprogramována v jazyce Java jako nativní aplikace pro zařízení s operačním systém Android. Její grafické rozhraní pro pořízení fotografie osobního dokladu, odeslání dat na server a kontrolu rozpoznaných identifikačních údajů je koncipováno jako série 3 kroků, které budou postupně představeny v následujících podkapitolách.

7.1 VÝBĚR TYPU DOKLADU

V úvodní obrazovce je uživateli předložen výběr ze třech typů podporovaných osobních dokladů (viz Obr. 91). Uživatel by měl zvolit ten, který hodlá pomocí systému nechat zpracovat. Výběr typu dokladu by sice nemusel být nutný, protože jej serverový modul je schopen z fotografie automaticky rozpoznat, jeho označením je však snížena doba zpracování dokladu na minimum, což je při interakci s uživatelem žádané.

Obr. 91: Úvodní obrazovka s výběrem typu dokladu

Po zvolení jednoho z typů dokladu je uživatel automaticky přesměrován na další obrazovku.

7.2 Pořízení fotografie dokladu

V tomto kroku je prováděno pořizování fotografie osobního dokladu. Uživatel je instruován k umístění dokladu do rámečku a k pořízení fotografie, které je řešeno manuálním stiskem tlačítka se symbolem fotoaparátu (viz Obr. 92). Stiskem tlačítka je provedeno automatické ostření a následně vytvoření snímku ve formátu JPEG, který zůstává uchován v operační paměti telefonu pro pozdější odeslání do serverové aplikace.

Pokud automatické ostření například z důvodu špatného osvětlení neproběhne úspěšně, je o tom uživatel zpraven a je požádán o další pokus o pořízení fotografie. V případě, že se ani po třech pokusech automatické ostření nepodaří úspěšně dokončit, při dalším pokusu je již fotografie i přes opětovné selhání ostření pořízena.

Obr. 92: Obrazovka pro pořizování fotografie osobního dokladu

Rámeček, do kterého má uživatel osobní doklad umístit, byl do řešení pořizování fotografie zahrnut z několika důvodů. Původní myšlenka, která jeho implementaci motivovala, byla taková, že bude uživatel při pořizování fotografie přirozeně stimulován k zarovnání dokladu podle okrajů rámečku a tím nebudou vznikat snímky, ve kterých by doklad byl zachycen například z příliš ostrého úhlu nebo velmi pootočený. Obraz je navíc

možné podle okrajů rámečku ořezat a ve výsledné fotografii pak doklad zaujímá většinu jejího obsahu. Přítomnost rámečku v procesu pořizování fotografie má však význam i z technického hlediska. Tím, že je doklad nutné fotit z větší výšky, má automatické ostření mobilního fotoaparátu větší šanci na úspěch a zároveň ve výsledné fotografii nevzniká vlivem vad objektivu soudkovité zkreslení okrajů dokladu. Ač použitím rámečku dochází ke snižování rozlišení výsledného obrazu, velikost snímačů současných mobilních fotoaparátů je dostatečně velká na to, aby v něm bylo stále ještě dostatek informací. Použití rámečku se při pozdějším používání aplikace velmi osvědčilo.

Po pořízení fotografie je před přesměrováním uživatele na další obrazovku provedena ještě validace jejího obsahu. Ta je realizována hledáním podobných zájmových oblastí mezi šablonou dokladu a dokladem na fotografii na základě mechanismu zvaného *feature matching* tak, jak jej popisuje kapitola 4.3.1.2. Pomocí detektoru a deskriptoru KAZE oblastí [105] lze určit, zda je v obrazu přítomen očekávaný typ dokladu, jestli je v něm obsažen celou svou plochou a zda zabírá většinu obsahu rámečku. V případě, že tomu tak není, je uživatel požádán o nápravu. Pomocí stejné techniky je obraz rotován o násobky 90 stupňů tak, aby byl doklad orientován dle očekávání, a lze tak napravit situaci, ve které jej uživatel vyfotí například po přetočení telefonu. Neúspěšné nalezení šablony dokladu v pořízené fotografii přitom pracuje jako detekce špatné kvality obrazu.

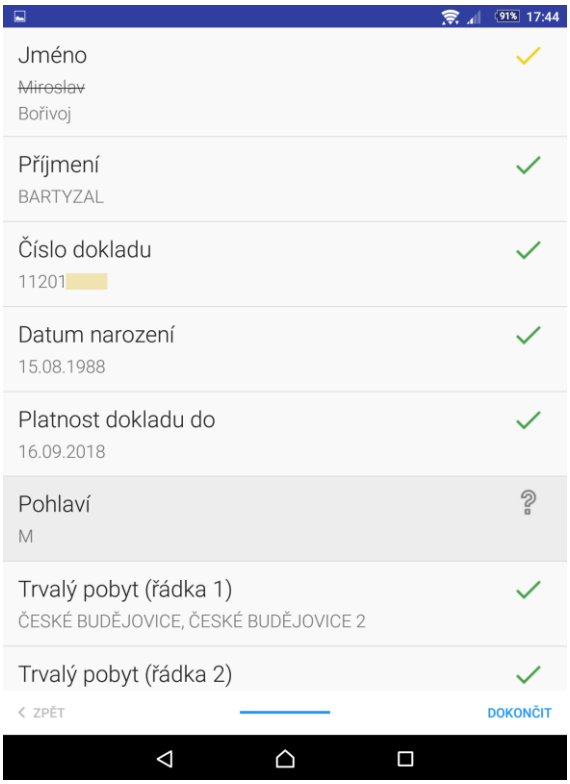
Po úspěšné validaci obsahu fotografie je uživatel automaticky přesměrován na další obrazovku, jež představuje buď stejný proces i pro zadní stranu dokladu, pokud jí daný typ dokladu disponuje, nebo poslední krok, popsaný v následující kapitole.

7.3 Kontrola rozpoznaných údajů

Na počátku posledního kroku klientská aplikace odesílá pomocí HTTP protokolu pořízené fotografie do serverové aplikace. Komunikace probíhá ve formátu JSON a binární data jsou kódována schématem BASE64. Na základě obdržené odpovědi, která obsahuje identifikátor zpracovávaného osobního dokladu, pak aplikace periodicky dotazuje stav jeho zpracování až do té doby, než je navrácena odpověď s rozpoznanými identifikačními údaji. Ty jsou pak uživateli zobrazeny k následné kontrole (viz Obr. 93).

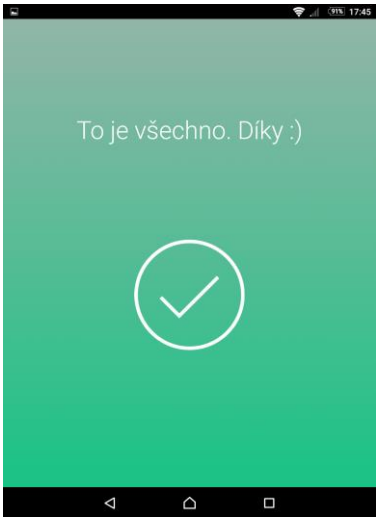
U každého ze zobrazovaných identifikačních údajů je v jeho pravém horním rohu pomocí zelené fajfky nebo šedého otazníku indikováno, s jakou jistotou se jej podařilo rozpoznat (viz kapitola 6.3.1). Údaje s nízkou jistotou rozpoznání jsou navíc zvýrazněny změnou barvy jejich pozadí tak, aby jim uživatel při kontrole věnoval zvýšenou pozornost. Pokud některý z údajů obsahuje chybu, je možné jej klepnutím přes editační formulář opravit.

