Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra kybernetiky

Bakalářská práce

Rozpoznávání objektů v obrazech a videosekvencích

Autor:

Milan Herbig

Vedoucí práce:

Ing. Pavel Campr, Ph.D.

Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne 12. května 2014

Podpis: _____

Poděkování

Rád bych poděkoval vedoucímu své bakalářské práce Ing. Pavlu Camprovi, Ph.D. za odborné vedení, podnětné konzultace, ochotnou pomoc a trpělivost při vypracování této práce.

Děkuji MetaCentru za přístup k výpočetnímu prostředí, ve kterém probíhaly veškeré výpočty. MetaCentrum je součástí programu "Projects of Large Infrastructure for Research, Development, and Innovations" (LM2010005).

Abstrakt

Téma práce je zaměřeno na rozpoznávání a klasifikaci obrazu ve videosekvencích, konkrétně na rozpoznání a zařazení snímků videa dle obsahu do předem známé kategorie. Práce se zabývá extrakcí příznaků jednak kombinací jednodušších metod a jednak komplexní metodou využívající konvoluční neuronovou síť. Získané příznaky jsou posléze klasifikovány pomocí Random Forest a SVM klasifikátoru.

Cílem práce je porovnat různé kombinace jednodušších i komplexnějších příznaků pro vybrané kategorie videí. Výstupem práce jsou statistiky úspěšnosti různých kombinací extrahovaných příznaků.

Klíčová slova: zpracování digitalizovaného obrazu, extrakce příznaků, detekce objektů, klasifikace obrazu, konvoluční neuronová síť

Abstract

The work is focused on classification of video footages to correct category by content. Two different approaches are used to describe images. The first creates ensemble from simple feature extractors and the second uses more complex description by convolution neural network which achieves state-of-the-art performance in common classification tasks. Extracted features are classified by Random Forest and SVM classifiers.

The goal is to compare combinations of simple features compared to more complex features extracted from neural network. Results present classification rate comparison of several combinations of extracted features and classifiers.

Keywords: image processing, feature extraction, object detection, image classification, convolution neural network

Obsah

1 Úvod

Počítačové zpracování a analýza obrazu hraje velice důležitou roli pro oblasti počítačového vidění a strojového učení. Za rychlou expanzí tohoto oboru stojí jednak rostoucí výpočetní síla a technologický pokrok a jednak neustále narůstající nároky na automatizaci a nahrazení úlohy člověka. Vzhledem k faktu, že velikou část objemu získaných informací z našeho okolí tvoří u člověka právě zrak, je počítačové zpracování a analýza obrazu nenahraditelnou součástí umělé inteligence. Širokého uplatnění tak nachází tento obor nejen ve vědě, průmyslu, či v medicíně, ale stále více se aplikace umělé inteligence objevuje v běžné spotřební elektronice.

Pro účely dalších projektů na katedře kybernetiky se tato práce zaměřuje na extrakci a klasifikaci obrazu scény ve spojitých videozáznamech. Pojmem obrazu v tomto případě rozumíme informace extrahované z originálního snímku videa reprezentující a popisující vhodným způsobem tento snímek ve formátu vektoru čísel. Způsoby extrakce těchto tzv. příznakových vektorů se odvíjí od charakteru úlohy a dostupných dat. Jejich vhodná volba zásadně ovlivňuje dosažené výsledky, a tedy tvoří důležitou část úlohy. Úkolem systému je posléze natrénovat z dostupných trénovacích dat klasifikátor na základě dané informace o příslušnosti těchto obrazů, tedy vytvořit mechanismus, jež má najít společné rysy mezi daty, které by nejlépe posloužily pro jejich reprezentaci, a tedy později ke klasifikaci doposud neznámých dat. Výstupem systému je informace o zařazení nových – testovacích dat, na základě nichž můžeme posoudit úspěšnost klasifikace a zároveň posoudit kvalitu navrženého systému. Tato výstupní data mohou být rovněž vodítkem k dalšímu zlepšení systému.

2 Motivace

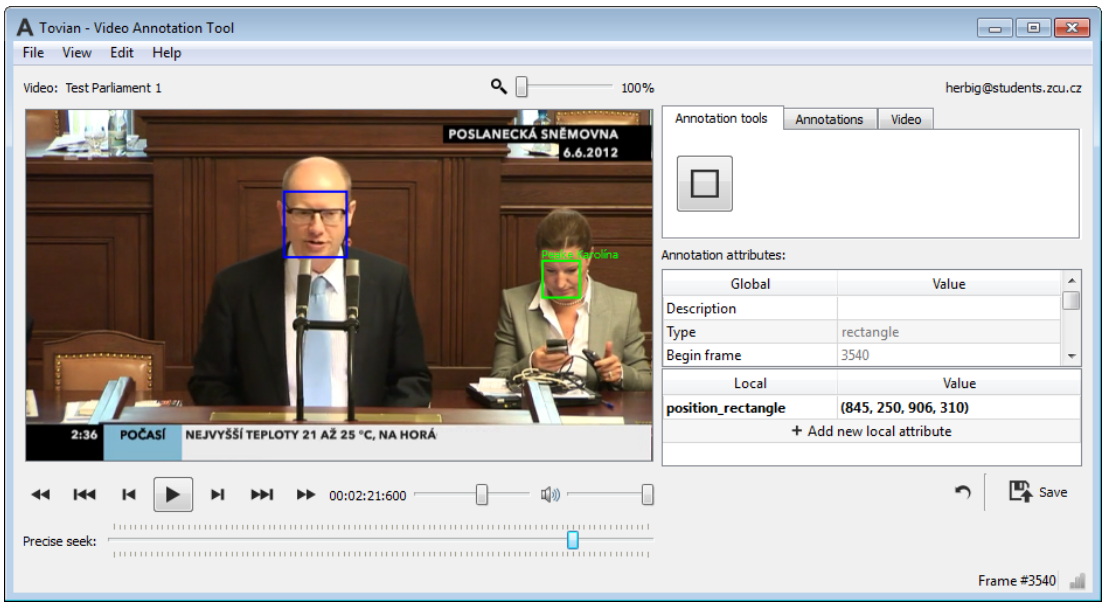
Úkolem počítačového zpracování a analýzy obrazu je zprostředkovat nadřazenému systému informace podobné těm, které získává člověk zrakem. Získané informace pak mohou sloužit například k orientaci a navigaci v prostoru, ke klasifikaci a rozhodování, k odvozování nových informací, atp. Příkladem dnes již běžně fungujících systémů může být automatická detekce obličejů na fotografii, detekce chodců a překážek v automobilech, detekce dopravních značek, detekce gest a pohybu částí těla, výstupní vizuální kontrola výrobků, ale i automatická tvorba panoramatických snímků či dálkový průzkum země.

Systém vyvíjený v této práci se zaměřuje na extrakci informací spíše na vyšší úrovni, které by nejlépe popsaly konkrétní scénu. Tento systém pak může posloužit k automatické anotaci nebo prohledávání obrovského množství videozáznamu, přičemž získanou informaci lze využít dalšími systémy automatické detekce k dosažení lepších výsledků. Konkrétně využitím informace o typu nebo změně videa systémem analýzy řeči lze nasadit hlasové modely uzpůsobené určitému typu videa. Jestliže tedy během televizního vysílání dojde ke změně programu z hokejového zápasu na poslaneckou sněmovnu, titulkovací systém může dynamicky zareagovat změnou parametrů hlasových modelů, a tak výrazně zvýšit svou přesnost.

Praktickým příkladem použití systému tohoto typu může být automatická anotace obrazu. Spolu s Ing. Pavlem Camprem, Ph.D. vyvíjíme pro katedru kybernetiky anotační aplikaci Tovian[1]. Tovian je nástroj vytvořený pro manuální anotaci audiovizuálního obsahu, respektive anotaci objektů a scén.

[1]Aplikace je volně dostupná na `tovian.zcu.cz`

Účelem aplikace je poskytnutí referenčních manuálně anotovaných dat. Anotovaná data jsou tak zdrojem či zpětnou vazbou algoritmů automatické detekce nebo klasifikace, konkrétně například dopravního značení, obličejů, či objektů.



Obrázek 2.1: Anotace obličejů poslanců v anotační aplikaci Tovian. Anotační nástroje jsou zároveň objekty obsahující předem nadefinované atributy, například pohlaví nebo jméno osoby.

Navrhovaný systém v této práci pak může posloužit pro automatickou anotaci scény, tedy nahrazení práce lidského anotátora. Popřípadě může být systém dále zdokonalen aplikací algoritmů strojového učení a použitím ve formě automatického našeptávače během anotace. Tedy systém se bude dále učit a zdokonalovat na základě činnosti anotátora.

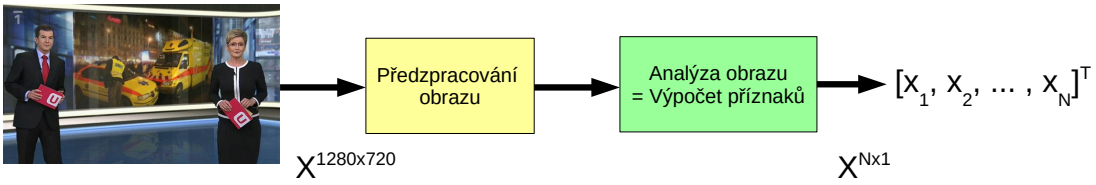
3 Analýza obrazu

K analýze obrazu můžeme přistoupit dvěma způsoby. V prvním přístupu máme k dispozici apriorní informaci o typu a charakteru obrazu a zajímá nás jen určitá informace, kterou se snažíme za pomoci aplikace výpočetních metod nalézt. Typicky jde o hledání a lokalizaci konkrétních objektů, přičemž je pro nás zbytek obrazu nezajímavý. Často máme navíc k dispozici exaktní popis hledaného objektu. Uplatňují se tak často metody předzpracování obrazu a segmentace.

Druhý přístup se snaží o pohled na obraz jako celek. Podobně, jako to zvládá člověk, se snažíme z obrazu extrahovat jeho výstižný popis, tudíž pro nás nehraje příliš roli šum nebo deformace obrazu. Tato vysokoúrovňová informace charakteristická pro každou skupinu snímků pak slouží jako kritérium pro jejich klasifikaci. Takovou informací může být rozložení barev, výskyt konkrétních objektů, či výskyt pohybu a prudkých změn v obraze pro spojité videozáznamy.

