VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

PŘÍZNAKY Z VIDEA PRO KLASIFIKACI

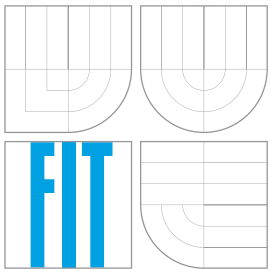
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. KAMIL BEHÚŇ

BRNO 2013

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

PŘÍZNAKY Z VIDEA PRO KLASIFIKACI

VIDEO FEATURE FOR CLASSIFICATION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. KAMIL BEHÚŇ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MICHAL HRADIŠ

BRNO 2013

Abstrakt

Tato práce porovnává ručně-navrženy příznaky s příznaky naučenými metodami učení příznaků při klasifikací videa. Příznaky naučené pomocí Analýzy nezávislých podprostorů, Řídkými Autoenkodéry a vybělením Analýzou hlavních komponent byly otestovány v systému pro klasifikaci videa pomocí Bag of Words, ve kterém nahradily ručně-navrženy příznaky (např. SIFT, HOG, HOF). Úspěšnost klasifikace těchto naučených příznaků byla testována na datových sadách Human Motion DataBase a YouTube Action Data Set, kde ukázaly lepší výsledky než ručně-navrženy příznaky. Tato práce také ukazuje pomocí navržené metody inspirovanej metódami Multiple Kernel Learning, že při kombinaci naučených příznaků s ručně-navrženými příznaky lze dosáhnout ještě výraznější zlepšení úspěšnosti klasifikace videa a to i v případě, když ručně-navrženy příznaky a naučené příznaky samostatně nedosahují příliš velké úspěšnosti klasifikace.

Abstract

This thesis compares hand-designed features with features learned by feature learning methods in video classification. The features learned by Principal Component Analysis whitening, Independent subspace analysis and Sparse Autoencoders were tested in a standard Bag of Visual Word classification paradigm replacing hand-designed features (e.g. SIFT, HOG, HOF). The classification performance was measured on Human Motion DataBase and YouTube Action Data Set. Learned features showed better performance than the hand-desined features. The combination of hand-designed features and learned features by Multiple Kernel Learning method showed even better performance, including cases when hand-designed features and learned features achieved not so good performance separately.

Klíčová slova

Klasifikace videa, video příznaky, učení příznaků, Analýza hlavních komponent, Nezávislá analýza podprostoru, Řídké Autoenkodéry, Bag of Words, Support Vector Machine, Multiple Kernel Learning

Keywords

Video Classification, video features, feature learning, Principal component analysis, Independent subspace analysis, Sparse Autoencoders, Bag of Words, Support Vector Machine, Multiple Kernel Learning

Citace

Kamil Behúň: Příznaky z videa pro klasifikaci, diplomová práce, Brno, FIT VUT v Brně, 2013

Příznaky z videa pro klasifikaci

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Michala Hradiše

.....
Kamil Behúň
22. května 2013

Poděkování

Děkuji vedoucímu mé práce Ing. Michalovi Hradišovi za odbornou pomoc při problémech, které vznikli při řešení této práce a za čas, který mi věnoval.

Obsah

Kapitola 1

Úvod

V posledných rokoch dochádza k veľkému nárastu produkcie videa, či už ide o videá zo športových prenosov, televízneho vysielania, filmu alebo ide o amatérske videa zachytávajúce rôzne situácie každodenného života. S touto produkciou súvisí aj extrémny nárast ich ukladania na internet a nahrávania do databáz. S obrovským množstvom takýchto dát vznikla nevyhnutná potreba efektívnej práce, organizovania a indexovania týchto videí podľa ich obsahu. Keďže však objem videí neustále narastá, ručné vykonávanie týchto úloh sa stáva nereálnym. Práve preto začalo vznikať v tejto oblasti značné množstvo prístupov zaoberajúcich sa automatizáciou riešenia týchto úloh. Príkladmi sú prístupy klasifikácie akcie vo videu [20], prístupy sémantického indexovania videí [17] alebo prístupy sumarizácie videa [18]. Všetky tieto prístupy majú spoločné to, že sa snažia extrahovať z videa dôležité informácie (príznaky), ktoré efektívne popisujú video pre danú klasifikačnú úlohu. A práve témou tejto práce sú príznaky pre klasifikáciu videa.