Opravené identifikační údaje jsou pak označeny žlutou fajfkou a zobrazeny včetně původně rozpoznaného textu (viz Obr. 93 položka „Jméno“).



Obr. 93: Grafické rozhraní pro kontrolu a případnou opravu rozpoznaných identifikačních údajů

Po dokončení kontroly rozpoznaných identifikačních údajů může již uživatel stisknout tlačítko „DOKONČIT“ (viz Obr. 93 vpravo dole), čímž jsou informace o případných opravách zaslány zpět do serverové aplikace. Uživatel je pak přesměrován na závěrečnou obrazovku s poděkováním, přičemž po klepnutí na fajfku je klientská aplikace ukončena (viz Obr. 94).



Obr. 94: Závěrečná obrazovka

8 OVĚŘENÍ SYSTÉMU V PRAXI

Jedním z cílů této práce bylo výsledný systém pro rozpoznávání identifikačních údajů z osobních dokladů rovněž otestovat v praktickém provozu, aby tak byla vyhodnocena jeho spolehlivost a odhaleny jakékoliv jeho případné nedostatky.

8.1 PRŮBĚH TESTOVÁNÍ

Než bude přistoupeno k samotným výsledkům z testování, bude v této kapitole věnována pozornost způsobu, jakým byl systém testován, a následně vylepšením, které byly ještě v průběhu testů realizovány.

8.1.1 PODMÍNKY TESTOVÁNÍ

Systém pro rozpoznávání identifikačních údajů z osobních dokladů byl testován pomocí klientské aplikace, jejíž popisem se zabývala předchozí kapitola.

Testování probíhalo výhradně na autorově mobilním zařízení Sony Xperia Z3 Tablet Compact (SGP621) s rozlišením snímače fotoaparátu 8,1 Mpx, ve kterém byla klientská aplikace předinstalována. Přístup k systému z jiného mobilního zařízení nebyl prozatím uskutečněn. Komunikace se serverovou aplikací, která byla po dobu testování spuštěna na domácím stolním počítači, byla realizována přes šifrované VPN spojení[20]. Veškeré fotografie osobních dokladů byly na serverové straně po rozpoznání identifikačních údajů ihned vymazány.

Každý uživatel systému se testování účastnil dobrovolně a manipuloval jak s mobilním zařízením, tak s klientskou aplikací samostatně, pouze za doprovodu verbálního výkladu autora práce. Aby byla rozpoznávaná data pro výslednou statistiku co nejrozmanitější, žádný z osobních dokladů nebyl do systému zaslán opakovaně.

Selhal-li systém na serverové straně ve fotografii osobní doklad lokalizovat a navrátil-li do klientské aplikace identifikační údaje pouze z jedné ze stran dokladu, byly nekompletní údaje i tak zkontrolovány a výsledky zaslány zpět na server. V takovém případě byl poté proces rozpoznávání identifikačních údajů opakován až do té doby, než se lokalizace dokladu podařila provést v plném rozsahu. Selhal-li systém v obraze osobní doklad lokalizovat i během opakování, odesílání nekompletních výsledků zpět na server již neprobíhalo.

[20] RSA klíče délky 4096 bitů, AES-CBC klíče délky 256 bitů

Nastala-li při testování systému situace, ve které byl v obraze osobní doklad sice lokalizován, ale většina rozpoznaných identifikačních údajů obsahovala nesrozumitelné hodnoty, výsledky nebyly opravovány ani zasílány zpět na server a proces rozpoznávání byl zopakován. Tento stav mohl být způsoben chybnou lokalizací dokladu, extrémně nekvalitní fotografií nebo kombinací obojího.

8.1.2 Realizovaná vylepšení

Během testování systému postupně vznikala řada poznatků o tom, jakým způsobem by jej bylo možné vylepšit. S některými poznatky přišli sami uživatelé v podobě zpětné vazby při užívání klientské aplikace, jiné vyplynuly z různých nastalých situací.

Většina námětů na vylepšení, která během testování vznikla a která neměla vliv na statistické vyhodnocení jeho výsledků, byla v průběhu testování ihned implementována. V původní verzi klientské aplikace uživatelé například po výběru typu osobního dokladu postrádali automatický přechod na další obrazovku, a stejně tak i ukončení aplikace po klepnutí na symbol fajfky v obrazovce závěrečné. Čas od času měli uživatelé při pořizování fotografie dokladu tendence jej k přítomnému rámečku zarovnat velmi těsně a následná validace snímku je pak instruovala k zopakování focení z důvodu oříznuté karty dokladu. Protože je však pro rozpoznávání identifikačních údajů důležité hlavně to, aby v pořízených fotografiích byly kompletní samotné identifikační údaje a nikoliv celé karty dokladů, byla tato validace zmírněna na oblast jejich řádek textu. Do klientské aplikace byla zabudována kontrola výsledku automatického ostření až na základě situace, ve které většina rozpoznaných údajů obsahovala nesmyslné hodnoty z důvodu jeho selhání při špatném okolním osvětlení.

Na základě praktického užití systému bylo odhaleno také to, že byla předtím v serverové aplikaci opomenuta podpora pro držitele občanských průkazů narozených v jiné zemi než v České republice. V takovém případě údaj o místě narození držitele obsahuje pouze kódové označení země a systém při zpracování tohoto dokladu selhával. Podobně systém vzácně selhával při zpracování přední strany občanského průkazu nového typu, což bylo následně napraveno změnou patřičných prahů v algoritmu pro lokalizaci textu.

8.2 Výsledky

Výsledky testování systému pro rozpoznávání identifikačních údajů z osobních dokladů jsou zaměřeny majoritně na dvě ze třech jeho částí, kterými jsou rozpoznávání celých řádek textu a korekce rozpoznaného textu. Lokalizace textu, na které jsou zbylé dvě zmíněné části systému závislé, byla totiž ještě v běhu testování vylepšována a nelze tak její aktuální stav pomocí

nasbíraných dat spolehlivě vyhodnotit. Je však vhodné uvést, že lokalizace textu napříč testováním nezávisle na jejích úpravách dokázala úspěšně lokalizovat řádky na 93% obrazů osobních dokladů.

Během testování systému se podařilo nashromáždit informace o zpracování celkem 90 fotografií odpovídajících 50 jedinečným osobním dokladům. Na základě těchto informací byl pak vypracován přehled výsledků z testování, který lze pozorovat v tabulce 9. Na základě dosažených výsledků lze konstatovat, že je vytvořený systém ve čtení identifikačních údajů z osobních dokladů velmi spolehlivý, o čemž svědčí jeho vysoká přesnost správného rozpoznání celého řádku textu, odpovídající 99,36% (13. řádek). Je však nutné poznamenat, že do tohoto výsledku nejsou započteny 3 případy vyčtení, při kterých většina rozpoznaných údajů obsahovala nesrozumitelné hodnoty (6. řádek). Minimálně dva z těchto případů však pravděpodobně souvisely s absencí kontroly výsledku automatického ostření, které bylo do klientské aplikace implementováno až později. Třetí případ vznikl při velmi nevhodném osvětlení snímané scény.

Z přehledu výsledků je patrné, že k velmi vysoké přesnosti celého systému významně přispěla korekce rozpoznaného textu (9. řádek vůči 8., 15. vůči 13. a 16. vůči 14.), bez které by systém dosahoval přesnosti správného rozpoznání celého řádku textu pouze 94,88% (15. řádek).