3.1 Extrakce příznaků

Z každého získaného snímku jsou konkrétní metodou extrahovány tzv. příznaky, které jsou ve výsledku spojeny do jediného příznakového vektoru. Tento číselný vektor slouží jako popis původního snímku. Dochází tak k výrazné redukci dat, díky které je vůbec dále možné použít další výpočetní metody. Volba těchto příznaků je velice důležitá a má přímý dopad nejen na kvalitu výsledků, ale zejména na složitost a výpočetní náročnost systému. Je naším cílem tedy najít kompromis mezi množstvím dat a jejich vypovídající hodnotou.



Obrázek 3.1: Výpočtem příznaků z originálního snímku dochází zejména k výrazné redukci dat.

V případech, kdy se nelze vyhnout vysoké dimenzi dat, se mnohdy uplatňují metody redukce počtu příznaků, jenž analýzou buďto vyloučí mezi daty vzájemně si podobné příznaky, které prakticky nemají žádnou vypovídající hodnotu, nebo za pomoci transformačního předpisu transformují příznaky do nižší dimenze. S rostoucí dimenzí totiž roste složitost trénování klasifikátoru. Pro velmi vysoké dimenze tak natrénování klasifikátoru nemusí být z důvodu vysokých paměťových nebo časový nároků vůbec možné.

3.2 Histogram barev

Histogram je jednou z nejzákladnějších metod počítačové analýzy obrazu. Pro černobílý obraz vyjadřuje histogram četnost jednotlivých diskrétních jasových intenzit z v celém obraze. Výsledkem je 2D informace, kolikrát se konkrétní jasová hodnota v daném snímku vyskytuje. Pro obraz barevný je pak tato četnost vyjádřená pro jednotlivé barevné složky.

Pro účely redukce dat se mnohdy osa x rozděluje pouze do k binů (buněk) [1], kdy v jednom binu $N_b^{(i)}$ je suma četností jasových hodnot $h^{(j)}$ na určitém intervalu. Délka l tohoto intervalu je dána podílem maximální hodnoty jasu M počtem binů k.

$$n = \sum_{i=1}^{k} N_b^{(i)} = \sum_{i=0}^{k-1} \sum_{j=i\cdot l}^{i\cdot l+l} h^{(j)} = \sum_{j=0}^{M} h^{(j)} \tag{3.1}$$

Získanou informaci o rozložení jasu můžeme po normalizaci interpretovat jako rozdělení pravděpodobnosti. Tedy lze vyjádřit pravděpodobnost $p(z, [x, y])$ výskytu jasové hodnoty z na souřadnicích $[x, y]$. Histogram je obvykle jediná globálních informace o obrazu nezávisející na jeho orientaci a velikosti. Typickým použitím histogramu je hledání lokálních jasových extrémů, které následně pomohou při odstranění šumu nebo nastavení prahů při segmentaci. Při snímání obrazu je potom obvykle informace o rozložení jasu využívána jako zpětná vazba k nastavení citlivosti snímače.

Pro popis barevného rozložení snímku je extrahována informace o dominantních barvách a jasových hodnotách. Ty lze získat převodem snímku do HSV prostoru a výpočtem histogramu. Histogram je posléze zredukován na k binů a normalizován tak, aby odpovídal pravděpodobnostnímu rozdělení.

3.2.1 HSV barevný prostor

HSV (Hue, Saturation, Value) prostor se skládá ze složek odstínu, sytosti a světlosti. Mnohdy se používá pro segmentaci objektu určité barvy, protože pro specifikaci hledané barvy lze použít spojité intervaly jednotlivých složek, čímž dojde k pokrytí různých jasových variant hledané barvy, a tedy zajištění částečné robustnosti. Výhodou je získaná informace, zda je barva sytá a jasná, či méně sytá a tmavá.

3.3 Změna obrazu mezi snímky

Úlohou detekce střihu je najít ve videozáznamu čísla snímků, kde došlo ke střihu. Střih se typicky vyznačuje výraznou změnou scény mezi dvěma sousedními snímky, respektive výrazným jasovým rozdílem pixelů na stejné pozici.

Pro extrakci informace o velikosti změny obrazu a detekce střihu je využit výpočet Sum of absolute differences (SAD) vždy pro dva sousední snímky [2]. Pro každou dvojici snímků k a $k-1$ je vypočítána absolutní hodnota rozdílu obrazových bodů $p_{x,y}$ na stejné pozici. Výsledná suma je dána součtem všech absolutních rozdílů pro každý celkový počet řádků $n^{(r)}$ a sloupců $n^{(c)}$.

$$SAD_k = \sum_{x=1}^{n^{(r)}} \sum_{y=1}^{n^{(c)}} |p_{x,y}^{(k)} - p_{x,y}^{(k-1)}| \,; \qquad k = 2, 3, \ldots, n^{(f)} \tag{3.2}$$

K rozhodnutí, zda došlo k dostatečně velké změně obrazu reprezentující střih, dojde porovnáním vypočtených rozdílů s dodanou prahovou hodnotou určenou při trénování algoritmu. Úspěšnost detekce střihů odpovídá dynamice videozáznamu, respektive vhodné volbě prahové hodnoty.

3.3.1 Četnost střihů

Pro různé kategorie videozáznamu je typická jejich odlišná dynamika. Zatímco pro sportovní přenosy či filmy je typická větší frekvence střihů, pro diskusní pořady či záznamy z poslanecké sněmovny je tomu naopak. Tato informace tedy může posléze posloužit jako jednoduchý příznak pro klasifikaci. Je tedy vhodné najít způsob, jak pro popis jednoho diskrétního snímku využít informaci ze spojitého videozáznamu.

Jedním z nejjednodušších způsobů, jak tuto informaci vyjádřit, je spočítat množství střihů za posledních n snímků. Abychom tento příznak dále znormalizovali, vyjádříme jej podílem sumy střihů počtem snímků n. Získanou informaci lze tak interpretovat jako pravděpodobnost, že v konkrétním snímku dojde ke střihu.

$$CR_k = \frac{\sum_{i=k-n}^{k} c^{(i)}}{n} \,; \qquad k = 1, 2, \ldots, n^{(f)} \tag{3.3}$$

$$c^{(i)} = \begin{cases} 1 & \text{pokud střih} \\ 0 & \text{jinak} \end{cases} \tag{3.4}$$

3.3.2 Velikost změny obrazu

Protože při detekci střihů extrahujeme informaci o změně obrazu $m^{(i)}$, lze použít tuto informaci rovněž jako příznak pro popis měnící se scény. Zprůměrováním této hodnoty přes l snímků zpět dojde k vyhlazení informace a potlačení potencionálního šumu, protože nepředpokládáme, že by se vyjma střihu obraz v rámci jednotek až desítek snímků výrazně měnil. Zajímají nás tedy jen výraznější změny v obraze.

$$MR_k = \frac{\sum_{i=k-l}^{k} m^{(i)}}{l}; \qquad k = 1, 2, \ldots, n^{(f)} \tag{3.5}$$

3.4 Přítomnost obličejů

Motivací detekce obličeje v počítačové analýze obrazu bychom vymysleli mnoho. Přítomnost obličeje a četnost jeho výskytu v tomto případě poslouží jako další příznak pro úspěšnější klasifikaci videa. Vezměme v úvahu například zpravodajství nebo diskusní pořad. Zde můžeme zcela jistě očekávat výskyt obličejů. Ty lze navíc detekovat vcelku úspěšně a zároveň se zde frekvence výskytu obličejů drží přibližně na konstantní hodnotě. Naproti tomu u sportovních přenosů, hraných nebo animovaných filmů může být detekce obličejů složitější, a tedy i četnost výskytu řidší nebo velmi kolísavá.

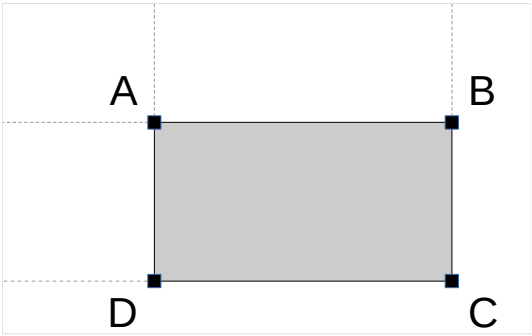
Detekce obličeje je díky jeho rozmanitosti (barva kůže, tvar, atp.) a předem neznámé poloze výskytu v obraze nejednoduchou úlohou, kterou se snaží řešit spousta odlišných metod dosahujících odlišných výsledků. V této práci je využita metoda Haar Feature-based Cascade Classifier [3]. Ta oproti metodám hledajících například odstíny barvy kůže využívá výpočet charakteristických příznaků pro danou oblast v obraze, na jejichž základě je posléze rozhodnuto, zda se jedná nebo nejedná o potencionální část obličeje. Díky implementaci vhodné logiky navíc dosahuje uspokojivé výpočetní náročnosti, čímž je použitelná pro detekci i v reálném čase.

3.4.1 Integrální obraz

V případech, kdy je opakovaně počítána celková intenzita jasu pod danou plochou, je velice efektivní pro výpočet použít integrální obraz. Integrální obraz je matice o stejných rozměrech jako originální obraz. Každý element této matice obsahuje sumu jasových intenzit plochy tvořené počátečním bodem v levém horním rohu a tímto bodem jak ilustruje obrázek 3.2.

$$AI[x, y] = \sum_{\substack{x' \leq x \\ y' \leq y}} A(x', y') \tag{3.6}$$

Tento obraz posléze při výpočtu plochy poslouží jako tzv. lookup table. Platí totiž, že po sestavení integrálního obrazu lze vypočítat libovolně velikou obdélníkovou plochu AI součtem/rozdílem maximálně čtyř hodnot získaných z této tabulky. Tím dochází k obrovskému zefektivnění opakovaného výpočtu celkové sumy jasových intenzit na ploše, protože integrální obraz je třeba sestavit pouze jednou.