V tejto práci som sa konkrétne zameral na príznaky pre klasifikáciu videa pomocou Bag of Words. Popis štandardne používaných príznakov spolu s klasifikáciou videa pomocou Bag of Words sa nachádza v kapitole 2. Mojim cieľom v tejto práci bolo použiť v systéme pre klasifikáciu videa pomocou Bag of Words také príznaky, ktoré by video popisovali čo najlepšie a to nezávisle na riešenom klasifikačnom probléme. Pre túto úlohu som sa rozhodol vyskúšať učenie príznakov pomocou viacerých metód, ktorými boli PCA vybielenie, Riedke Autoenkódere a Analýza Nezávislých podpriestorov. Popis metód učenia príznakov sa nachádza v kapitole 3. Použitiu zvolených metód učenia príznakov v systéme pre klasifikáciu videa pomocou Bag of Words sa venuje kapitola 4. Táto kapitola obsahuje popis použitia zvolených metód učenia príznakov v procese extrakcie príznakov, ale aj popis extrakcie referenčných príznakov, ktorými boli SIFT deskriptory extrahované z reprezentatívnych snímok videa. Okrem toho tiež táto kapitola popisuje ostatné časti systému klasifikácie videa pomocou Bag of Words, s ktorým prebiehalo testovanie použitých príznakov. V kapitole 4 sa ešte nachádza aj popis navrhnutej metódy inšpirovanej metódami Multiple Kernel Learning, ktorú som použil pre účel kombinácie použitých príznakov s cieľom zlepšiť úspešnosť klasifikácie videa pomocou vytvorenia nového jadra Support Vector Machine navrhnutou metódou z viacerých predpočítaných jadier. Implementačné detaily použitých metód a častí systému klasifikácie videa pomocou Bag of Words obsahuje kapitola 5. Popis sád videí, na ktorých boli experimenty pre jednotlivé typy príznakov a metódy Multiple Kernel Learning vykonávané obsahuje kapitola 6. Popis dosiahnutých výsledkov týchto experimentov a ich porovnanie sa nachádza v kapitole 7. Záverečné zhodnotenie dosiahnutých výsledkov a popis budúcich rozšírení obsahuje kapitola 8.

Kapitola 2

Štandartná klasifikácia videa pomocou Bag of Words

Existuje veľa problémov pri vývoji automatického systému pre klasifikáciu videa. Jedným z týchto problémov je sémantický rozdiel (angl. semantic gap) medzi nízkoúrovňovými príznakmi (príznaky popisujúce obsah videa: histogram farby, energia pohybu, priestorové usporiadanie, tvary, ...) a sémantickou informáciou, ktorú značia (Napríklad na základe príznakov hrán určiť, že zodpovedajú videu, v ktorom človek píše). Okrem toho veľa videí je natáčaných amatérmi alebo videá majú rôzne formáty a podobne. To má za následok videá s rôznym osvetlením, s rôznym pohybom kamery, s rôznym uhlom pohľadu a to ešte umocňuje sémantický rozdiel medzi nízkoúrovňovými príznakmi a sémantickou informáciou videa. S tým sa preto musí systém pre klasifikáciu videa vysporiadať [16].

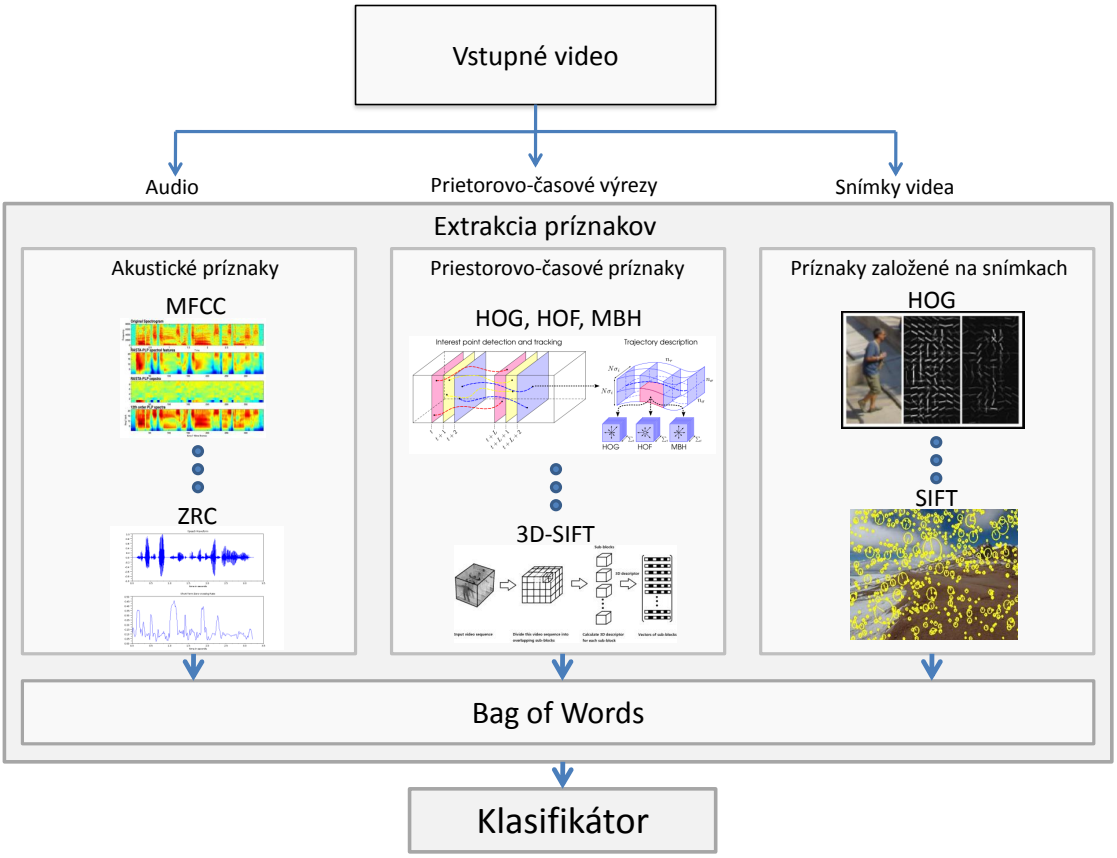
Táto časť práce má za úlohu popísať jeden z možných systémov, ktorý automatickú klasifikáciu vykonáva a je ním štandardný systém pre klasifikáciu videa pomocou Bag of Words.