Dále je v tabulce možné vidět, že z celkem 4 chyb, kterých se systém dopustil (8. řádek), 3 byly označeny sníženou jistotou rozpoznání a 1 chyba jistotou vysokou (12. řádek). Po vyšetření chyby, která byla označena vysokou měrou jistoty rozpoznání, bylo odhaleno, že se systém zmýlil v počátečním písmenu „W“, které zaměnil za písmeno „V“. Jistota rozpoznání znaku na této pozici však opravdu nijak nenaznačovala, že by v tomto případě mohlo jít o nejisté rozpoznání. Protože se jedná o počáteční znak a po jeho částečném oříznutí zleva by se mohlo opravdu jevit jako písmeno „V“, pravděpodobně v tomto případě došlo k chybě spíše na straně lokalizace textu, než na straně jeho rozpoznávání.

Z celkových 48 údajů se sníženou jistotou rozpoznání (10. řádek) byly jen 3 údaje opravdu chybné (vyplývá z 8. a 12. řádku). Vzhledem k okolnostem jediné chyby v označení vysoké jistoty rozpoznání (viz předchozí odstavec) by se proto dalo uvažovat o zmírnění prahů pro označování nejistě rozpoznaných údajů.

Každá ze 4 chyb, kterých se systém dopustil, se týkala odlišného typu identifikačního údaje. Chyby se vztahovaly k chybnému určení čísla popisného v trvalém pobytu držitele, rodného příjmení držitele, data vydání dokladu a označení kódu země, která doklad vydala.

V popisném čísle držitelova trvalého pobytu bylo chybně lomítko „/“ označeno za číslici sedm „7“.

Dle zpětné vazby uživatelé byli při testování překvapeni přesností výsledků rozpoznávání a podporou diakritiky včetně přehlásek. Pochvalovali si i rychlou odezvu zpracování, která se na serverové straně pohybovala v hodnotách kolem 3-8 vteřin.

Tab. 9: Přehled nashromážděných dat a výsledků z testování (druhy přesností jsou popsány v kapitole 6.1.3.2 na str. 93)

	Popis hodnoty	Hodnota
1.	Počet dokladů	50
2.	Počet občanských průkazů nového typu	36
3.	Počet občanských průkazů staršího typu	9
4.	Počet cestovních pasů	5
5.	Počet obrazů dokladů (většinou dva obrazy na doklad - přední a zadní strana)	90
6.	Počet dokladů, pro které nebyla z důvodu kompletního selhání rozpoznání identifikačních údajů provedena oprava a nejsou proto ve statistice zahrnuty (většina rozpoznaných identifikačních údajů obsahovala nesrozumitelné hodnoty)	3
7.	Počet řádek textu	625
8.	Počet nesprávně rozpoznaných řádek textu	4
9.	Počet řádek textu, které nebyly prvotně rozpoznány správně, ale byly automaticky opraveny systémem korekce rozpoznaného textu	32
10.	Počet rozpoznaných řádek textu se sníženou měrou jistoty správného rozpoznání (jistota typu UNSURE)	48
11.	Počet rozpoznaných řádek textu s vysokou měrou jistoty správného rozpoznání (jistota typu HIGH)	577
12.	Počet nesprávně rozpoznaných řádek textu navzdory vysoké jistotě správnosti rozpoznaných údajů (jistota typu HIGH)	1
13.	**Průměrná přesnost celého systému na úrovni sekvence znaků (včetně korekce rozpoznaných údajů)**	**99,36%**
14.	Průměrná přesnost celého systému na úrovni znaku v rámci délky textu (včetně korekce rozpoznaných údajů)	99,92%
15.	Průměrná přesnost na úrovni sekvence znaků přes všechny modely neuronové sítě (bez korekce rozpoznaných údajů)	94,88%
16.	Průměrná přesnost na úrovni znaku v rámci délky textu přes všechny modely neuronové sítě (bez korekce rozpoznaných údajů)	98,75%

8.3 Náměty na další vylepšení

Přestože testováním systému v praktickém provozu bylo prokázáno, že je schopen identifikační údaje z fotografií osobních dokladů rozpoznávat s vysokou přesností, nabízí se ještě mnoho možností, jak jej dále vylepšit.

Část aktuálně rozpoznávané sady identifikačních údajů by mohla být porovnávána se strojově čitelnou oblastí dokladu. Na základě výsledků porovnání by pak mohla pracovat například oprava číselných údajů, které v současném stavu systému nepodléhají žádné kontrole. Systém pro zpracování strojově čitelné oblasti je přitom připraven a funkční, stačí jej už jen k aktuálnímu řešení připojit.

Přestože je systém schopen zpracovat většinu identifikačních údajů, které v dokladech figurují, stále ještě chybí podpora pro vyčítání rodinného stavu a titulu držitele a informace o úřadu, který doklad vydal. Doplnění systému o zpracování údajů rodinného stavu držitele a označení úřadu bude spočívat pouze v naučení nových specializovaných modelů neuronové sítě (oba údaje jsou tvořeny specifickou abecedou) a úpravě serverové aplikace. Ostatní prvky systému, jako lokalizace textu nebo klientská aplikace jsou na jejich zpracování již připraveny. Podporu zpracování titulu držitele bude navíc potřeba doplnit i do patřičných šablon řešení lokalizace textu.

Protože neexistuje spolehlivý způsob, jakým by mohla být systémem ověřena správnost rozpoznání čísla popisného v trvalém pobytu držitele, mohl by pro tento účel být navržen specializovaný model neuronové sítě. Ten by pak mohl těžit ze svého úzkého zaměření a podobně jako numerický model podávat velmi přesvědčivé výsledky (viz kapitola 6.1.4).

Aby bylo zabráněno situacím, ve kterých je nedopatřením pořízen takový snímek dokladu, který je obtížně čitelný i člověkem, mohla by klientská aplikace uživateli předložit pořízenou fotografii ke kontrole. Bylo by pak na uvážení samotného uživatele, jestli se pokusí osobní doklad vyfotit znovu, nebo jej odešle k dalšímu zpracování.

Někteří uživatelé se při testování systému zmínili o tom, že při pořizování snímku dokladu čekali asistenci v podobě automatické spouště fotoaparátu. S tou bylo původně při implementaci klientské aplikace počítáno, v zájmu dodržení termínů od ní však bylo upuštěno. Mohla by ale být realizována v příští verzi aplikace.

9 ZÁVĚR

V rámci této práce byl vytvořen serverový systém pro čtení identifikačních údajů z fotografií osobních dokladů. Systém je plně automatizovaný. Jeho vstupem je fotografie osobního dokladu a výstupem strukturovaný výčet identifikačních údajů včetně určení kvality jejich rozpoznání. Vstupní fotografie přitom mohou mít různou kvalitu obrazu odpovídající snímkům z fotoaparátu mobilních telefonů pořízených za různých světelných podmínek.

Na rozdíl od konkurenčních řešení stejného zaměření dokáže vytvořený systém zpracovat většinu informací, které se v daném osobním dokladu nacházejí. Zpracovávány jsou fotografie obou stran dokladu a kromě informací, které obsahuje jeho strojově čitelná oblast, na níž se konkurenční řešení často zaměřují, jsou čteny i údaje například o trvalém pobytu, místě narození, rodném čísle a rodném příjmení držitele. Systém navíc podporuje širokou abecedu znaků a veškeré identifikační údaje jsou zpracovány včetně diakritických znamének (háčky, čárky, kroužky, přehlásky a další).