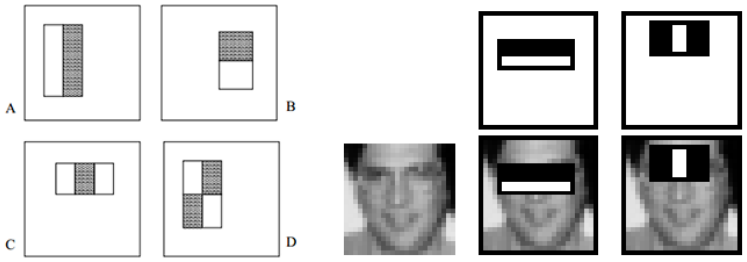


Obrázek 3.2: Výpočet libovolné obdélníkové plochy lze provést jednoduchým sčítáním/odčítáním.

$$AI = \sum_{\substack{x_0 \leq x \leq x_1 \\ y_0 \leq y \leq y_1}} i(x, y) = I(C) + I(A) - I(B) - I(D) \tag{3.7}$$

3.4.2 Haar Feature-based Cascade Classifier

Principem použitého detektoru prezentovaného v [3] je výpočet konkrétních hodnot relativně umístěných příznaků v plovoucím okénku, které vzniknou výpočtem rozdílů sum jasových intenzit pod plochou "příznakových obdélníků"(angl. haar-like features) zobrazených na obrázku 3.3. Tyto hodnoty příznaků pak posuzuje klasifikátor natrénovaný z dostatečného množství pozitivní a negativních vzorků, který rozhodne, zda se může pod plochou okénka nacházet potencionální oblast obličeje. Protože množství kombinací umístění a natočení jednotlivých příznaků je obrovské, algoritmus implementuje logiku značně zefektivňující výpočet jen na nezbytně nutné množství.



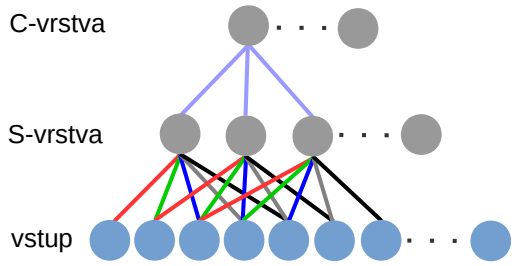
Obrázek 3.3: Grafické znázornění základních haar-like feature boxů. Zdroj: VIOLA, Paul; JONES, Michael; Rapid object detection using a boosted cascade of simple features, 2001.

Při jednotlivých průchodech je v okénku vypočítáno jen určené množství příznaků v závislosti na druhu implementace. Ty jsou následně posuzovány. Při prvním průchodu je vypočten a posouzen pouze jeden příznak. Pokud je vyhodnocen jako nepravděpodobný, celé okénko je zahozeno a výpočet se přesouvá jinam. Pokud jej klasifikátor schválí, je v dalších průchodech vypočítáno více příznaků. Ve výsledku jsou tak zcela přeskočeny oblasti, kde je velice nízká pravděpodobnost výskytu obličeje a naopak jsou podrobně analyzovány části, kde je pravděpodobnost výskytu vysoká. Pro další zvýšení efektivity a přesnosti se samostatné příznaky řadí do kaskád, což souvisí s nabízejícím se rekurzivním charakterem prohledávání oblasti. Tedy je-li detekována pravděpodobná oblast hlavy, lze předpokládat oblast polohy očí, polohy nosu, úst, atp. Citlivost detekce přímo souvisí s parametry klasifikátorů a její nárůst je přímo úměrný nárůstu falešně pozitivních nálezů.

Velikosti plovoucích oken jsou odstupňovány v krocích, čímž je umožněno detekovat jak obličej zabírající celý obraz, tak například větší množství menších obličejů. Volitelná velikost kroku tak ovlivňuje jak citlivost, tak dobu detekce. Pro potřeby vytvářeného systému hraje množství falešně pozitivních nálezů důležitější roli, než několik nedetekovaných obličejů v pozadí, tudíž jsou preferovány méně citlivé parametry.

3.5 Konvoluční neuronové sítě

Konvoluční neuronové sítě jsou novějším typem dopředných neuronových sítí vytvořených převážně pro úlohy počítačového rozpoznávání obrazu rozšiřující základní model Neocognitron [4]. Jednotlivé neurony jsou uspořádány hierarchickým způsobem tak, že reagují pouze na určitý velmi malý zlomek obrazu - okénko (angl. receptive field), přičemž se okénka sousedních neuronů částečně překrývají. Základní inspirací této architektury je biologická funkce mozku savců. Studie zrakového centra mozku odhalila, že na konkrétní tvary objektů reagují vždy konkrétní shluky neuronů [5]. Podrobnější zkoumání ukázalo, že jednotlivé neurony řazené hierarchicky do kaskád zrakového centra mozku opakovaně reagují na konkrétní příznaky vyskytující se v obraze. Fungují tedy jako detektory.

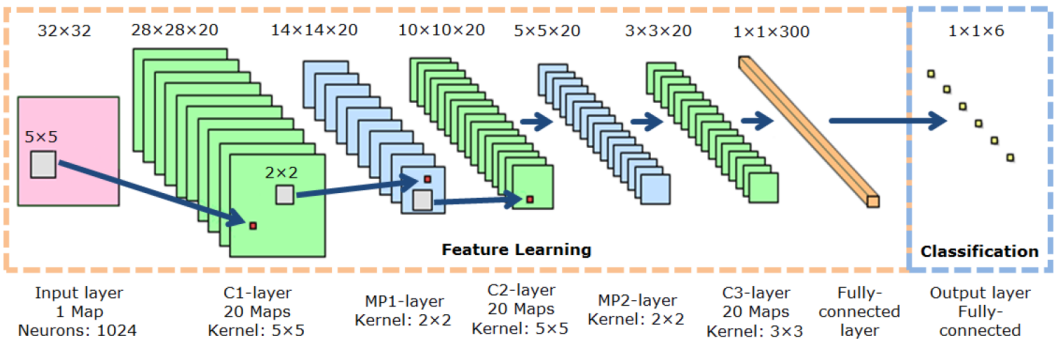


Obrázek 3.4: Ilustrace kombinace S-vrstvy (konvoluce) a C-vrstvy (pooling). Barevně shodné vstupy mají provázané váhy.

Základním prvkem konvoluční sítě jsou S-buňky (angl. simple cell) a C-buňky (angl. complex cell), jejichž způsob propojení ilustruje obrázek 3.4. Extrakci příznaků provádí S-buňky nastavením svých vah. Aktivace neuronu simuluje de-facto

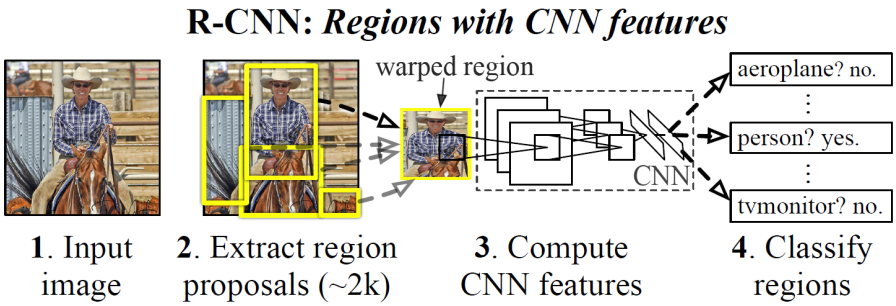
diskrétní konvoluci, kde váhy neuronu zastupují konvoluční jádro. Váhy daného vstupu konkrétního neuronu jsou provázány přes celou vrstvu. Tím je docíleno, že na každý pixel je aplikován stejný filtr pro dané konvoluční jádro (angl. feature map) . Cílem je natrénovat síť tak, aby jednotlivé neurony S-vrstvy reagovaly pouze na určitý příznak v obraze, respektive v okénku – hranu, roh, atp. Tato vrstva posléze tvoří vstup komplexní vrstvy tvořené C-buňkami. Aplikuje se zde tzv. pooling (max-pooling, average-pooling). Každá C-buňka je propojena pouze s několika S-buňkami a její aktivační hodnota je v případě max-poolingu rovna maximální hodnotě na vstupu neuronu. To zajišťuje jistou formu invariance vůči posunu příznaku a vnáší nelinearitu. Dojde-li totiž k posunu obrazu, dojde rovněž k posunu příznaků, avšak nadřazená C-buňka bude stále aktivní. Dochází tak k výrazné redukci dat díky zakódování informace z několika podřazených neuronů do jednoho nadřazeného. Svým způsobem tedy kombinace S-vrstvy a C-vrstvy slouží jako filtry, které filtrují pouze podstatné informace z obrazu.

Kompletní síť se skládá ze vstupní vrstvy, na kterou je přiveden často nikterak nepředzpracovaný vstupní snímek. Skryté vrstvy jsou tvořeny z několika dvojic konvoluční vrstvy a subsamplingové vrstvy (pooling). V závislosti na konkrétní aplikaci může být přidána ještě další vrstva. Nakonec je skrytá vrstva doplněna o několik vrstev plně propojené neuronové sítě, kde se počet neuronů poslední, tedy výstupní vrstvy, liší v závislosti na požadavcích. Trénování konvoluční sítě probíhá algoritmem back-propagation.



Obrázek 3.5: Grafické znázornění konkrétní max-pooling konvoluční neuronové sítě (MPCNN). Zdroj: NAGI, Jawad, et al; Max-Pooling Convolutional Neural Networks for Vision-based Hand Gesture Recognition, 2011

Protože konkrétní neurony reagují na konkrétní příznaky v obraze, hierarchickým skládáním de-facto získáme neurony, které reagují na konkrétní objekty. Přiváděním těchto objektů na vstup můžeme identifikovat, jaké neurony reagují na jaké objekty. Tuto informaci lze posléze použít pro úlohy detekce objektů v obraze. Posouváním okének hledáme maximální aktivační hodnoty, a posléze pro konkrétní polohu v obraze prohlásíme, že se zde nachází obličej, auto, atp. Ze schopnosti přímo identifikovat jednotlivé objekty jasně vyplývá možnost vytvořit rovnou popis celé scény [6].



Obrázek 3.6: Demonstrace detekce jednotlivých objektů v obraze pomocí konvoluční neuronové sítě. Zdroj: GIRSHICK, Ross, et al; Rich feature hierarchies for accurate object detection and semantic segmentation, 2013.

Důkazem schopností konvolučních neuronových sítí je časté nasazení této technologie v praxi. Jako aktuální příklady lze uvést nasazení sítí pro rozpoznávání ručně psaného textu, automatické hledání a rozmazávání obličejů nebo čtení čísel popisných z domků ve službě Google Street View, či například rozpoznávání synteticky deformovaného textu v ověřovací službě CAPTCHA.

4 Klasifikace obrazu

Cílem úlohy klasifikace obrazu je navrhnout a vytvořit mechanismus, který by na základě poskytnutých trénovacích dat nalezl mezi těmito daty souvislost (model), a posléze tak dokázal s co nejmenší chybou rozhodnout o zařazení (třídě) dat nových, neznámých, testovacích. Trénování klasifikátoru provádíme buď to metodou bez učitele nebo s učitelem.

Při aplikaci metody trénování bez učitele nemá klasifikátor žádnou apriorní informaci o původu dat. Samostatně se snaží mezi těmito daty objevit podobnost, nejčastěji na základě euklidovské vzdálenosti. Tato data posléze rozdělí v závislosti na metodě do několika shluků, které ve výsledků odpovídají jednotlivým třídám.