Štandardný systém pre klasifikáciu videa pomocou Bag of Words (viď. obrázok 2.1) sa skladá z extrakcie príznakov a klasifikácie. Kritickú úlohu v tomto systéme zohráva práve extrakcia príznakov. Dobré príznaky by mali byť odolné voči variantnosti tak, aby klasifikátor dokázal rovnaké triedy za rôznych podmienok stále úspešne klasifikovať.

Pri extrakcií príznakov z videa sa využívajú dva hlavné zdroje videa, ktorými sú vizuálny a akustický kanál. Vizuálny kanál zachytáva vizuálne informácie vzťahujúce sa k objektom na videu, ich pohybu, ale aj pohybu kamery. Vizuálne príznaky extrahované z videa je možné rozdeliť na príznaky založené na snímkach a priestorovo-časové príznaky. Akustický kanál videa obsahuje zvuk okolia, hudbu a podobne. Okrem týchto dvoch hlavných zdrojov existujú práce zaoberajúce sa aj inými zdrojmi, ako je napríklad textový zdroj [6]. Väčšina prístupov sa však zameriava najčastejšie na analýzu vizuálnej stránky videa a ostatné zdroje sa používajú len ako doplnkové práve k obrazu [16].

Ako je vidieť na obrázku 2.1, tak z týchto príznakov sa ďalej vytvára reprezentácia pre celé video alebo časť videa prostredníctvom Bag of Words (BOW) reprezentácie, čo je frekvenčná reprezentácia týchto príznakov [16]. BOW reprezentácia je potom použitá ako vektor príznakov vstupujúci do klasifikátora.

Ako klasifikátor sa často používa Support Vector Machine (SVM), ktorý pre túto úlohu dosahuje veľmi dobré výsledky. Najčastejšie sa používa v kombinácií s χ^2 jadrovou funkciou [31, 17, 36, 20, 33] alebo inou nelineárnou jadrovou funkciou [27, 35]. Okrem SVM sa používa aj Adaboost [24], prípadne iné klasifikátory ako HHM (skryté markovove modely),



Obrázek 2.1: Štandardná klasifikácia videa pomocou Bag of Words s najpoužívanejšími typmi príznakov.

GMM (model zmesí Gaussových funkcií) alebo iné [11, 7, 32].

Štandardné príznaky založené na snímkach sú bližšie popísané v sekcií 2.1, štandardné priestorovo-časové príznaky sú popísané v sekcií 2.2 a akustické príznaky v sekcií 2.3. Vytváranie modelu BOW z týchto príznakov je popísané v sekcií 2.4.

2.1 Príznaky založené na snímkach

Príznaky založené na snímkach sú počítané z jedného snímku. Sú to príznaky využívané pre spracovanie obrazu. Tento typ príznakov neuvažuje časovú informáciu, ale sú často používané vo video analýze, pretože sú ľahko spočítateľné a bolo preukázané, že v praxi dobre fungujú [16, 18, 23, 14].

Pred tým ako však možno spočítať tieto príznaky, je nutné vybrať kľúčové snímky, ktoré budú reprezentovať dané video a z ktorých sa príznaky extrahujú. Tu existuje viacero metód ako tento výber vykonať. Najjednoduchšou metódou je náhodný výber kľúčových snímok z videa alebo výber snímok v určitých časových intervaloch [16]. Problém týchto metód je však to, že nemusia byť vybraté najreprezentatívnejšie snímky alebo sa výberu redundantné. Preto existujú aj špeciálne algoritmy pre výber snímok, medzi ktoré patria sekvenčná alebo zhlukovacia metóda [23]. Sekvenčná metóda vyberá snímky na základe

podobností susedných snímok alebo na základe podobnosti s predchádzajúcou kľúčovou snímkou. Výhodou tejto metódy je, že uvažuje časové usporiadanie snímok a nevýhodou je, že ignoruje celkovú zmenu trendu vo videu. Zhlukovacia metóda zoskupuje snímky do konečného počtu zhlukov vo vybratom priestore príznakov a set kľúčových snímok sa získa ako kolekcia reprezentatívnych snímok zhlukov. Táto metóda neuvažuje časovú zložku, ale dokáže poskytnúť pochopenie vizuálneho obsahu videa.