Vedle serverového systému byla vytvořena i klientská aplikace pro mobilní zařízení s operačním systémem Android. Klientská aplikace je určena k pořizování fotografií osobních dokladů, validaci obsahu zhotovených snímků, komunikaci s vytvořeným systémem a následné kontrole rozpoznaných identifikačních údajů. Validace obsahu pořízené fotografie je schopna rozpoznat špatnou kvalitu obrazu, nepřítomnost nebo uříznutí očekávaného osobního dokladu a napravit jeho případné přetočení.

Vytvořený systém byl pomocí klientské aplikace otestován v praktickém provozu. Po nashromáždění celkem 90 fotografií osobních dokladů byly systémem chybně rozpoznány pouze 4 z 625 zpracovaných identifikačních údajů a systém tak dosáhl nevídaných 99,36% správných rozpoznání. Uživatelé při testování systému manipulovali s mobilním zařízením samostatně a snímky dokladů byly pořizovány za různých světelných podmínek. Žádný z osobních dokladů nebyl při testování použit více než jednou.

Pro vytvořený systém byla navržena a implementována řada originálních algoritmů. Mezi tyto algoritmy patří například lokalizace řádek textu v obraze, určení významu nalezených řádek textu, prahování řádky textu nebo segmentace znaků strojově čitelné oblasti dokladu. Algoritmy lokalizace řádek textu a prahování řádky textu jsou přitom použitelné i pro práce s jiným zaměřením, a byly proto v zájmu podpory vědy uloženy na datový nosič, který je k této práci přiložen.

Rozpoznávání identifikačních údajů z obrazů řádek textu je realizováno s využitím nejmodernějších pokroků v oblasti hlubokého učení neuronových sítí. Model konvoluční neuronové sítě, který je v systému použit, přijímá na svém vstupu obraz celé řádky textu, na jehož základě přímo odvozuje přítomnou posloupnost znaků. Podobná architektura neuronové sítě byla již jinými pracemi použita pro čtení posloupnosti až 5 čísel [47] a později i celých slov [68], o čtení celé řady slov o délce až 44 znaků se ale žádná jiná práce ještě nepokusila. Tato práce tak přináší nové poznatky pro stav poznání.

Výsledný systém bude nasazen do reálného provozu finančně technologickou společností GoPay s.r.o., v rámci něhož bude autorem dále vyvíjen a vylepšován.

LITERATURA

[1] RUSSAKOVSKY, Olga, Jia DENG, Hao SU, et al. *ImageNet Large Scale Visual Recognition Challenge* [online]. 2014 [cit. 2018-03-16]. Dostupné z: https://arxiv.org/abs/1409.0575

[2] *ICDAR Robust Reading Competition* [online]. [cit. 2018-03-16]. Dostupné z: http://rrc.cvc.uab.es/

[3] Kaggle Competitions. *Kaggle* [online]. c2018 [cit. 2018-03-16]. Dostupné z: https://www.kaggle.com/competitions

[4] CIREŞAN, Dan, Ueli MEIER a Juergen SCHMIDHUBER. M*ulti-column Deep Neural Networks for Image Classification* [online]. 2012 [cit. 2018-03-16]. Dostupné z: https://arxiv.org/abs/1202.2745

[5] HE, Kaiming, Xiangyu ZHANG, Shaoqing REN a Jian SUN. *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification* [online]. 2015 [cit. 2018-03-16]. Dostupné z: https://arxiv.org/abs/1502.01852

[6] Změny v regulaci sázení od roku 2017. *Tipsport* [online]. Beroun: Tipsport, 2018 [cit. 2018-03-17]. Dostupné z: https://www.tipsport.cz/napoveda/kategorie/600-regulace-sazeni-2017

[7] SOJKA, Eduard, Jan GAURA a Michal KRUMNIKL. *Matematické základy digitálního zpracování obrazu* [online]. Ostrava, 2011 [cit. 2018-03-18]. Dostupné z: http://mi21.vsb.cz/sites/mi21.vsb.cz/files/unit/digitalni_zpracovani_obrazu.pdf

[8] YU, Fisher a Vladlen KOLTUN. *Multi-Scale Context Aggregation by Dilated Convolutions* [online]. 2015 [cit. 2018-03-19]. Dostupné z: https://arxiv.org/abs/1511.07122

[9] Thresholding methods. In: *MathWorks* [online]. c1994-2018 [cit. 2018-03-19]. Dostupné z: https://www.mathworks.com/discovery/image-segmentation.html

[10] FORSYTH, David A. a Jean PONCE. *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference, 2002.

[11] Edge Detection. In: *Introduction to computer vision* [online]. [cit. 2018-03-19]. Dostupné z: http://ai.stanford.edu/~syyeung/cvweb/tutorial1.html

[12] CANNY, John. A computational approach to edge detection. In: IE*EE Transactions on Pattern Analysis and Machine Intelligence* [online]. IEEE, 1986, s. 679-698 [cit.

2018-03-19]. DOI: 10.1109/TPAMI.1986.4767851. ISSN 0162-8828. Dostupné z: https://pdfs.semanticscholar.org/55e6/6333402df1a75664260501522800cf3d26b9.pdf

[13] *OpenCV* [online]. c2018 [cit. 2018-03-19]. Dostupné z: https://opencv.org/

[14] *MathWorks* [online]. Natick (Massachuttes): MathWorks, c1994-2018 [cit. 2018-03-19]. Dostupné z: https://www.mathworks.com/

[15] *ImageMagick* [online]. ImageMagick Studio, c1999-2018 [cit. 2018-03-19]. Dostupné z: https://www.imagemagick.org/

[16] VAN DER WALT, Stéfan, Johannes L. SCHÖNBERGER, Juan NUNEZ-IGLESIAS, et al. *Scikit-image: image processing in Python* [online]. PeerJ 2:e453, 2014 [cit. 2018-03-19]. Dostupné z: http://scikit-image.org

[17] Global image threshold using Otsu's method. In: *MathWorks* [online]. Natick (Massachuttes), c1994-2018 [cit. 2018-03-19]. Dostupné z: https://www.mathworks.com/help/images/ref/graythresh.html

[18] Contours: Getting Started. *OpenCV* [online]. 2017 [cit. 2018-03-19]. Dostupné z: https://docs.opencv.org/3.3.1/d4/d73/tutorial_py_contours_begin.html

[19] JAN, Jiří. *Medical image processing, reconstruction, and restoration: concepts and methods*. Boca Raton, FL: CRC Press, 2006. ISBN 978-0-8247-5849-3.

[20] DUDA, Richard O. a Peter E. HART. Use of the Hough transformation to detect lines and curves in pictures. In: *Communications of the ACM* [online]. 1972, s. 11-15 [cit. 2018-03-19]. Dostupné z: https://www.cse.unr.edu/~bebis/CS474/Handouts/HoughTransformPaper.pdf

[21] Straight line Hough transform. In: *Scikit-image: Image processing in Python* [online]. PeerJ 2:e453, 2014 [cit. 2018-03-19]. Dostupné z: http://scikit-image.org/docs/0.13.x/auto_examples/edges/plot_line_hough_transform.html

[22] EPSHTEIN, Boris, Eyal OFEK a Yonatan WEXLER. Detecting Text in Natural Scenes with Stroke Width Transform. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* [online]. San Francisco: IEEE, 2010, s. 2963-2970 [cit. 2018-03-19]. DOI: 10.1109/CVPR.2010.5540041. ISBN 978-1-4244-6985-7. ISSN 1063-6919. Dostupné z: http://nlp.cs.rpi.edu/paper/all.pdf

[23] NEUMANN, Lukáš. *Scene text localization and recognition in images and videos* [online]. Praha, 2017 [cit. 2018-03-19]. Dostupné z: http://cyber.felk.cvut.cz/teaching/radaUIB/disertace_Neumann.pdf. Disertace. České vysoké učení technické v Praze, Fakulta elektrotechnická. Vedoucí práce Jiří Matas.