Na druhé straně metoda učení s učitelem má k dispozici apriorní informaci o třídě. Při procesu trénování je cílem, aby klasifikační mechanismus na základě trénovacích dat zobecnil model, a tak posléze dokázal s co nejmenší chybou klasifikovat data testovací.

4.1 Rozdělení dat

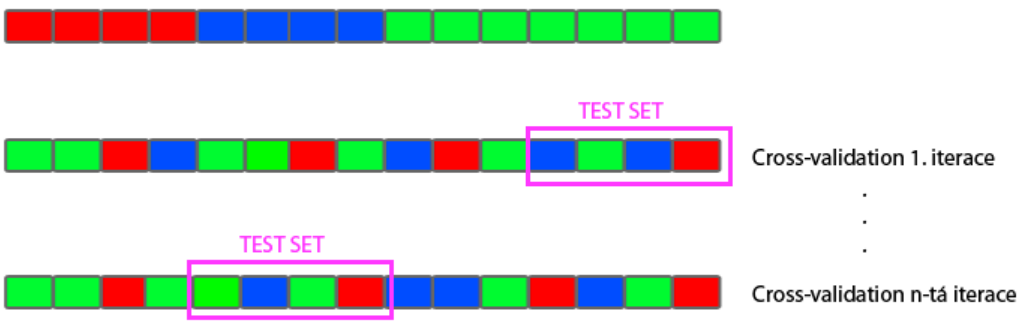
Při trénování se hledá takové nastavení klasifikátoru, aby klasifikátor dokázal klasifikovat trénovací set s co nejmenší chybou. Nutné je však brát v potaz tzv. přetrénování (angl. overfitting). Tento stav se projevuje tak, že klasifikátor dokáže velmi dobře klasifikovat trénovací data, avšak selhává při klasifikaci dat nových, neznámých. Jeho parametry jsou totiž bezchybně uzpůsobeny přímo trénovacím datům. Klasifikátor tak ztrácí schopnost zobecňovat. Proto je důležitým krokem při trénování zpětná validace. Tedy abychom snížili riziko přetrénování, je datový set rozdělen na trénovací a testovací. Nejprve se provede trénování na trénovacím

datasetu a posléze ověření kvality klasifikace na setu testovacím.

Výsledné skóre na testovacím setu může být ovlivněno způsobem rozdělení původních dat. Máme-li totiž vstupní data částečně na sebe navazující a vzájemně velice podobná, jen pouhým rozdělením lze získat velice dobré výsledky testovací sady. Tyto výsledky jsou však zkreslené díky velké podobnosti sousedních dat, nevypovídají tak příliš o kvalitách klasifikátoru a jeho schopnosti zobecňovat. U spojitého videa se právě tento problém vyskytuje. Po rozdělení videa si totiž budou data na hranici velice podobná. Řešením je použít buďto jiný záznam pro trénování, jiný pro testování, nebo tyto hraniční úseky vynechat. Pro zjištění, jakým způsobem se volba trénovacích dat nebo nastavení parametrů podílí na kvalitě klasifikace, se používá metoda cross-validation.

4.2 Cross-validation

Metoda křížové validace se ve statistice využívá k ověření schopnosti konkrétního modelu zobecňovat nová neznámá predikovaná data. Pomocí validace lze vyjádřit, jak moc přesně bude natrénovaný model fungovat v praxi. Principem metody je rozdělit dataset na trénovací a testovací část. Cílem je toto rozdělení provést několikrát na náhodně zamíchaných datech, získat výsledky pro danou testovací sadu a porovnat s předchozími iteracemi.



Obrázek 4.1: Ilustrace rozdělení dat. Jednotlivé barvy znázorňují různé zdroje dat. Při každé iteraci jsou data zamíchána a posléze je vybrán jiný úsek dat jako trénovací a testovací.

Tímto způsobem můžeme zhodnotit a porovnávat modely s jinými parametry, a tak učinit základní předpoklad, jak úspěšně se tento model bude chovat v praxi. Zároveň provedením validace získáme jistotu, že naše jedno rozdělení datové sady bude na úrovni jednotlivých vzorků nezávislé. Aplikace této metody by měla proběhnout vždy dříve, než se rozhodneme pro aplikaci nebo změnu modelu, případně pro sběr dalších dat.

4.3 Normalizace příznaků

Funkce některých výpočetních metod pracující s vícedimenzionálními vektory může být ovlivněna nerovnoměrnou velikostí jednotlivých prvků vektoru. Rozdíly několika řádů tak mohou výrazně ovlivnit proces výpočtu. Příkladem může být výpočet eukleidovské vzdálenosti, kde je výsledkem suma podobností jednotlivých složek. Zde totiž velikost podobnosti dvou složek v řádu stovek zcela přebije podobnost ostatních složek v řádu desetin, tedy výsledná suma bude ovlivněna pouze složkami v řádu stovek. Jiným příkladem je minimalizace kriteriální funkce výpočtem gradientů, přičemž je ovlivněna rychlost konvergence k minimální hodnotě funkce.

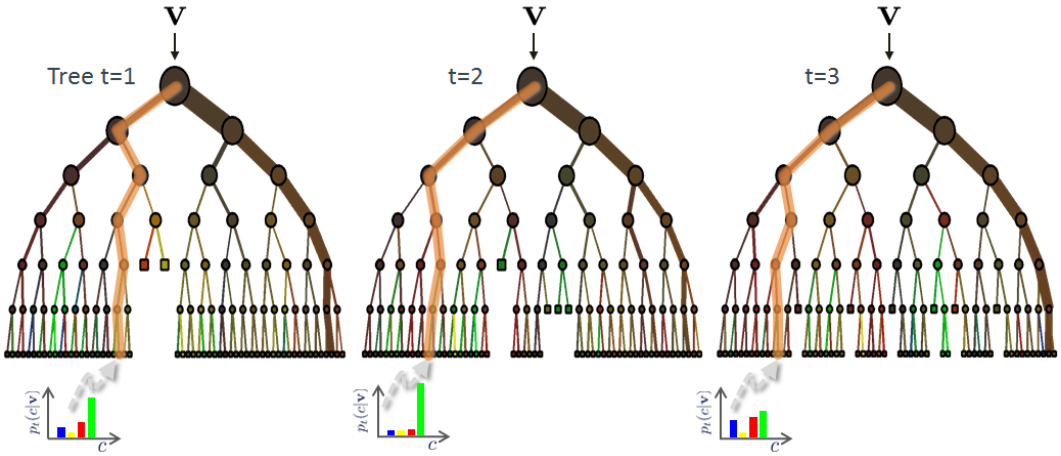
$$x_i{}' = \frac{x_i - \mu_x}{max(x) - min(x)} \tag{4.1}$$

Volba způsobu normalizace se odvíjí od charakteru dat. Požadavkem může být nulová střední hodnota jak ukazuje vztah 4.1, posun řádu jednotek, jednotková velikost vektoru, aj. Výsledkem normalizace je pak často transformace hodnot na intervaly $[-1; 1]$ nebo $[0, 1]$.

4.4 Klasifikátor Random Forest

Random Forest je metoda využívaná pro klasifikaci i regresi konstruující soubor náhodných rozhodovacích stromů. Jednotlivé náhodné rozhodovací stromy hrají roli dílčích klasifikátorů, jež následně průměrovací metodou rozhodují o konečném výsledku. Kombinací množství těchto slabých klasifikátorů lze získat silný klasifikátor schopný docílit konzistentního výsledku i u rozsáhlých dat. Příkladem úspěšného použití Random Forest je velice rychlá klasifikace částí těla a gest v zařízení Microsoft Kinect sloužící jako bezdrátové ovládání herní konzole snímáním pohybu celého těla.

Typické konstruování rozhodovacích stromů obvykle probíhá hledáním nejvhodnějšího dělícího kritéria zajišťující vždy rozdělení o maximálním informačním zisku. Každý uzel stromu tvoří jednoduché kritérium (otázku), na základě níž jsou data rozdělena. Vzniklé listy jsou posléze tvořeny množstvím dat z různých tříd, přičemž pravděpodobnost, že příznak x náleží do třídy C_i lze vyjádřit $P(C_i|x)$ pro každý list. V ideálním případě, kdy je list tvořen pouze daty z jedné třídy C_i, je tedy tato pravděpodobnost rovna jedné. Přestože je konstrukce rozhodovacího stromu relativně rychlá, proces hledání dělící hranice a výpočtu informačního zisku je u velkého množství dat, navíc vysoké dimenze, velice náročný. Rovněž má rozhodovací strom tendenci k přetrénování, protože hledá vždy to nejoptimálnější rozdělení daných dat. I když lze hledání mnohdy částečně zrobustnit omezením maximální hloubky stromu nebo minimálním počtem dat v jednom listu.



Obrázek 4.2: Ukázka různých výsledků konstrukcí tří náhodných stromů. Konečný výsledek získáme konkrétní metodou průměrování. Zdroj: CRIMINISI, Antonio, et al; Decision Forests for computer vision and medical image analysis, 2011.

Naproti tomu při konstrukci náhodných rozhodovacích stromů je hledání dělícího kritéria prováděno jen na náhodně vybrané podmnožině z vektoru příznaků, čímž dochází k redukci dimenze. Hledání dělícího kritéria pak probíhá opět na základě maximalizace informačního zisku nebo podobného kritéria. Vzhledem k náhodnému výběru a omezenému počtu příznaků samotný rozhodovací strom nepodává nejlepší možný výsledek. Bylo však dokázáno, že konstrukcí množství těchto náhodných stromů vždy na náhodné podmnožině vektoru trénovacích dat a jejich výsledným zprůměrováním (angl. metoda bagging) lze získat velice silný a spolehlivý klasifikátor. Přestože konstrukce Random Forest přináší množství výpočtu navíc, díky svému charakteru mohou být výpočty distribuovány na více strojů (cluster). Jeden stroj totiž pracuje pouze omezeným množstvím dat, čímž jsou výrazně sníženy paměťové nároky u obrovských datasetů odvíjející se od velikosti podmnožiny.

4.5 Klasifikátor Support Vector Machine

Support Vector Machine (SVM) je metoda používaná pro klasifikaci i regresi. Úkolem metody je najít nadrovinu, která vzájemně odděluje v prostoru příznaků jednotlivé třídy. Tato nadrovina optimálně rozděluje jednotlivé třídy tak, že minimální vzdálenost bodů z obou tříd od této nadroviny je co nejvyšší. Na obě strany tak vzniká co nejširší pruh (angl. margin). Z tohoto důvodu je metoda často označována angl. jako Large margin classifier. Typicky se jedná o binární dělení do dvou tříd. V případě rozdělení do více tříd se používá postup jeden-vs-všichni nebo jeden-vs-jeden v závislosti na implementaci algoritmu.