Príznaky extrahované z určených kľúčových snímok sa rozdeľujú na globálne a lokálne príznaky. Globálna reprezentácia kóduje kóduje celý obraz a jej reprezentácia je založená na celkovom rozložení farieb (farebné momenty), textúry alebo hranových informácií obrazu [16]. Výhodou globálnych príznakov je, že vytvárajú veľmi kompaktnú reprezentáciu snímky. Na druhej strane sú však globálne príznaky citlivé na oklúziu a zmeny okolia. Preto sú vhodné hlavne pre snímky s oddelenými objektmi od pozadia a len s jedným objektom na snímke [22]. Populárne príznaky zahŕňajú histogram farby, momenty farby a Gaborovú textúru. Medzi najpoužívanejšie patrí GIST deskriptor [32], čo je nízko-dimenzionálna reprezentácia scény a predstavuje dominantnú priestorovú štruktúru scény. Veľa týchto globálnych príznakov prijalo mriežkovú reprezentáciu snímky, ktorá berie v úvahu priestorové rozloženie scény. Globálne príznaky sú potom počítane v každej časti mriežky zvlášť a konečná reprezentácia sa získa konkatenáciou príznakov týchto častí. Tento spôsob potom zmierňuje spomenuté nevýhody globálnych príznakov. [16].

Lokálne príznaky využívajú skutočnosť, že video snímka môže byť efektívne reprezentovaná použitím skupiny diskriminačných lokálnych príznakov zo snímky. Vzhľadom na to, že sú lokálne príznaky počítané pre malé oblasti, tak riešia problém globálnych príznakov a sú čiastočne robustné k oklúzii a zmenám pozadia. Extrakcia lokálnych príznakov sa skladá z dvoch krokov, ktorými sú detekcia a popis. Detekcia slúži na lokalizáciu stabilných oblastí v snímke, ktoré majú žiaduce vlastnosti (rýchle zmeny v obraze: hrana, roh a podobne). Tieto lokálne oblasti sú potom použité pre popis snímku [16]. Príkladom takéhoto detektora oblastí je Harris-Laplacov detektor alebo Hessian detektor. Obidva tieto detektory detekujú rohy v snímke pričom Harris-Laplaceov detektor pri tejto úlohe spája 2D Harrisov detektor rohov s reprezentáciou priestoru Gaussovým merítkom. Používaný je tiež Difference-of-Gaussian detektor (DoG), ktorý detekuje beztvaré oblasti, kde stred oblasti sa odlišuje od okolia [25]. Detektory však pri scénach s malými zmenami v snímkach môžu detekovať málo oblastí a tak sa tiež často používa hustý odber vzoriek, ktorý určuje lokálne oblasti prostredníctvom homogénnej mriežky a vykazuje pri klasifikácií podobné výsledky ako pri použití detektorov. Metódy využívajúce aj detektor oblasti a aj hustý odber vzoriek sa ukazujú ako najlepšie [16].

Druhým krokom pri extrakcií lokálnych príznakov je vhodné popísanie týchto detekovaných (určených) oblastí. Cieľom deskriptorov popisujúcich tieto oblasti je, aby boli čiastočne invariantné voči rotácií, osvetleniu, zmene veľkosti a uhlu pohľadu. Z týchto deskriptorov je najznámejší a najpoužívanejší Scale-invariant feature transform (SIFT) deskriptor [16, 23], ktorý podľa mriežky rozdeľuje oblasť na rovnako veľké časti a každá z nich je popísaná histogramom gradientov. Výsledný SIFT je konkatenácia týchto histogramov do 128-rozmerného vektoru. Hlavná myšlienka SIFTu je, že oblasť je reprezentovaná relatívne k dominantnej orientácii, čo poskytuje invariantnosť voči otáčaniu. Existujú rôzne varianty SIFT deskriptorov ako napríklad GLOH deskriptory využívajúce logpolárne umiestnenie mriežky na rozdiel od pravouhlej mriežky v SIFTe alebo PCA-SIFT deskriptory, ktoré redukujú dimenzionalitu SIFTu pomocou PCA a tým ho robia robustnejším voči šumu klasického SIFTu [16]. Ďalšími príkladmi lokálnych deskriptorov sú BRIEF binárny lokálny deskriptor [7], SURF ako rýchly alternatívny deskriptor používajúci odozvu 2D Haarovej vlkovej transformá-

cie [34], HOG (histogram orientovaných gradientov) zachytávajúci rozloženie hrán v obraze alebo LBP (lokálny binárny vzor), ktorý je príznakom textúry a ktorý používa binárne čísla pre označenie každého pixlu snímku porovnaním jeho hodnoty s hodnotami okolitých pixlov [38].

2.2 Priestorovo-časové príznaky

Keďže príznaky založené na snímkach neuvažovali časovú dimenziu vo videu, nemožno zachytiť dôležitý zdroj informácií, ktorým je pohyb. Práve výhodou Priestorovo-časových príznakov je, že uvažujú aj časovú informáciu, pretože počítajú príznaky z malých priestorovo-časových objemov videa. Tieto príznaky sa rozdeľujú do dvoch skupín a to na Priestorovo-časové lokálne príznaky a Deskriptory trajektórií (Trajectory Descriptors) [16].