[24] OTSU, Nobuyuki. A Threshold Selection Method from Gray-Level Histograms. In: *IEEE Transactions on Systems, Man, and Cybernetics* [online]. IEEE, 1979, 62 - 66 [cit. 2018-03-19]. DOI: 10.1109/TSMC.1979.4310076. ISSN 0018-9472. Dostupné z: http://ieeexplore.ieee.org/document/4310076/

[25] NIBLACK, Wayne. *An introduction to digital image processing*. Englewood Cliffs, N.J.: Prentice-Hall International, c1986. ISBN 978-0134806747.

[26] BRADLEY, Derek a Gerhard ROTH. Adaptive Thresholding using the Integral Image. In: *Journal of Graphics Tools* [online]. 2011, **12**(2), s. 13-21 [cit. 2018-03-19]. DOI: 10.1080/2151237X.2007.10129236. ISBN 10.1080/2151237X.2007.10129236. ISSN 1086-7651. Dostupné z: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.420.7883&rep=rep1&type=pdf

[27] FISHER, Robert, Simon PERKINS, Ashley WALKER a Erik WOLFART. Adaptive Thresholding. *Hypermedia Image Processing Reference* [online]. c2003 [cit. 2018-03-19]. Dostupné z: http://homepages.inf.ed.ac.uk/rbf/HIPR2/adpthrsh.htm

[28] FENG, Meng-Ling a Yap-Peng TAN. Contrast adaptive binarization of low quality document images. In: *IEICE Electronics Express* [online]. 2004, **1**(16), s. 501-506 [cit. 2018-03-19]. DOI: 10.1587/elex.1.501. ISSN 1349-2543. Dostupné z: https://pdfs.semanticscholar.org/17e4/37f4ac8921b6629e0b4bcb869f9e6e9761d5.pdf

[29] SAUVOLA, J. a M. PIETIKÄINEN. Adaptive document image binarization. In: *Pattern Recognition* [online]. 2000, **33**(2), s. 225-236 [cit. 2018-03-19]. DOI: 10.1016/S0031-3203(99)00055-2. ISSN 00313203. Dostupné z: http://www.ee.oulu.fi/research/mvmp/mvg/files/pdf/pdf_24.pdf

[30] HORÁK, Karel, Ilona KALOVÁ, Petr PETYOVSKÝ a Miloslav RICHTER. *Počítačové vidění* [online]. Brno: Vysoké učení technické v Brně, 2008 [cit. 2018-03-19]. Dostupné z: http://www.uamtold.feec.vutbr.cz/vision/TEACHING/MPOV/Pocitacove_videni_S.pdf

[31] RUSS, John C. *The Image Processing and Analysis Cookbook* [online]. Asheville, c2001 [cit. 2018-03-20]. Dostupné z: http://web4.cbm.uam.es/joomla-rl/images/Servicios/070.Microscopia-optica-cfocal/documentos/manuales/Image_tool_kit.pdf

[32] DVOŘÁK, Pavel. *Popis objektů v obraze* [online]. Brno, 2011 [cit. 2018-03-20]. Dostupné z:

https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=39299. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Vedoucí práce Martin Zukal.

[33] ŽELEZNÝ, Miloš. *Zpracování digitalizovaného obrazu* [online]. 2015 [cit. 2018-03-24]. Dostupné z: http://www.kky.zcu.cz/uploads/courses/zdo/ZDO_aktual_130215.pdf

[34] ŠONKA, Milan, Václav HLAVÁČ a Roger BOYLE. *Image processing, analysis, and machine vision* [online]. 3rd ed., International student edition. Toronto [u.a.]: Thomson, 2008 [cit. 2018-03-24]. ISBN 0-495-24438-4.

[35] GOODFELLOW, Ian, Yoshua BENGIO a Aaron COURVILLE. *Deep Learning* [online]. MIT Press, 2016 [cit. 2018-03-24]. Dostupné z: http://www.deeplearningbook.org/

[36] KARPATHY, Andrej. *CS231n Convolutional Neural Networks for Visual Recognition* [online]. 2015 [cit. 2018-03-24]. Dostupné z: http://cs231n.github.io/

[37] TAUFER, I., O. DRÁBEK a P. SEIDL. Umělé neuronové sítě - základy teorie a aplikace. *CHEMagazín* [online]. 2009, Květen/Červen 2009, **XIX**(3), 38-41 [cit. 2018-03-24]. ISSN 1210-7409. Dostupné z: http://www.chemagazin.cz/userdata/chemagazin_2010/file/chxix_3_cl11.pdf

[38] KRIZHEVSKY, Alex, Ilya SUTSKEVER a Geoffrey E. HINTON. *ImageNet Classification with Deep Convolutional Neural Networks* [online]. 2012 [cit. 2018-03-24]. Dostupné z: http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf

[39] NIELSEN, Michael A. *Neural Networks and Deep Learning* [online]. Determination Press, 2015 [cit. 2018-03-24]. Dostupné z: http://neuralnetworksanddeeplearning.com/

[40] CYBENKO, George. Approximation by superpositions of a sigmoidal function. In: *Mathematics of control, signals and systems* [online]. 1989, s. 303-314 [cit. 2018-03-24]. Dostupné z: http://www.dartmouth.edu/~gvc/Cybenko_MCSS.pdf

[41] TRENZ, Oldřich. *Umělá inteligence I: Neuronové sítě* [online]. Brno: Mendelova univerzita v Brně, 2009 [cit. 2018-03-24]. Dostupné z: https://is.mendelu.cz/eknihovna/opory/zobraz_cast.pl?cast=21471

[42] JOHNSON, Justin. Benchmarks for popular CNN models. *GitHub* [online]. Sep 25, 2017 [cit. 2018-03-24]. Dostupné z: https://github.com/jcjohnson/cnn-benchmarks

[43] CHANGHAU, Isaac. Loss Functions in Artificial Neural Networks. *Isaac Changhau* [online]. 2017-06-07 [cit. 2018-03-24]. Dostupné z:

https://isaacchanghau.github.io/2017/06/07/Loss-Functions-in-Artificial-Neural-Networks/

[44] RUMELHART, David E., Geoffrey E. HINTON a Ronald J. WILLIAMS. Learning representations by back-propagating errors. *Nature* [online]. 1986, 9 October 1986, **323**(6088), 533-536 [cit. 2018-03-24]. Dostupné z: https://www.iro.umontreal.ca/~vincentp/ift3395/lectures/backprop_old.pdf

[45] GOH, Gabriel. Why Momentum Really Works. *Distill* [online]. 4 April 2017 [cit. 2018-03-24]. ISSN 2476-0757. Dostupné z: https://distill.pub/2017/momentum/

[46] SRIVASTAVA, Nitish, Geoffrey HINTON, Alex KRIZHEVSKY, Ilya SUTSKEVER a Ruslan SALAKHUTDINOV. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research* [online]. 2014, 6/14, **15**(1), 1929-1958 [cit. 2018-03-24]. Dostupné z: http://www.cs.toronto.edu/~rsalakhu/papers/srivastava14a.pdf

[47] GOODFELLOW, Ian J., Yaroslav BULATOV, Julian IBARZ, Sacha ARNOUD a Vinay SHET. *Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks* [online]. 2013 [cit. 2018-03-25]. Dostupné z: https://arxiv.org/abs/1312.6082

[48] ZEILER, Matthew D a Rob FERGUS. *Visualizing and Understanding Convolutional Networks* [online]. 2013 [cit. 2018-03-25]. Dostupné z: https://arxiv.org/abs/1311.2901

[49] SIMONYAN, Karen a Andrew ZISSERMAN. V*ery Deep Convolutional Networks for Large-Scale Image Recognition* [online]. 2014 [cit. 2018-03-25]. Dostupné z: https://arxiv.org/abs/1409.1556

[50] LIU, Zongyi a Sudeep SARKAR. Robust outdoor text detection using text intensity and shape features. *2008 19th International Conference on Pattern Recognition* [online]. IEEE, 2008 [cit. 2018-03-25]. DOI: 10.1109/ICPR.2008.4760921. ISBN 978-1-4244-2174-9. Dostupné z: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.214.5416&rep=rep1&type=pdf

[51] LIU, Xiaoqian, Ke LU a Weiqiang WANG. Effectively localize Text in Natural Scene Images. In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)* [online]. IEEE, 2012, s. 1197-1200 [cit. 2018-03-25]. ISBN 978-4-9906441-0-9. ISSN 1051-4651.