Pro lineárně nedělitelná data není však lineární rozdělení vhodné. Provede se tedy transformace příznaků do typicky mnohem vyšší dimenze pomoci specifické funkce - jádra (angl. kernel) a podpůrných vektorů (angl. support vectors). Až zde je následně provedena lineární separace jak ilustruje obrázek 4.3. Výsledné rozdělení může být výrazně ovlivněno tzv. outlier body, proto metoda zavádí penalizační parametr. Ten udává, jak moc má výsledné rozdělení zohledňovat tyto body.

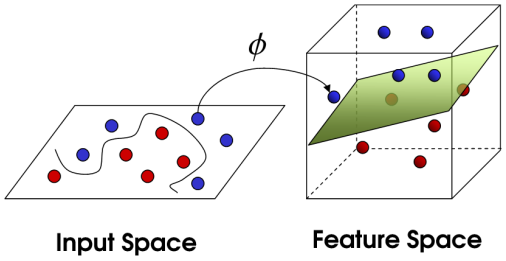
$$f_i = e^{-\frac{\|x-l^{(i)}\|^2}{2\sigma^2}} \qquad \approx \begin{cases} 1 & \text{pokud } x \approx l^{(i)} \\ 0 & \text{pokud } x \text{ je vzdáleno } l^{(i)} \end{cases} \tag{4.2}$$

Typicky používanými jádry jsou lineární (žádné jádro), Gaussovská funkce ve tvaru podobném vztahu 4.2, nebo polynomiální funkce. Transformace příznaku x je provedena vyjádřením míry podobnosti vůči m podpůrným vektorům $l^{(i)}$. Z n-dimenzionálního příznaku x tak vznikne m-dimenzionální vektor $f = [f_1\, f_2 \dots f_m]^T$. Po transformaci příznaku x na vektor f a natrénování parametrů θ minimalizací kriteriální funkce při učení s učitelem je pak rozdělení provedeno podle lineárního vztahu 4.3, kde h_θ zastupuje hypotézu klasifikace.

$$\theta_0 + \theta_1 f_1 + \dots + \theta_m f_m \geq 0 \qquad h_\theta(x) = \begin{cases} 1 & \text{pokud vztah platí} \\ 0 & \text{jinak} \end{cases} \tag{4.3}$$

Volba podpůrných vektorů se odvíjí od konkrétní implementace algoritmu. Mohou být zvoleny jen některé, nebo všechny příznaky z trénovací množiny. Výsledná rozdělující hranice tříd je ovlivněna jak velikostí penalizačního parametru, tak ještě v případě Gaussovského jádra variancí σ^2 ovlivňující ostrost rozhodující hranice. Nesprávnou volbou těchto parametrů docílíme buďto podtrénování nebo přetrénování klasifikátoru.

Výhodou použití jádra je schopnost rozdělit data silně nelineární funkcí i pro vysoké dimenze. Nevýhodou je však velmi vysoká výpočetní náročnost, která s počtem trénovacích vzorů dramaticky roste. Před použitím jádra je navíc vhodné provést normalizaci příznaků z důvodu výpočtu míry podobnosti.



Obrázek 4.3: Grafické znázornění transformace do vyšší dimenze, kde následně probíhá lineární separace, která má v původní dimenzi nelineární rozdělovací charakter. Zdroj: A. Wibisono, L. Rosasco: Hierarchical Learning Machines: Derived Kernels and the Neural Response

5 Návrh systému pro klasifikaci

Úlohou práce je navrhnout mechanismus, který dokáže klasifikovat jednotlivé snímky videa do předem určených kategorií dle vizuálního obsahu.

5.1 Návrh a vývoj algoritmů

Z důvodu obrovského množství (stovky GB) byla experimentální data uložena na diskovém poli Metacentra. Díky snadné možnosti napojení vzdáleného pole do lokálního souborového systému tak mohl vývoj bez komplikací pokračovat i na lokálním stroji. Přesto byly veškeré skripty vytvářeny s ohledem na možné spouštění na vzdáleném clusteru, a tedy na možnost zcela nezávislého vícenásobného simultánního spuštění. Jednotlivé části systému jsou tvořeny samostatnými skripty napsanými v jazyce Python. Použité open-source knihovny a jiné nástroje pak převážně v jazycích Python a C/C++. Na konec jsou jednotlivé části spojeny do samostatné aplikace.

Vstupní data ve formě seznamu videí s vybranými intervaly v sekundách jsou uložena v textové souboru ve formátu JSON. To umožnuje jednak velice snadnou editaci a jednak velice snadné načtení a parsování v jazyce Python. Výstupní data jsou uložena v komprimovaném binární formátu.

5.1.1 Extraktory příznaků

Pro extrakci příznaků jsem vytvořil nadřízený skript, který na základě požadavků volá jednotlivé extraktory uložené opět ve formě samostatných skriptů. Vstupem je soubor se seznamem videí a výčet extrahovaných příznaků uvedený

jako konzolové argumenty skriptu. Protože se jedná o vzájemně nezávislé výpočty, nabízí se tak možnost paralelizace. Volitelně lze skript spustit s parametrem udávajícím prioritu a počet současně spuštěných úloh. Pro extrakci snímků z videa a předzpracování používám knihovny OpenCV [7] a SciPy [8]. Výsledky jsou exportovány na disk v komprimované binární formě knihovnou Numpy [8], která později umožňuje jejich rychlé načtení zpět do paměti.

Detektor střihů

Použil jsem skript vytvořený v [2] pro vyjádření velikosti změny v obraze, a tedy čísel snímků, kde se předpokládá výskyt střihu. Získaná data slouží jako vstup extraktorů četnosti střihů a pohybu.

Četnost střihů

Vytvořený skript počítá počet výskytu střihů za posledních n snímků. Výsledná hodnota je normalizována podílem n snímků. Svým způsobem tedy vyjadřuje, s jakou pravděpodobností se v daném snímků vyskytne střih.

Četnost pohybu

Vytvořený skript aritmeticky průměruje velikost změny v obraze za posledních m snímků. Není zde prováděna žádná další normalizace, protože se hodnoty pohybují maximálně v řádu jednotek.

Barevné rozložení

Vytvořený skript převede RGB snímek do HSV prostoru a vypočte histogram pro každou ze složek. Následně je histogram transformován do b binů a vektor normalizován na jednotkovou hodnotu. Oproti původnímu záměru použít pouze dominantní barvu a jasovou hodnotu extrahuji všechny tři složky HSV. Výsledný vektor tak odpovídá pravděpodobnostnímu rozložení jednotlivých intenzit barevných složek.

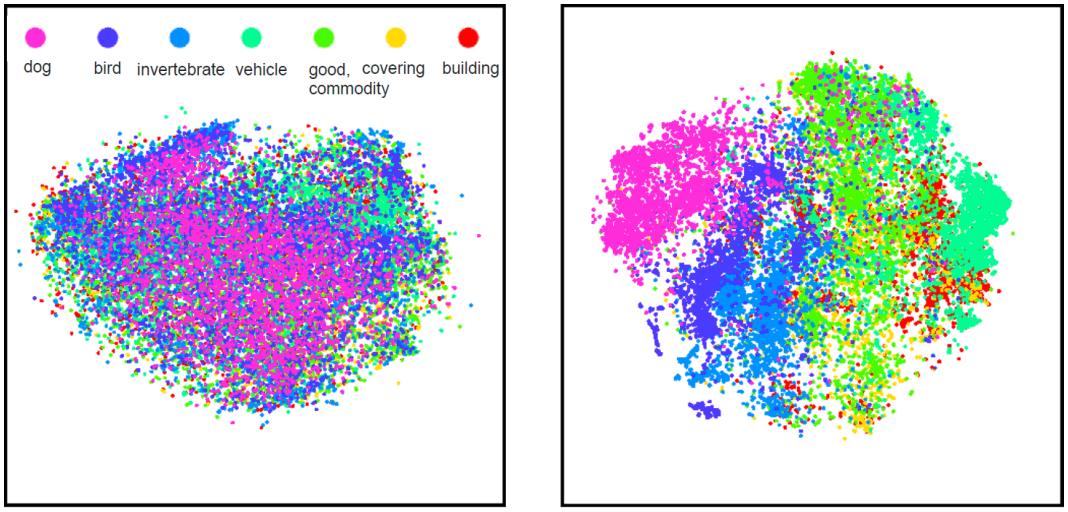
Detekce obličejů

Použil jsem natrénovaný obličejový detektor [3] z knihovny OpenCV. Celková suma obličejů x za k snímků je dále normalizována funkcí $x' = min(1.0,\ (x/k)/10.0)$. Funkce nabývá jednotkové hodnoty, pouze vyskytne-li se průměrně za posledních k snímků v každém snímku deset nebo více obličejů. V opačném případě hodnota vyjadřuje průměrný počet obličejů za k snímků normalizovaný z rozsahu $[0, 10]$ na interval $[0, 1]$. Nastavitelným parametrem skriptu je zmenšovací krok plovoucího okénka detektoru, jenž má přímý vliv na kvalitu detekce.

Detekce objektů konvoluční neuronovou sítí

Použil jsem dostupný framework Caffe [9], který je modifikací cuda-convnet [10] modelu konvolučních sítí. Caffe umožňuje využít pro výpočty jak procesor, tak CUDA rozhraní grafických karet NVIDIA, které dokáží dobu průchodu i několikrát zkrátit. Využil jsem již hotový model natrénovaný na derivát obrazové databáze ILSVRC-2012 schopný klasifikovat objekty do jednoho tisíce tříd. Na vstup sítě přivádím celý snímek extrahovaný z videa pouze s odstraněným prokládáním. Výstupem softmax výstupní vrstvy je vektor dimenze tisíc, kde jednotlivé složky vektoru jsou z intervalu $[0; 1]$ a celková suma vektoru je rovna jedné. Výstup lze tedy interpretovat jako pravděpodobnosti jednotlivých tříd (objektů).

Jako výstupní informaci používám experimentálně i aktivační hodnoty skrytých vrstev těsně před výstupní vrstvou (vrstvy FC_8, FC_7 [9]). Jedná se totiž už o samotné příznaky, přičemž poslední plně propojené vrstvy slouží jako klasifikátor. Úkolem celé sítě je totiž extrakce a posléze transformace příznaků, viz obrázek 5.1.