Priestorovo-časové lokálne príznaky sú založené na lokálnych príznakoch (z 2D) rozšírených do 3D, kde je treťou dimenziou čas. Okrem niektorých globálnych príznakov sa používajú hlavne lokálne príznaky (lokálne oblasti sú malé priestorovo-časové objemy). Podobne ako pri metódach založených na snímkach má aj výpočet týchto príznakov dva kroky, ktorými sú detekcia a popis. Pri detekcií (STIPs - priestorovo-časové oblasti záujmu) sa používajú rôzne rozšírenia detektorov 2D kľúčových oblastí, ktoré sa používajú pri lokálnych príznakoch založených na snímkach. Príkladom je Gaborov filter pre detekciu 3D kľúčových oblastí. Tento detektor je nazývaný Cuboid a hľadá lokálne maximá z odozvy funkcie, ktorá obsahuje jadro 2D Gausovsského vyhľadenia a 1D temporálne Gaborove filtre. Ďalším príkladom je Harrisov rohový detektor [10]. Okrem detektorov je možnosťou ako určiť kľúčové oblasti (podobne ako pri príznakoch založených na snímkach) hustý odber vzoriek. Kľúčové oblasti sú v tejto variante určené pomocou 3D homogénnej mriežky a pri reálnom videu dosahujú podobné a niekedy aj lepšie úspešnosti ako pri použití detektorov [33]. Pre popísanie získaných kľúčových oblastí sa používajú 3D deskriptory inšpirované 2D deskriptormi, ktoré sú používané pre popis 2D lokálnych oblastí. Príkladmi sú rozšírené SIFT alebo SURF descriptory [33]. Ako deskriptory môžu byť použité histogramy orientovaných gradientov HOG alebo histogramy optického toku HOF. HOG zachytáva vzhľad a HOF zachytáva pohyb [16]. Používa sa preto často ich konkatenácia HOG/HOF, čím sa súčasne zachytí pohyb a aj vzhľad popisovanej 3D lokálnej oblasti.

Deskriptory trajektórií pracujú na princípe sledovania lokálnych príznakov v snímkach videa. Keďže sledujú trajektórie lokálnych príznakov v snímkach, tak by mali dosahovať lepšie výsledky ako Priestorovo-časové lokálne príznaky popísané vyššie, ktoré mali predefinovanú 3D štruktúru. Tým, že sa sleduje trajektória lokálnych príznakov v snímkach, sa však značne zvyšuje aj výpočetná náročnosť týchto príznakov [16]. Takýto prístup k extrakcií príznakov používa aj prístup Kanade-Lucas-Tomasi (KLT) tracker pre extrakciu trajektórií kľúčových oblastí popísaných SIFTami a detekciou kľúčových oblastí pomocou DoG metódy rozdielov Gaussových funkcií. Príznaky sú vypočítané na základe modelovania pohybu medzi každým pármi trajektórií [26]. Existujú rôzne rozšírenia tejto metódy a jedným z nich je počítanie troch úrovní trajektórií [35]. Tieto úrovne sú: úroveň kontextu oblasti, čo je priemerný SIFT deskriptor, kontext intra-trajektórie, ktorý kóduje prechody trajektórií v čase a kontext inter-trajektórie, ktorý kóduje vzdialenosť medzi trajektóriami. Hlavný problém pri extrakcií deskriptorov trajektórií je pohyb kamery, ktorý môže ovplyvniť kvalitu príznakov. Deskriptor trajektorie nazývaný Motion Boundary Histogram (MBH) [36, 16] je založený na deriváciách optického toku. Tieto derivácie sa snažia potlačiť konštantný pohyb, čo MBH robí robustným pre pohyb kamery. Záznam rýchlosti trajektórií kľúčových bodov modeluje Messing et al. [28], ktorý pozoroval, že rýchlostná informácia je

užitočná pre detekciu každodenných akcií vo videách s vysokým rozlíšením. Ako už bolo spomenuté, tak hlavnou nevýhodou Deskriptorou trajektórií je ich výpočetná náročnosť.

2.3 Akustické príznaky

Akustický kanál videa neposkytuje takú veľkú informáciu ako vizuálny kanál. Príkladom je zachytenie pohybu objektu vo videu, ktorý je pri väčšine klasifikačných úloh dôležitý. Preto príznaky nedosahujú takú úspešnosť ako vizuálne. Ich použitie v kombinácií s vizuálnymi príznakmi však môže zlepšiť úspešnosť klasifikácie videa [16].

Asi najpoužívanejšími akustickými príznakmi s najlepšou úspešnosťou pri rozpoznávaní udalostí pomocou akustických príznakov [11] sú Mel-frekvenčné kepstrálne koeficienty MFCC [16]. MFCC sú počítané na krátkych úsekoch audia a vyjadrujú energiu spektra tohto úseku signálu. Tento výpočet je na báze kosínusovej transformácie logaritmickej energie spektra v nelineárnej Melovej frekvenčnej mierke. Je možné ich použiť aj s inými príznakmi audia (ZCR, ...). Tieto príznaky sa používajú napríklad pri športe na detekciu situácie ako je faul alebo gól.

Ďalšie príznaky, ktoré sa používajú sú príznaky pre rozpoznávanie reči (ARS automatic speech recognition), podľa ktorých sa hľadá kontext vo videu [16].