[52] KASAR, Thotreingam a Angarai G. RAMAKRISHNAN. M*ulti-script and multi-oriented text localization from scene images* [online]. Bangalore, 2011 [cit. 2018-03-25]. Dostupné z: https://core.ac.uk/download/pdf/11662911.pdf

[53] MATAS, J., O. CHUM, M. URBAN a T. PAJDLA. Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. *Proceedings of the British Machine Vision Conference 2002* [online]. British Machine Vision Association, 2002, 36.1-36.10 [cit. 2018-03-30]. DOI: 10.5244/C.16.36. ISBN 1-901725-19-7. Dostupné z: http://www.bmva.org/bmvc/2002/papers/113/index.html

[54] NEUMANN, Lukáš a Jiří MATAS. A Method for Text Localization and Recognition in Real-World Images. *Computer Vision – ACCV 2010* [online]. 2010, 2011, 770-783 [cit. 2018-03-30]. Lecture Notes in Computer Science. DOI: 10.1007/978-3-642-19318-7_60. ISBN 978-3-642-19317-0. Dostupné z: http://waltz.felk.cvut.cz/~matas/papers/neumann-text-accv10.pdf

[55] GONZÁLEZ, Álvaro, Luis M. BERGASA, J. Javier YEBES a Sebastián BRONTE. Text location in complex images. *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)* [online]. Tsukuba, 2012, 617-620 [cit. 2018-03-30]. ISSN 1051-4651. Dostupné z: http://www.robesafe.es/personal/alvaro/papers/0106.pdf

[56] HUANG, Weilin, Yu QIAO a Xiaoou TANG. Robust Scene Text Detection with Convolution Neural Network Induced MSER Trees. *European Conference on Computer Vision (ECCV 2014)* [online]. 2014, 497–511 [cit. 2018-03-30]. Dostupné z: http://www.whuang.org/papers/whuang2014_eccv.pdf

[57] MISHRA, Anand, Karteek ALAHARI a C. V. JAWAHAR. Top-down and bottom-up cues for scene text recognition. 20*12 IEEE Conference on Computer Vision and Pattern Recognition* [online]. IEEE, 2012, 2687-2694 [cit. 2018-03-30]. DOI: 10.1109/CVPR.2012.6247990. ISBN 978-1-4673-1228-8. ISSN 1063-6919. Dostupné z: http://www.di.ens.fr/willow/pdfscurrent/mishra12.pdf

[58] DALAL, Navneet a Bill TRIGGS. Histograms of Oriented Gradients for Human Detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)* [online]. IEEE, 2005, 886-893 [cit. 2018-03-30]. DOI: 10.1109/CVPR.2005.177. ISBN 0-7695-2372-2. Dostupné z: https://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf

[59] GUPTA, Ankush, Andrea VEDALDI a Andrew ZISSERMAN. Synthetic Data for Text Localisation in Natural Images. *2016 IEEE Conference on Computer Vision and*

Pattern Recognition (CVPR) [online]. IEEE, 2016, 2315-2324 [cit. 2018-03-30]. DOI: 10.1109/CVPR.2016.254. ISBN 978-1-4673-8851-1. Dostupné z: https://arxiv.org/abs/1604.06646

[60] REDMON, Joseph, Santosh DIVVALA, Ross GIRSHICK a Ali FARHADI. You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* [online]. IEEE, 2016, 779-788 [cit. 2018-03-30]. DOI: 10.1109/CVPR.2016.91. ISBN 978-1-4673-8851-1. Dostupné z: https://arxiv.org/abs/1506.02640

[61] JADERBERG, Max, Andrea VEDALDI a Andrew ZISSERMAN. *Deep Features for Text Spotting* [online]. University of Oxford, 2014 [cit. 2018-03-31]. Dostupné z: http://www.robots.ox.ac.uk/~vedaldi//assets/pubs/jaderberg14deep.pdf

[62] YAO, Cong, Xiang BAI, Baoguang SHI a Wenyu LIU. Strokelets: A Learned Multi-scale Representation for Scene Text Recognition. *2014 IEEE Conference on Computer Vision and Pattern Recognition* [online]. IEEE, 2014, 2014, 4042-4049 [cit. 2018-03-31]. DOI: 10.1109/CVPR.2014.515. ISBN 978-1-4799-5118-5. ISSN 1063-6919. Dostupné z: https://pdfs.semanticscholar.org/9e09/04828d2ef3e60ffbaf705f4c70bc7a880e09.pdf

[63] BREIMAN, Leo. *Random forests* [online]. University of California, 2001 [cit. 2018-03-31]. Dostupné z: https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf

[64] WANG, Tao, David J. WU, Adam COATES a Andrew Y. NG. End-to-End Text Recognition with Convolutional Neural Networks. *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)* [online]. IEEE, 2012, 3304-3308 [cit. 2018-03-31]. ISSN 1051-4651. Dostupné z: https://crypto.stanford.edu/~dwu4/papers/TextRecogCNN.pdf

[65] ALSHARIF, Ouais a Joelle PINEAU. *End-to-End Text Recognition with Hybrid HMM Maxout Models* [online]. 2013 [cit. 2018-03-31]. Dostupné z: https://arxiv.org/abs/1310.1811

[66] COATES, Adam, Blake CARPENTER, Carl CASE, Sanjeev SATHEESH, Bipin SURESH, Tao WANG, David J. WU a Andrew Y. NG. Text Detection and Character Recognition in Scene Images with Unsupervised Feature Learning. *2011 International Conference on Document Analysis and Recognition* [online]. IEEE, 2011, 440-445 [cit. 2018-03-31]. DOI: 10.1109/ICDAR.2011.95. ISBN 978-1-4577-1350-7. ISSN 2379-

2140. Dostupné z: http://ai.stanford.edu/~ang/papers/icdar01-TextRecognitionUnsupervisedFeatureLearning.pdf

[67] SAIDANE, Zohra a Christophe GARCIA. Automatic Scene Text Recognition using a Convolutional Neural Network. *Proceedings of the Second International Workshop on Camera-Based Document Analysis and Recognition* [online]. 2007, 100-106 [cit. 2018-03-31]. Dostupné z: http://www.m.cs.osakafu-u.ac.jp/cbdar2007/proceedings/papers/P6.pdf

[68] JADERBERG, Max, Karen SIMONYAN, Andrea VEDALDI a Andrew ZISSERMAN. *Synthetic Data and Artificial Neural Networks for Natural Scene Text Recognition* [online]. 2014 [cit. 2018-03-31]. Dostupné z: https://arxiv.org/abs/1406.2227