Obrázek 5.1: Vizualizace prováděné nelineární transformace extrahovaných příznaků z prostoru neumožnující separaci jednotlivých tříd (nižší vrstvy neuronové sítě) do prostoru, kde je toto rozdělení možné (vyšší vrstvy neuronové sítě). Zdroj: JIA, Y.; Caffe presentation, 2014

5.1.2 Rozdělení dat

Vytvořený skript rozdělí data náhodně v určitém poměru na trénovací a testovací sadu. Dělící poměr je variabilní. Typicky používám 70% dat trénovacích a 30% dat testovacích. Díky vhodné implementaci lze data velice snadno přerozdělit změnou parametru "random seed", který nastavuje generátor náhodných čísel. Získané různě namíchané datasety lze dále použít pro metodu cross-validation, a tedy vybalancovat optimální nastavení parametrů systému.

5.1.3 Trénování

Vytvořený skript používá implementované klasifikátory z knihovny scikit-learn [11]. Použití již hotových knihoven přináší několik výhod. Krom úspory času při implementaci jsou tyto algoritmy často numericky upraveny na míru a nezřídkakdy používají vysoce optimalizované nízkoúrovňové matematické knihovny. V případě knihovny scikit-learn lze pro každý klasifikátor volitelně upravovat parametry ovlivňující jeho funkci, tudíž nic nebrání jejich použití. Stěžejní funkcí knihovny je navíc

možnost exportovat kompletní natrénovaný klasifikátor do souboru a jeho následná přenositelnost na jiné zařízení.

Proces trénování pro tento relativně malý, byť pro jeden stroj obrovský, dataset je velice hardwarově náročný. Pozitivní vlastností je nezávislost dat, tudíž mohou být výpočty spuštěny simultánně na více strojích ve výpočetním centru. Pro vyšší dimenze navíc drasticky roste paměťová náročnost (desítky GB), tudíž většina výpočtů nemohla být ani realizována na osobním počítači. Použil jsem proto možností Metacentra, jež nabízí potřebné hardwarové kapacity. Výsledné časy se tak dramaticky snížily. Samotný proces trénování probíhal metodou učení s učitelem.

5.1.4 Testování a vyhodnocení

Vytvořený skript dokáže načíst z disku do paměti kompletní natrénovaný klasifikátor, predikovat testovací data a na základě informací od učitele vyhodnotit výsledky klasifikace. Pro základní statistické vyhodnocení klasifikace hledáme číslo, které nejlépe vyjádří úspěšnost naší predikce. Samotná informace o velikosti chyby mnohdy nemusí stačit. Uvažme totiž příklad binární klasifikace, kdy máme polovinu pacientů zdravou a polovinu pacientů nemocnou. Vytvoříme systém, který bude vždy predikovat, že pacient je nemocný. Nabízí se možnost prohlásit, že systém pracuje s nulovou chybou vzhledem k detekci nemocných pacientů. Jenomže je třeba brát v úvahu úspěšnost detekce i vzhledem k druhé, tedy zdravé polovině pacientů. Pro predikovaná data proto rozlišujeme čtyři způsoby zaklasifikování, jak ukazuje tabulka 5.1. Úspěšnost klasifikace se pak častěji vyjadřuje hodnotami tzv. precision a recall.

původní třída	1	1	0	0
predikovaná třída	1	0	0	1
	True Positive TP	False Negative FN	True Negative TN	False Positive FP

Tabulka 5.1

Hodnota precision P (přesnost) nám pro uvedený příklad binární klasifikace udává, jak velká část klasifikovaných pozitivních pacientů má opravdu nemoc. Hodnota recall R (úplnost) posléze vyjadřuje, jak velkou část ze všech skutečně nemocných pacientů jsme identifikovali správně. Aplikací těchto metrik na výše zmíněný primitivní systém už získáme hodnoty, které jsou velmi vzdálené od bezchybného chování. Protože pro konečné posouzení kvality systému je nutné uvažovat výsledky dvou čísel, častěji se používá vyjádření metriky zvané F_1 score, která mnohem věrohodněji kombinuje výsledky precision a recall než jejich pouhý aritmetický průměr. Výstupem je tedy pouze jedna konkrétní hodnota, pomocí níž dokáže i laik velice snadno a rychle porovnat úspěšnost více systémů.

$$P = \frac{TP}{TP + FP} \qquad R = \frac{TP}{TP + FN} \qquad F_1 = 2\frac{PR}{P + R} \tag{5.1}$$

Z predikovaného výstupu systému lze nejen vypočítat jeho úspěšnost, ale i pro získání lepší intuice vypočítat matici záměn (angl. confusion matrix). Ta vyjadřuje, jaké třídy jsou mezi sebou nejčastěji zaměňovány. Výhodou je i možná grafická reprezentace. Získaná data pak mohou posloužit jako vodítko buďto pro odhalení chyby systému, nebo pro jeho další zdokonalení.

5.1.5 CLI aplikace

Na konec vytvořená CLI (command-line interface) aplikace zkombinuje používané moduly dohromady. Na základě vstupní informace od uživatele provede automaticky extrakci požadovaných příznaků a predikuje kategorii videa. Z výsledku může být volitelně sestaven histogram predikovaných kategorií, či lze volitelně vybrat velikost postprocessingu. Ten vyhladí výstupní informaci přes plovoucí časové okénko určité šířky (medián). Výsledky jsou uloženy v textové podobě pro potencionální budoucí využití.

5.2 Experimentální data

Sestaveny byly celkem dvě úrovně – obecné kategorie a sportovní kategorie. Pro účely experimentu byl ke každé kategorii sesbírán určitý časový úsek z rozmanitých zdrojů v rámci dostupných možností.

Všechna videa byla použita v kvalitě dosahující nejméně DVD formátu. Z důvodu výskytu prokládání (angl. interlacing) musela být videa dále předzpracována odstraněním lichých řádků, čímž došlo k redukci jejich velikosti.

5.2.1 Obecné kategorie

- animovaný film *(animated)*
- diskusní pořad *(talkshow)*
- hraný film *(movie)*
- kreslený film *(cartoon)*
- poslanecká sněmovna *(politics)*
- sport *(sport)*
- zprávy *(news)*

Obecnou úroveň tvoří výše vyjmenované kategorie, kde v závorce je uveden unikátní identifikátor. Pro každou kategorii byl vybrán cca dvouhodinový záznam sestavený převážně z deseti videí. Z každého videa je tedy použit přibližně dvanáctiminutový úsek vyjma sportovní kategorie, kde je videoset daleko rozmanitější.

Při výběru videí bylo cílem alespoň částečně pokrýt rozmanitost dané kategorie. Tedy u sportovní kategorie byl vybrán několikaminutový interval z většiny dostupných sportovních záznamů z letní a zimní olympiády. Při sestavování hraných filmů byl vybrán prostřih filmovými žánry typickými svou dynamikou (film akční, romantický, muzikál, atp.), u kreslených filmů sestřih napříč světovou tvorbou (západní tvorba, japonské anime, atp.) a analogicky u animovaných filmů (3D díla studia Pixar, Dreamworks, Disney, atp.).

5.2.2 Sportovní kategorie

- akrobatické lyžování *(freestyle_skiing)*
- badminton *(badminton)*
- basketbal *(basketball)*
- beach volejbal *(beach_volleyball)*
- běžecké lyžování *(cross-country_skiing)*
- biatlon *(biathlon)*
- box *(box)*
- curling *(curling)*
- cyklistika *(cycling)*
- floorbal *(floorball)*
- fotbal *(football)*
- golf *(golf)*
- gymnastika *(gymnastics)*
- házena *(handbal)*
- hokej *(ice_hockey)*
- jachting *(yachting)*
- jezdectví *(equestrianism)*
- judo *(judo)*
- krasobruslení *(figure_skating)*
- lehká atletika *(athletics)*
- lukostřelba *(archery)*
- plavání *(swimming)*
- pozemní hokej *(field_hockey)*
- rychlobruslení *(speed_skating)*
- saně *(bobsleigh)*
- šerm *(fencing)*
- sjezdové lyžování *(downhill_skiing)*
- skoky na lyžích *(ski_jumping)*
- snowboarding *(snowboarding)*
- stolní tenis *(table_tennis)*
- střelba *(shooting_sports)*
- taekwondo *(taekwondo)*
- tenis *(tennis)*
- veslování *(rowing)*
- vodní polo *(water_polo)*
- vodní slalom *(whitewater_slalom)*
- volejbal *(volleyball)*
- vzpírání *(weightlifting)*
- zápas *(wrestling)*

Sportovní kategorie tvoří výše vyjmenovaný seznam, kde v závorce je vždy uveden unikátní identifikátor. Pro každou kategorii byl vybrán cca hodinový záznam sestaven z různého počtu dostupných videí. Zdrojem sportovních videí byly uplynulá letní olympiáda v Londýně 2012 a zimní olympiáda v Sochi 2014.

6 Výsledky experimentů

Tabulka 6.1 zobrazuje veškeré testované kombinace příznaků. Všechny uvedené kombinace příznaků byly natrénovány a otestovány pouze klasifikátorem Random Forest. Ukázalo se totiž, že trénování klasifikátoru SVM je časově i paměťově mnohem náročnější. Aby bylo vůbec možné natrénovat takové množství kombinací, rozhodl jsem se pouze pro jeden klasifikátor. Přičemž parametry klasifikátoru byly ponechány pro všechny kombinace stejné, tedy výchozí. Pro konvoluční sítě je dále natrénována dvojice obou klasifikátorů s upravenými parametry.

Uvedeny jsou výsledky dvojího typu – nevyhlazené a vyhlazené přes okénko délky M mediánovým filtrem. Nevyhlazené výsledky reprezentují surový predikovaný výstup klasifikátoru, který predikuje jednotlivé snímky videosekvence nezávisle. Protože klasifikujeme snímky ze spojitých videozáznamů, zcela se zde nabízí možnost postprocessingu. Motivací použití postprocessingu je jednak apriorní znalost, že snímky po sobě jdoucí tvoří bloky odpovídající konkrétní kategorii, a jednak samotný charakter záznamu. Uvažme totiž například záznam některé ze sportovních kategorií. Záběr kamery se může opakovaně stočit do publika na fanoušky. V takovéto situaci však klasifikátor nemá nejmenší šanci rozlišit, o jaký sport se jedná, pomineme-li oblečení lidí. O mnoho lépe na tom není sám člověk. Jestliže ukážeme jedinci pouze jeden snímek, může na základě informace z obrázku sice prohlásit, že se jedná o zimní sport (sníh, lidé v čepicích, atp.), avšak pravděpodobně nebude schopen správně určit, o jaký sport se konkrétně jedná, aniž by měl nějakou apriorní znalost. Proto zde uplatníme možnost vyhlazení. Tedy jestliže se vlevo a vpravo nachází dlouhý blok sjezdu na lyžích, je nanejvýš pravděpodobné, že daný snímek rovněž patří do kategorie sjezdu na lyžích.