2.4 Bag of Words reprezentácia

Po extrakcií hore uvedených príznakov je nutné nejakým spôsobom porovnať podobnosť dvoch rôznych videí, ktoré sa môžu líšiť dĺžkou, rozmermi, či zložitosťou obsahu. To predstavuje problém, keďže väčšina meraní podobnosti a klasifikátorov pracuje s pevným počtom dimenzií.

Možným riešením, tohto problému je hľadanie zodpovedajúcich príznakov medzi dvoma videami a určenie podobnosti videí na základe týchto zodpovedajúcich príznakov, čo je však výpočetne náročné a to aj za pomoci indexovacích štruktúr [16].

Efektívnym riešením tohto problému je Bag of Words (Bag of Features) reprezentácia [12]. Bag of Words (BOW) je štatistický model motivovaný metódou Bag of Words používanou pre reprezentáciu dokumentov. BOW reprezentácia je histogramom vizuálnych slov, ktoré sa získajú prekladom príznakov extrahovaných z videa pomocou vizuálneho slovníka. Slovník je vytvorený natrénovaním na reprezentatívnej vzorke príznakov pomocou trénovacieho algoritmu. Tento algoritmus vytvorí na základe podobnotnej (vzdialenostnej) funkcie a trénovacej množiny príznakov zhluky, ktorých stredmi sú vizuálne slová. Príznakový vektor sa potom preloží tak, že sa na základe vzdialenostnej funkcie určí zhluk, do ktorého patrí a daný príznakový vektor bude reprezentovaný pomocou vizuálneho slova tohto zhluku.

Je viacero možností ako jednotlivé časti tohto modelu implementovať alebo nastaviť (trénovací algoritmus, vzdialenostná funkcia, veľkosť slovníka ...) a rôzne implementácie teda majú rôzne výsledky [16]. Existuje veľa štúdií, ktoré sa snažia nájsť najoptimálnejšie nastavenie a implementáciu BOW pre konkrétnu úlohu.

Jaing et al. [17] vykonal sériu analýz niekoľkých typov BOW vo video klasifikácii vrátane váhových schém a veľkostí slovníka. Dôležitým poznatkom je, že váha termov je veľmi dôležitá a soft-váhovacia schéma zmierňuje účinky kvantizačných chýb, ktoré sa ešte zväčšujú pre viac-dimenzionálne vstupné vektory. Gemert et al. navrhujú, ako soft-váhovaciu metódu použiť postup neurčitého kódového slova (codeword uncertainty). Tento

spôsob zaraďuje príslušný vstupný vektor s určitou váhou ku každému kódovému slovu, pričom súčet váh je jedna. Gemert et al. ďalej uvádzajú, že najčastejšie používaný trénovací algoritmus je algoritmus k-means.

Čo sa týka veľkosti slovníku, tak sa ukazuje, že pre väčšinu klasifikačných úloh postačuje slovník niekoľkých stoviek až tisícok vizuálnych slov [16].

Čo sa týka výhod BOW reprezentácie a nevýhod pri jej použití, tak hlavnou nevýhodou je, že zanedbáva priestorovú lokalizáciu príznakov vo videu. Tým sa môžu stratiť dôležité informácie. Ďalšou nevýhodou je neschopnosť získať hlboké sémantické porozumenie videa. To je preto, že poskytuje kompaktnú reprezentáciu komplexnej udalosti vo videu na základe podkladových príznakov bez pochopenia hierarchickej štruktúry komponentov. Okrem toho BOW trpí obvyklými nevýhodami kvantovania (diskretizácia) [16].

Napriek tomu je však táto technika používaná a dosahuje dobré výsledky [16, 20, 31, 24] a ako už bolo spomenuté je kompaktná a dovoľuje jednoduché porovnávanie častí videí alebo celých videí. Okrem toho BOW zabezpečuje nezávislosť systému na extrahovaných príznakoch.

Kapitola 3

Učenie príznakov

Štandardne používané príznaky pre klasifikáciu (nielen) videa sú takzvane ručne-navrhnuté. Sú to príznaky, pre ktorých zisk bol navrhnutý algoritmus, ktorý počíta príznaky nezávisle na vstupných dátach a teda pre všetky dátové sady rovnako. Medzi tieto príznaky patria príznaky ako SIFT, HOG, HOF a podobne. Ručne-navrhnuté príznaky však majú viacero nevýhod, medzi ktoré patrí napríklad ich časovo náročné ladenie pre rôzne dátové sady a zložitá rozšíriteľnosť na rôzne typy vstupných dát (text, výstupy dialkomeru, ...). Veľkou nevýhodou však je, že napriek tomu neexistujú ručne-navrhunté príznaky, ktoré by pre všetky video dátové sady a klasifikačné úlohy fungovali vždy rovnako dobre. To pri experimentoch v rozpoznávaní videa potvrdili aj Wang et al [37]. Je to spôsobené práve tým, že ručne-navrhnuté príznaky sú nezávislé na vstupných videách. Vzniká teda snaha vyriešiť tento problém a jednou z možností, ktorá sa naskytuje ako riešenie je oblasť učenia príznakov [20].