[69] LECUN, Yann, Corinna CORTES a Christopher J.C. BURGES. *The MNIST database of handwritten digits* [online]. 1998 [cit. 2018-03-31]. Dostupné z: http://yann.lecun.com/exdb/mnist/

[70] FERGUS, Rob. Deep Learning for Computer Vision. *Neural Information Processing Systems (NIPS)* [online]. 2013 [cit. 2018-03-31]. Dostupné z: http://cs.nyu.edu/~fergus/presentations/nips2013_final.pdf

[71] LECUN, Yann, León BOTTOU, Yoshua BENGIO a Patrick HAFFNER. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* [online]. 1988, **86**(11), 2278-2324 [cit. 2018-04-06]. DOI: 10.1109/5.726791. ISSN 00189219. Dostupné z: http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf

[72] CANZIANI, Alfredo, Adam PASZKE a Eugenio CULURCIELLO. *An Analysis of Deep Neural Network Models for Practical Applications* [online]. 2016 [cit. 2018-04-13]. Dostupné z: https://arxiv.org/abs/1605.07678

[73] SZEGEDY, Christian, Wei LIU, Yangqing JIA, et al. *Going Deeper with Convolutions* [online]. 2014 [cit. 2018-04-13]. Dostupné z: https://arxiv.org/abs/1409.4842

[74] SIMONYAN, Karen a Andrew ZISSERMAN. *Very Deep Convolutional Networks for Large-Scale Image Recognition* [online]. 2014 [cit. 2018-04-13]. Dostupné z: https://arxiv.org/abs/1409.1556

[75] HE, Kaiming, Xiangyu ZHANG, Shaoqing REN a Jian SUN. *Deep Residual Learning for Image Recognition* [online]. 2015 [cit. 2018-04-13]. Dostupné z: https://arxiv.org/abs/1512.03385

[76] SZEGEDY, Christian, Sergey IOFFE, Vincent VANHOUCKE a Alex ALEMI. *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning* [online]. 2016 [cit. 2018-04-13]. Dostupné z: https://arxiv.org/abs/1602.07261

[77] NANDA, Yugandhar. What is the VGG neural network? In: *Quora* [online]. 1 Mar 2018 [cit. 2018-04-13]. Dostupné z: https://www.quora.com/What-is-the-VGG-neural-network

[78] Osobní doklady. *Ministerstvo vnitra České republiky* [online]. c2018, 5. března 2018 [cit. 2018-04-13]. Dostupné z: http://www.mvcr.cz/clanek/osobni-doklady-642319.aspx

[79] ISO/IEC 7810. *Identification cards: Physical characteristics*. 3rd. Switzerland, 2003.

[80] Občanský průkaz. *Wikipedie* [online]. 2001, 10. 1. 2018 [cit. 2018-04-13]. Dostupné z: https://cs.wikipedia.org/wiki/Ob%C4%8Dansk%C3%BD_pr%C5%AFkaz

[81] ISO 3166-1. *Codes for the representation of names of countries and their subdivisions*. 3rd. Switzerland, 2013.

[82] ISO 1073-2. *Alphanumeric character sets for optical recognition*. 1976.

[83] *Doc 9303: Machine Readable Travel Documents* [online]. 7th. International Civil Aviation Organization, 2015 [cit. 2018-04-13]. ISBN 978-92-9249-790-3. Dostupné z: https://www.icao.int/publications/pages/publication.aspx?docnum=9303

[84] Cestovní pas České republiky. *Wikipedie* [online]. 2001, 23. 1. 2018 [cit. 2018-04-13]. Dostupné z: https://cs.wikipedia.org/wiki/Cestovn%C3%AD_pas_%C4%8Cesk%C3%A9_republiky

[85] ISO/IEC 15438. *Automatic identification and data capture techniques: PDF417 bar code symbology specification*. 3rd. 2015.

[86] ČESKO. § 3 zákona č. 328/1999 Sb., o občanských průkazech. In: *Zákony pro lidi.cz* [online]. © AION CS 2010-2018 [cit. 21. 3. 2018]. Dostupné z: https://www.zakonyprolidi.cz/cs/1999-328#p3

[87] ČESKO. § 5 odst. 2 zákona č. 329/1999 Sb., o cestovních dokladech a o změně zákona č. 283/1991 Sb., o Policii České republiky, ve znění pozdějších předpisů, (zákon o cestovních dokladech). In: *Zákony pro lidi.cz* [online]. © AION CS 2010-2018 [cit. 24. 3. 2018]. Dostupné z: https://www.zakonyprolidi.cz/cs/1999-329#p5-2

[88] MRZ Recognition. *Google Play* [online]. Google, c2018, 19 June 2017 [cit. 2018-04-15]. Dostupné z: https://play.google.com/store/apps/details?id=biz.smartengines.mrzrecognition

[89] *Accura Scan* [online]. AccuraTechnolabs [cit. 2018-04-15]. Dostupné z: https://accurascan.com/

[90] *Regula Document Reader* [online]. Regula, c2017 [cit. 2018-04-15]. Dostupné z: https://mobile.regulaforensics.com/

[91] IDscan Mobile App. *IDscan* [online]. Idscan Biometrics, c2018 [cit. 2018-04-15]. Dostupné z: https://www.idscan.com/idscan-app/

[92] AcuFill. *Acuant* [online]. c2018 [cit. 2018-04-15]. Dostupné z: https://www.acuantcorp.com/autofill-software/

[93] Netverify. *Jumio* [online]. c2010-2018 [cit. 2018-04-15]. Dostupné z: https://www.jumio.com/trusted-identity/netverify/

[94] Smart ID Reader. *Smart Engines* [online]. c2010-2018 [cit. 2018-04-15]. Dostupné z: http://smartengines.biz/smart-id-reader/

[95] IdScan GO. *Google Play* [online]. Google, c2018, 12 April 2018 [cit. 2018-04-15]. Dostupné z: https://play.google.com/store/apps/details?id=com.cssn.idscango

[96] Jumio Showcase. *Google Play* [online]. Google, c2018, 23 January 2018 [cit. 2018-04-15]. Dostupné z: https://play.google.com/store/apps/details?id=com.jumio.demo.netverify

[97] Smart IDReader. *Google Play* [online]. Google, c2018, 23 March 2018 [cit. 2018-04-15]. Dostupné z: https://play.google.com/store/apps/details?id=biz.smartengines.smartid

[98] BAY, Herbert, Tinne TUYTELAARS a Luc VAN GOOL. SURF: Speeded Up Robust Features. *Computer Vision – ECCV 2006* [online]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, 2006, 404-417 [cit. 2018-04-16]. Lecture Notes in Computer Science. DOI: 10.1007/11744023_32. ISBN 978-3-540-33832-1. Dostupné z: http://www.vision.ee.ethz.ch/~surf/eccv06.pdf

[99] RUBLEE, Ethan, Vincent RABAUD, Kurt KONOLIGE a Gary BRADSKI. ORB: An efficient alternative to SIFT or SURF. *2011 International Conference on Computer Vision* [online]. IEEE, 2011, 2564-2571 [cit. 2018-04-16]. DOI: 10.1109/ICCV.2011.6126544. ISBN 978-1-4577-1102-2. Dostupné z: http://www.willowgarage.com/sites/default/files/orb_final.pdf

[100] LEUTENEGGER, Stefan, Margarita CHLI a Roland Y. SIEGWART. BRISK: Binary Robust invariant scalable keypoints. *2011 International Conference on Computer Vision* [online]. IEEE, 2011, 2548-2555 [cit. 2018-04-16]. DOI:

10.1109/ICCV.2011.6126542. ISBN 978-1-4577-1102-2. Dostupné z: https://www.robots.ox.ac.uk/~vgg/rg/papers/brisk.pdf

[101] HARRIS, Chris a Mike STEPHENS. A Combined Corner and Edge Detector. *Proceedings of the Alvey Vision Conference 1988* [online]. Alvey Vision Club, 1988, 147-152 [cit. 2018-04-16]. DOI: 10.5244/C.2.23. Dostupné z: http://www.bmva.org/bmvc/1988/avc-88-023.pdf

[102] ROSTEN, Edward a Tom DRUMMOND. Machine Learning for High-Speed Corner Detection. *Computer Vision – ECCV 2006* [online]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, 430-443 [cit. 2018-04-16]. Lecture Notes in Computer Science. DOI: 10.1007/11744023_34. ISBN 978-3-540-33832-1. Dostupné z: https://www.edwardrosten.com/work/rosten_2006_machine.pdf

[103] ALAHI, Alexandre, Raphael ORTIZ a Pierre VANDERGHEYNST. FREAK: Fast Retina Keypoint. *2012 IEEE Conference on Computer Vision and Pattern Recognition* [online]. IEEE, 2012, 2012, 510-517 [cit. 2018-04-16]. DOI: 10.1109/CVPR.2012.6247715. ISBN 978-1-4673-1228-8. Dostupné z: https://infoscience.epfl.ch/record/175537/files/2069.pdf

[104] FUNAYAMA, Ryuji, Hiromichi YANAGIHARA, Luc VAN GOOL, Tinne TUYTELAARS a Herbert BAY. *Robust Interest Point Detector and Descriptor*. US20070298879 20070430. Dostupné také z: https://worldwide.espacenet.com/publicationDetails/biblio?CC=US&NR=2009238460&KC=&FT=E&locale=en_EP

[105] ALCANTARILLA, Pablo Fernández, Adrien BARTOLI a Andrew J. DAVISON. KAZE Features. *Computer Vision – ECCV 2012* [online]. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, 214-227 [cit. 2018-04-16]. Lecture Notes in Computer Science. DOI: 10.1007/978-3-642-33783-3_16. ISBN 978-3-642-33782-6. Dostupné z: https://www.doc.ic.ac.uk/~ajd/Publications/alcantarilla_etal_eccv2012.pdf

[106] NEUMANN, Lukáš a Jiří MATAS. Real-time scene text localization and recognition. 2012 IEEE Conference on Computer Vision and Pattern Recognition [online]. IEEE, 2012, 2012, 3538-3545 [cit. 2018-04-16]. DOI: 10.1109/CVPR.2012.6248097. ISBN 978-1-4673-1228-8. Dostupné z: http://cmp.felk.cvut.cz/~neumalu1/neumann-cvpr2012.pdf

[107] Scene Text Detection. *OpenCV Documentation* [online]. c2011-2014 [cit. 2018-04-16]. Dostupné z: https://docs.opencv.org/3.0-beta/modules/text/doc/erfilter.html

[108] File Exchange. *MatchWorks* [online]. Natick (Massachuttes), c1994-2018 [cit. 2018-04-16]. Dostupné z: http://www.mathworks.com/matlabcentral/fileexchange/

[109] Tesseract OCR. *GitHub* [online]. San Francisco, c2018 [cit. 2018-04-16]. Dostupné z: https://github.com/tesseract-ocr/tesseract

[110] Java OCR. *SourceForge* [online]. Slashdot Media, c2018 [cit. 2018-04-16]. Dostupné z: https://sourceforge.net/projects/javaocr/

[111] *Ocrad: The GNU OCR* [online]. Boston: Free Software Foundation, c2016 [cit. 2018-04-16]. Dostupné z: https://www.gnu.org/software/ocrad/

[112] OCRopus. *GitHub* [online]. San Francisco, c2018 [cit. 2018-04-16]. Dostupné z: https://github.com/tmbdev/ocropy

[113] BERNSEN, J. Dynamic Thresholding of Grey-level Images. *8th International Conference on Pattern Recognition* [online]. Paris, 1986, 1251-1255 [cit. 2018-04-01].

[114] KHURSHID, Khurram, Kathrin BERKNER, Laurence LIKFORMAN-SULEM, Imran SIDDIQI, Claudie FAURE a Nicole VINCENT. Comparison of Niblack inspired binarization methods for ancient documents. In: *Document Recognition and Retrieval XVI* [online]. 2009 [cit. 2018-04-01]. DOI: 10.1117/12.805827.

[115] WOLF, Christian a Jean-Michel JOLION. Extraction and recognition of artificial text in multimedia documents. *Formal Pattern Analysis & Applications* [online]. 2004, **6**(4), 309–326 [cit. 2018-04-01]. DOI: 10.1007/s10044-003-0197-7. ISSN 1433-7541. Dostupné z: http://liris.cnrs.fr/Documents/Liris-753.pdf

[116] *TensorFlow* [online]. [cit. 2018-04-05]. Dostupné z: https://www.tensorflow.org/

[117] IOFFE, Sergey a Christian SZEGEDY. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift* [online]. 2015 [cit. 2018-04-04]. Dostupné z: https://arxiv.org/abs/1502.03167

[118] *Caffe: Deep learning framework* [online]. [cit. 2018-04-05]. Dostupné z: http://caffe.berkeleyvision.org/

[119] *NVIDIA DIGITS: Interactive Deep Learning GPU Training System* [online]. c2018 [cit. 2018-04-05]. Dostupné z: https://developer.nvidia.com/digits

[120] Četnost jmen a příjmení. *Ministerstvo vnitra České republiky* [online]. c2018, 5. září 2017 [cit. 2018-04-06]. Dostupné z: http://www.mvcr.cz/clanek/cetnost-jmen-a-prijmeni-722752.aspx

[121] Adresní místa RÚIAN ve formátu CSV. *ČÚZK Nahlížení do katastru nemovitostí* [online]. c2004-2018 [cit. 2018-04-06]. Dostupné z: http://nahlizenidokn.cuzk.cz/StahniAdresniMistaRUIAN.aspx

[122] LEVENSHTEIN, Vladimir I. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics - doklady* [online]. 1966, **10**(8), 707-710 [cit. 2018-04-06]. Dostupné z: https://nymity.ch/sybilhunting/pdf/Levenshtein1966a.pdf

[123] HARTL, Andreas, Clemens ARTH a Dieter SCHMALSTIEG. Real-time Detection and Recognition of Machine-Readable Zones with Mobile Devices. *VISSAP 2015* [online]. 2015 [cit. 2018-04-07]. Dostupné z: https://pdfs.semanticscholar.org/e4d6/0075b43f4e3d881482b5d744a85ca0142967.pdf

[124] Performance Tradeoff - When is MATLAB better/slower than C/C++. In: *Stack Overflow* [online]. 2008, 8 July 2014 [cit. 2018-04-13]. Dostupné z: https://stackoverflow.com/questions/20513071/performance-tradeoff-when-is-matlab-better-slower-than-c-c#24643248

[125] HE, Kaiming, Xiangyu ZHANG, Shaoqing REN a Jian SUN. *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification* [online]. 2015 [cit. 2018-04-13]. Dostupné z: https://arxiv.org/abs/1502.01852

[126] MATLAB API for Python. *MathWorks* [online]. Natick (Massachuttes), c1994-2018 [cit. 2018-04-13]. Dostupné z: https://www.mathworks.com/help/matlab/matlab-engine-for-python.html