Nejlepší dosažené výsledky s klasifikátorem Random Forest

	obecné kategorie			sportovní kategorie		
	mean 0	mean M		mean 0	mean M	
kombinace příznaků	F_1	F_1	M	F_1	F_1	M
HSV	61%	63%	1001	66%	77%	2501
HSV+CR	62%	66%	1001	67%	78%	2501
HSV+CR+FC	65%	69%	2501	66%	78%	2501
HSV+CR+FC+MR	66%	71%	7501	68%	77%	2501
CNN	66%	**86%**	12501	63%	74%	1251
CNN+HSV	**70%**	78%	7501	**70%**	**80%**	2501
CNN+HSV+CR	68%	76%	10001	69%	78%	2501
CNN+HSV+CR+MR	70%	78%	5001	69%	79%	2501
CNN+HSV+CR+FC+MR	70%	77%	15001	69%	79%	1001
CNN+CR	68%	79%	12501	63%	74%	1001
CNN+CR+FC	68%	79%	7501	63%	74%	1001
CNN+CR+FC+MR	70%	80%	7501	63%	74%	1001
CNN+CR+MR	69%	80%	12501	63%	74%	1001
CNN_FC8	68%	**81%**	10001	65%	77%	2501
CNN_FC7	**71%**	80%	10001	-	-	-

Tabulka 6.1: Hodnoty zastoupené pomlčkou značí, že uvedenou kombinaci nebylo možné z důvodu obrovských paměťových nároků možné natrénovat. Vysvětlivky použitých zkratek: HSV = barevné rozložení; CR = četnost střihů; FC = četnost obličejů; MR = četnost změny obrazu; CNN = konvoluční neuronové sítě; FC8 = poslední skrytá vrstva sítě; FC7 = předposlední skrytá vrstva sítě; mean = aplikace postprocessingu.

Z výsledků v tabulce 6.1 vidíme, že i velice primitivní příznaky, jako je rozložení barev, podávají dobré výsledky. Zde je ale třeba vzít v úvahu velmi omezený dataset. Zejména u sportovních kategorií je třeba si uvědomit, že byla použita data pouze z olympijských her. Prostory konání jednotlivých sportů na olympijských hrách se totiž vyznačují svou podobnou, až totožnou, ale zejména výraznou barevností (tyrkysová, fialová, žlutá, aj.). Spíše než o schopnosti klasifikátoru poznat sport zde tedy můžeme hovořit o příkladu přetrénování.

Pro primitivní příznaky výsledky experimentů ukázaly, že spojením více příznaků lze docílit zlepšení detekce o jednotky procent. Například už použití všech tří složek HSV během vývoje ukázalo zlepšení detekce až o 4%. Zajímavé je vyjádření přínosu detekce obličejů pro obecné kategorie v porovnání se sportovními. Přínos 3% u obecných kategorií lze vysvětlit zlepšením detekce tam, kde je a kde naopak není očekáván výskyt obličejů. Naopak sportovní kategorie obecně výskytem obličejů překypují, a pravděpodobně proto zde tento příznak nemá pozitivní dopad na výsledek. Nejlepší vyhlazené výsledky kombinace čtyř primitivních příznaků dosahují překvapivě vysokých hodnot 71% u obecné a 78% u sportovní kategorie. Vysvětlení takto vysokých hodnot v porovnání s komplexními konvolučními sítěmi tedy odhaduji spíše na přetrénování než na nabytou schopnost zobecnění.

Použití konvolučních sítí spolu s kombinací primitivních příznaků posouvá schopnost korektní detekce ve většině případů dále. Nejlepších nevyhlazených výsledků dosahuje kombinace konvolučních sítí spolu s barevným rozložením, konkrétně 70% u obecné i sportovní kategorie. Aplikací postprocessingu jsem získal nejvyšší hodnoty 86% pro samotnou síť u obecné kategorie a 80% pro kombinaci konvoluční sítě a barev u sportovní kategorie.

Všimněme si zajímavých výsledků příznaků extrahovaných ze skryté vrstvy konvoluční sítě, zejména vrstvy FC_7. Tyto příznaky experimentálně zařazené do testu jasně dokazují funkci klasifikátoru plně propojené vrstvy konvoluční sítě. Bohužel se z enormních paměťových nároků přes 100GB RAM nepodařilo získat výsledky pro sportovní kategorii (cca 4.5mil snímků dimenze 4096). Nabízející se možností je aplikace algoritmů redukce dimenze PCA, LDA, nebo použití jen každého n-tého snímku.

Pro podrobnější analýzu jsem vybral příznaky z výstupní vrstvy konvoluční sítě. Tabulka 6.2 ukazuje výsledky třech testovacích iterací cross-validation vždy na jiném složení trénovací a testovací sady. Z výsledků je patrné, že rozdíly jsou v řádu jednotek procent, tudíž získané výsledky nejsou závislé na složení datasetů.

Cross-validation výsledky dosažené s klasifikátorem Random Forest

	obecné kategorie			sportovní kategorie		
	mean 0	mean M		mean 0	mean M	
kombinace příznaků	F_1	F_1	M	F_1	F_1	M
CNN (1)	66%	85%	7501	63%	74%	5001
CNN (2)	66%	80%	7501	65%	78%	5001
CNN (3)	65%	78%	7501	64%	80%	5001

Tabulka 6.2: Tabulka ukazuje výsledky cross-validation pro tři různě rozdělené datasety. Vysvětlivky použitých zkratek: CNN = konvoluční neuronové sítě; (1) = číslo iterace

Původním cílem bylo porovnat výsledky dosažené klasifikátory Random Forest a SVM. Proces trénování a testování klasifikátoru SVM je z důvodů velkého množství dat velmi náročný i přesto, že byl volen každý n-tý snímek. Natrénování a otestování SVM klasifikátoru se podařilo pouze pro obecné kategorie, pro kategorie sportu nikoliv. Srovnání výsledků klasifikátorů Random Forest a SVM s výchozími parametry ukazuje tabulka 6.3.

Protože způsob konstrukce a výsledná funkce klasifikátoru Random Forest je ovlivněna mnoha parametry, experimentálně jsem otestoval několik variant nastavení. Jedním ze základních parametrů mající vliv na funkci Random Forest je počet vytvářených náhodných stromů, ze kterých je posléze průměrován výsledek, jak bylo popsáno v kapitole klasifikace obrazu. Tabulka 6.3 ukazuje vliv počtu konstruovaných dílčích stromů na podané výsledky klasifikace. Pro větší počet konstruovaných dílčích stromů roste výpočetní a paměťová náročnost, tudíž byl experiment proveden z výchozích deseti maximálně pro stovku dílčích stromů. Výsledky ukazují, že pro vyšší počty stromů neroste dosažená přesnost klasifikace tak zásadně. Dá se tedy předpokládat, že pro rostoucí počet konstruovaných dílčích stromů bude mít dosažená přesnost ustalující charakter.

Výsledky konvoluční neuronové sítě pro varianty klasifikátoru Random Forest

	obecné kategorie			sportovní kategorie		
	mean 0	mean M		mean 0	mean M	
Random Forest	F_1	F_1	M	F_1	F_1	M
10 estimátorů	66%	86%	12501	63%	75%	2501
20 estimátorů	69%	79%	12501	67%	79%	1001
50 estimátorů	73%	83%	12501	71%	81%	1001
70 estimátorů	74%	84%	12501	**72%**	**81%**	1001
100 estimátorů	74%	84%	12501	-	-	-
SVM	F_1	F_1	M	F_1	F_1	M
lineární kernel	**77%**	**93**%	5001	-	-	-

Tabulka 6.3: Tabulka ukazuje výsledky srovnání klasifikátorů Random Forest s různým počtem dílčích stromů konstruovaných během procesu trénování. Výsledky zastoupené pomlčkou se nepodařilo z důvodu vysokých výpočetních nároků natrénovat.

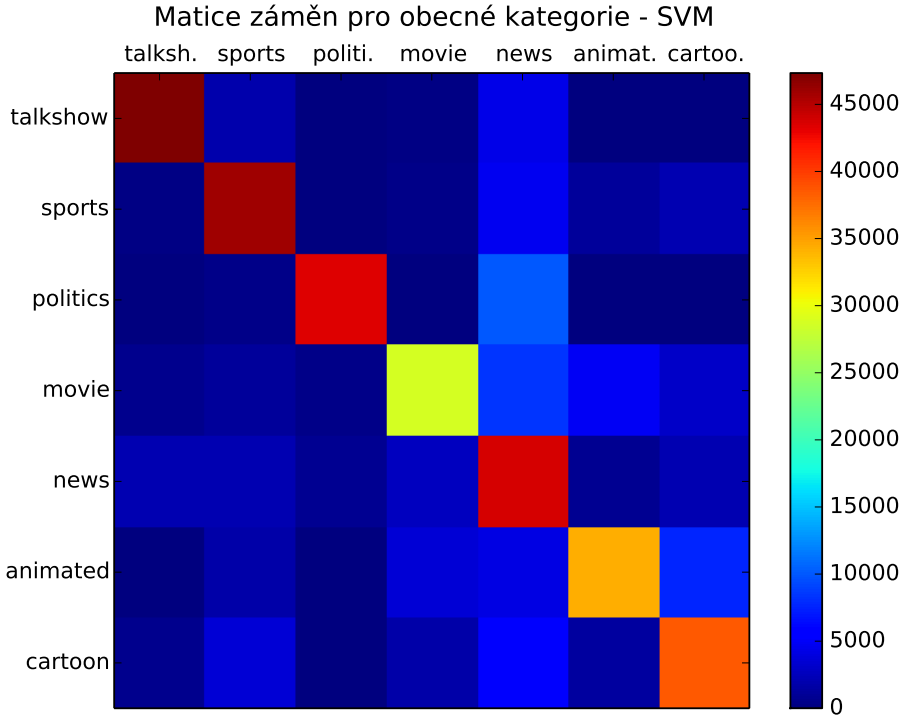
Tabulky 6.4 a 6.5 podrobně ukazují nejlepší dosažené výsledky uvedené v tabulce 6.3 včetně úspěšnosti klasifikace dílčích kategorií. Pro získání lepší intuice grafy 6.1 a 6.2 ilustrují matice záměn, tedy podávají informace, jaké kategorie jsou v průběhu klasifikace mezi sebou zaměňovány. V ideálním případě bezchybné detekce bude matice záměn pouze diagonální, tedy snímky z konkrétní kategorie jsou klasifikovány pouze do této kategorie.