Základom metód učenia príznakov je učenie, ktoré môže prebiehať s učiteľom (s olabelovanými dátami) alebo bez učiteľa (s neolabelovanými dátami). V súčasnosti učenie príznakov prebieha skoro výhradne ako učenie bez učiteľa [9]. Cieľom metód tohto učenia príznakov je prostredníctvom učenia získať z jednotlivých neolabelovaných vstupných vzorov nové reprezentácie tak, aby tieto nové reprezentácie odrážali štatistickú štruktúru a závislosti celkovej množiny vstupných vzorov. Práve vďaka tomu, že sú tieto nové reprezentácie získané učením z dát, sú na rozdiel od ručne-navrhnutých príznakov prispôsobené pre konkrétny typ vstupných dát (napríklad rôzne typy dátových sád videí).

Asi najstaršou metódou učenia príznakov je Analýza hlavných komponentov (PCA) [4]. Táto metóda využíva na transformáciu vstupných dát takzvané vlastné vektory kovariančnej matice, získanej z trénovacej množiny dát. Tieto vlastné vektory sú potom využité ako bázy, prostredníctvom ktorých sú vstupné dáta prevedené na ich dekorelovanú reprezentáciu. Tým možno oddeliť od seba triedy dát, ktoré sa pred transformáciou prekrývali. Výhodou tejto metódy je jej rýchlosť, ale v súčasnosti sa používa veľmi často len ako jeden z krokov predspracovania dát.

Analýza hlavných komponentov však nie je jedinou metódou učenia príznakov. Existuje viacero prác zaoberajúcich sa rôznymi metódami učenia príznakov a ich využitím v klasifikácií. Jednou z nich je aj práca, ktorú vytvorili Coates et al. a zaoberá sa využitím metód učenia príznakov bez učiteľa pri rozpoznávaní obsahu obrazu [9]. Táto práca porovnáva úspešnosť klasifikácie príznakov získaných štyrmi metódami učenia príznakov, ktorými sú Riedke Autoenkodére, Riedky Obmedzený Boltznamov stroj, K-means a Modely zmesí Gaussových funkcií. Autonekodér je neurónová sieť, ktorá prostredníctvom skrytej vrstvy rekonštruuje svoj vstup. Nová reprezentácia vstupného vektoru je z Autoenkodéru získaná

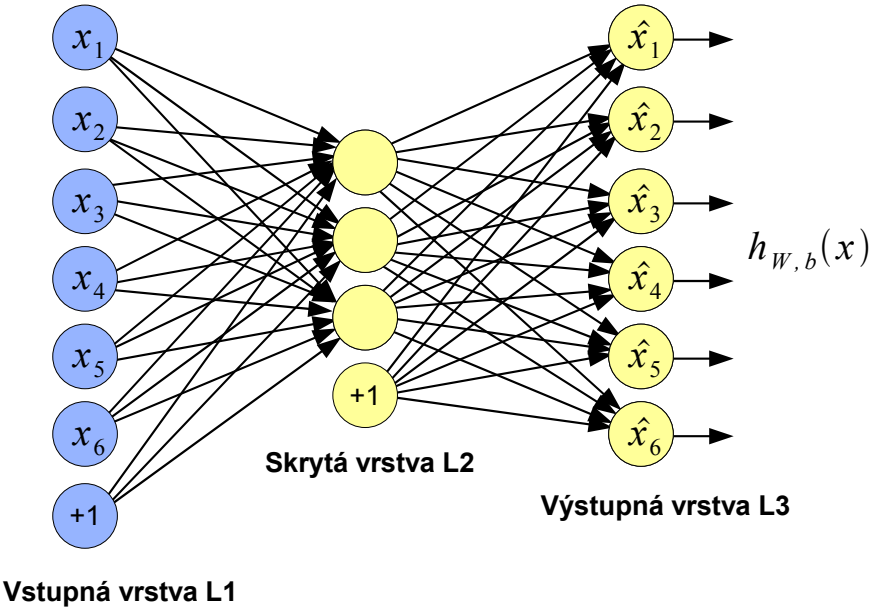
z aktivácií neurónov skrytej vrstvy. Zmenou počtu neurónov skrytej vrstvy teda možno kontrolovať dimenzionalitu novej reprezentácie vstupného vektora. Riedke Aukoenkodére navyše zavádzajú do skrytej vrstvy kontrolu riedkosti. Obmedzený Boltzmanov stroj je stochastická neurónová sieť vychádzajúca z Hopfieldových sietí a predstavujúca bipartitný graf rozdeľujúci sa na viditeľnú a skrytú vrstvu. Podobne ako v Autoenkodéroch sa aj tu nová reprezentácia vstupného vektoru získa z aktivácií neurónov skrytej vrstvy. Riedky obmedzený Boltznamov stroj rozširuje Obmedzený Boltznamov stroj o kontrolu riedkosti v skrytej vrstve. K-means spolu s Modelmi zmesí Gaussových funkcií sú zhlukovacie algoritmy hľadajúce vo vstupných dátach zhluky. K-means k tomu využíva vzdialenostnú funkciu a reprezentáciu zhlukov pomocou centroidov. Modely zmesí Gaussových funkcií zasa pre popis zhlukov využívajú zmesicu Gaussových funkcií.