Pro obecné kategorie lze pozorovat, že nejobtížněji se detekují diskusní pořady spolu s animovanými a kreslenými filmy. Matice záměn na obrázku 6.1 ukazuje, že nejčastěji se kategorie zaměňují se zprávami. Pokud si však představíme obsah zpravodajské relace, dojdeme k závěru, že například klasifikace většiny reportáží představuje opravdu problém. To proto, že klasifikátor klasifikuje jednotlivé snímky zcela nezávisle, tedy nemá informaci, že předchozí tisíce snímků klasifikoval jako zpravodajství, a tedy současný snímek může být právě reportáž. Kvůli možné délce reportáže zde nemusí pomoci ani postprocessing.

Podrobné výsledky pro obecné kategorie s klasifikátorem SVM pro lineární kernel.

Video kategorie	precision	recall	F_1
talkshow	77%	61%	68%
sports	80%	84%	82%
politics	93%	87%	90%
movie	96%	80%	88%
news	54%	81%	65%
animated	80%	66%	73%
cartoon	72%	75%	73%
průměr	79%	77%	77%

Tabulka 6.4

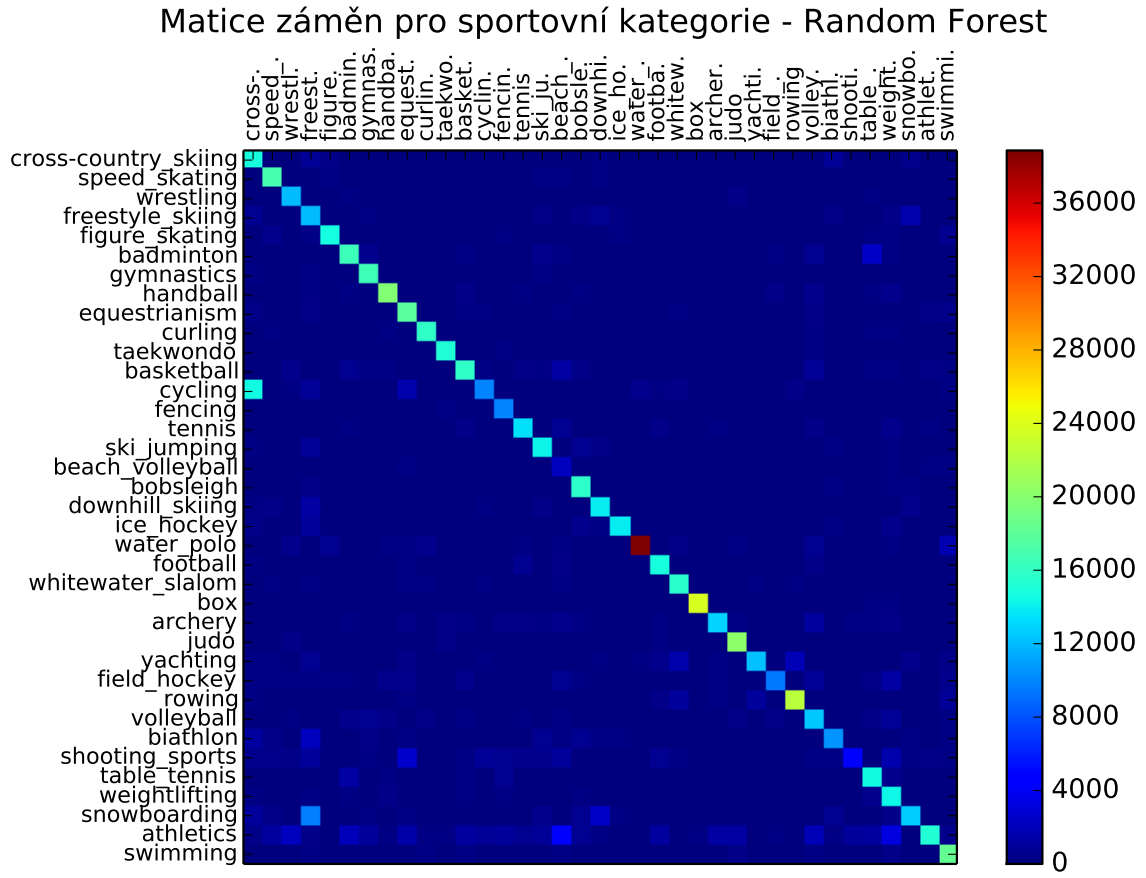


Obrázek 6.1: Grafická reprezentace matice záměn z výsledků uvedených v tabulce 6.4. Lze pozorovat, že nejčastěji zaměňovanými kategoriemi jsou zprávy, filmy hrané, kreslené i animované. Například můžeme vidět, že kategorie politiky je nejčastěji zaměňována se zprávami, stejně tak kategorie filmů.

Podrobné výsledky pro sportovní kategorie s klasifikátorem Random Forest pro 70 estimátorů.

Video kategorie	precision	recall	F_1
cross-country_skiing	40%	82%	54%
speed_skating	74%	90%	81%
wrestling	69%	88%	78%
freestyle_skiing	35%	67%	46%
figure_skating	85%	81%	83%
badminton	71%	73%	72%
gymnastics	78%	92%	85%
handball	85%	82%	84%
equestrianism	65%	80%	72%
curling	84%	88%	86%
taekwondo	89%	91%	90%
basketball	76%	70%	73%
cycling	72%	33%	45%
fencing	78%	87%	82%
tennis	69%	93%	79%
ski_jumping	75%	75%	75%
beach_volleyball	71%	79%	75%
bobsleigh	16%	52%	24%
downhill_skiing	71%	88%	78%
ice_hockey	71%	78%	74%
water_polo	86%	78%	82%
football	97%	85%	91%
whitewater_slalom	77%	82%	79%
box	51%	79%	62%
archery	96%	94%	95%
judo	78%	67%	72%
yachting	84%	90%	87%
field_hockey	88%	58%	70%
rowing	81%	53%	64%
volleyball	88%	83%	86%
biathlon	57%	70%	63%
shooting_sports	74%	59%	65%
table_tennis	67%	24%	35%
weightlifting	65%	80%	72%
snowboarding	73%	42%	54%
athletics	87%	33%	47%
swimming	71%	85%	78%
průměr	76%	72%	72%

Tabulka 6.5



Obrázek 6.2: Grafická reprezentace matice záměn z výsledků uvedených v tabulce 6.5. Lze pozorovat, že nejčastěji zaměňovanými kategoriemi jsou například cyklistika, snowboarding a atletika. Obrovskou chybu můžeme pozorovat právě u cyklistiky, která je nejvíce zaměňována s běžeckým lyžováním.

Z výsledků klasifikace sportovních kategorií vidíme výrazně větší výkyvy v úspěšnosti klasifikace jednotlivých kategorií. Zatímco některé kategorie jako fotbal a lukostřelba se daří klasifikovat velice přesně, klasifikace jízdy na saních nebo stolního tenisu naopak selhává. Zde by schopnost klasifikace mohly potencionálně zlepšit další příznaky popisující například směr pohybu přes více snímků.

7 Závěr

V této práci jsem navrhl postup pro klasifikaci jednotlivých snímků videa dle obsahu do určených kategorií za pomoci jednodušších i složitějších příznaků včetně jejich kombinací. Výsledky experimentů ukázaly tendenci jednodušších příznaků k přetrénování, zvláště barevného rozložení. Podrobněji jsem se zaměřil metodu extrakce příznaků využívající konvoluční neuronové sítě. Konečná podrobná analýza ukázala, jaké konkrétní kategorie je nejobtížnější detekovat. Nejlepšího výsledku na úrovni obecných kategorií jsem dosáhl 77% a po aplikaci postprocessingu 93%. Tento výsledek ukázal výrazný potenciál konvolučních sítí jako relativně nové metody klasifikace. Pro detekci jednotlivých sportů bylo z důvodu výpočetní a paměťové náročnosti dosaženo nejlepšího výsledku 72% a po aplikaci postprocessingu 81%.

Dalšího zlepšení dosažených výsledků, jak ukázal experiment, lze docílit například optimalizací parametrů klasifikátorů. Jejich natrénování a otestování je však velice náročné, proto by bylo vhodné použití metod snižující dimenzi příznaků (např. PCA). Tím bychom snížili náročnost výpočtů a umožnili natrénovat více kombinací parametrů klasifikátorů, mezi nimiž bychom mohli posléze porovnávat. Alternativou je vyhodnocení pouze každého n-tého snímku pro vyšší hodnoty n (každý pátý, desátý, apod.). Celkově však navržený systém podává uspokojivé výsledky a potencionální využití by nalezl například pro automatickou anotaci scény či indexaci videa.

Literatura

[1] SONKA, Milan, et al. *Image processing, analysis, and machine vision.* Toronto: Thomson, 2008.

[2] MATOUSEK, Jindrich; VIT, Jakub. *Improving automatic dubbing with subtitle timing optimisation using video cut detection.* In: Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on. IEEE, 2012. p. 2385-2388.

[3] VIOLA, Paul; JONES, Michael. *Rapid object detection using a boosted cascade of simple features.* In: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. IEEE, 2001. p. I-511-I-518 vol. 1.

[4] FUKUSHIMA, Kunihiko. *Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position.* Biological cybernetics, 1980, 36.4: 193-202.

[5] HUBEL, David H.; WIESEL, Torsten N. *Receptive fields, binocular interaction and functional architecture in the cat's visual cortex.* The Journal of physiology, 1962, 160.1: 106.

[6] FARABET, Clément, et al. *Learning hierarchical features for scene labeling.* Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2013, 35.8: 1915-1929.

[7] BRADSKI, Gary. *The opencv library.* Doctor Dobbs Journal, 2000, 25.11: 120-126.

[8] JONES, Eric; OLIPHANT, Travis; PETERSON, Pearu. *SciPy: Open source scientific tools for Python*. http://www. scipy. org/, 2001.

[9] JIA, Y. *Caffe: An open source convolutional architecture for fast feature embedding*. h ttp://caffe. berkeleyvision. org, 2013.

[10] KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. *ImageNet Classification with Deep Convolutional Neural Networks*. In: NIPS. 2012. p. 4.

[11] PEDREGOSA, Fabian, et al. *Scikit-learn: Machine learning in Python*. The Journal of Machine Learning Research, 2011, 12: 2825-2830.