Coates et al. zamerali svoje experimenty na testovanie rôznych nastavení a porovnávanie úspešností klasifikácie týchto metód učenia príznakov pre obrazové dáta. Vo svojich experimentoch skúmali aj vplyv ZCA (Zero component analysis) vybielenia vstupných dát[1] na úspešnosť klasifikácie testovaných metód učenia príznakov. Z výsledkov je vidieť ako Autoenkodére a Obmedzené Boltzmanove stroje lokalizujú bázy podobné Gaborovým filtrom a to pre dáta bez ZCA vybielenia, ale aj pre ZCA vybielené dáta, kde to vidieť dokonca lepšie. Avšak podobné filtre získali Coates et al. aj pomocou zhlukovacích algoritmov na ZCA vybielených dátach. Pri vybielených dátach dosiahli zhlukovacie metódy ostro lokalizované filtre podobné filtrom lokalizovaným prostredníctvom Riedkych Autoenkodérov a Riedkymi Obmedzenými Boltzmanovými strojmi. Coates et al. dokonca s K-means na ZCA vybielených dátach dosiahli zo všetkých metód učenia príznakov najlepšiu úspešnosť klasifikácie. To znamená, že je možné získať podobné príznaky jednoducho pomocou zhlukovacích algoritmov na vybielených dátach bez potreby ladenia rôznych parametrov ako tomu je pri iných metódach učenia príznakov. Coates et al. uvádzajú, že ZCA vybielenie je zásadným krokom v preprocesingu pre zhlukovacie algoritmy, pretože tieto metódy si neporadia s korelovanými dátami. Z experimentov však zistili, že ZCA vybielenie naopak nemá skoro žiadny vplyv pre úspešnosť klasifikácie príznakov získaných Riedkymi Autoenkodérmi a Riedkymi Obmedzenými Boltzamovými strojmi.

Ďalšou prácou, ktorá skúma úspešnosť klasifikácie príznakov získaných metódou učenia príznakov je práca, ktorú vytvorili Le et al [20]. Táto práca skúma úspešnosť klasifikácie videa hierarchickými invariantnými temporálnymi príznakmi vytvorenými pomocou Analýzy nezávislých podpriestorov (Indepentdant Subspace Analysis). Analýza nezávislých podpriestorov je metóda reprezentovaná neurónovou sieťou s dvomi vrstvami, kde každý neurón výstupnej vrstvy zlučuje určitý počet neurónov skrytej vrstvy a vytvára tak jeden podpriestor. Pri trénovaní dochádza k lokalizácií filtrov prvej vrstvy, ktoré sú pre každý podpriestor podobné. Vďaka tejto podobnosti potom Analýza nezávislých podpriestorov produkuje príznaky, ktoré sú čiastočne invariantné voči posunu. Le et al. využili Analýzu nezávislých komponentov k popisu temporálnych výrezov videa a ukázujú, že jedna vrstva Analýzy Nezávislých komponentov dokáže vo videu detekovať pohybujúce sa hrany v čase. Le et al. ďalej prostredníctvom viacerých vrstiev sietí Analýzy nezávislých podpriestorov vytvárali hierarchické príznaky, ktoré sú schopné vo videu detekovať high-level štruktúry a dosahovali pri klasifikácií videa veľmi dobré výsledky.

Nasledujúce sekcie sú zamerané na bližší popis dvoch vybratých prístupov učenia príznakov a to na Riedke Autoenkodére v sekcií 3.1 a Analýzu Nezávislých podpriestorov v sekcií 3.2.

[1] Princíp podobný ako pri PCA vybielení s tým rozdielom, že dáta sú naspäť prevedené do vstupného priestoru.



Obrázek 3.1: Neurónová sieť s jednou skrytou vrstvou reprezentujúca Riedky Autoenkodér. Neurón označený ako +1 reprezentuje bias. Vstupná vrstva a výstupná vrstva má rovnaký počet neurónov. Musí platiť, že výstup $h_{W,b}(x) \approx x$, kde $x = \{x_1, x_2, \ldots, x_6\}$ je vstupný vektor.

3.1 Riedke Autoenkodére

Autoenkodér je typ algoritmu učenia bez učiteľa. Ide o neurónovú sieť vykonávajúcu aproximáciu funkcie identity. To znamená, že pre vstupný vektor $x = \{x_1, \ldots, x_m\}$ sa na výstupe siete očakáva vektor $\hat{x} = \{\hat{x_1}, \ldots, \hat{x_m}\}$, ktorý je podobný ako vstupný vektor x. Aby toto pravidlo platilo je zrejmé, že vstupná a výstupná vrstva bude mať rovnaký počet neurónov. Nová reprezentácia vstupného vektoru sa po natrénovaní Autoenkodéru získa z aktivácií neurónov skrytých vrstiev. Počet skrytých vrstiev v Autoenkodére môže byť obecne rôzny.

Riedke Autoenkodére [29], na ktoré sa táto sekcia zameriava sú špeciálnym typom Autoenkodérov obsahujúcim práve jednu skrytú vrstvu. Vlastnosť, ktorú Riedke Autoenkodére na rozdiel od obecných Autoenkodérov zabezpečujú je takzvaná kontrola riedkosti v skrytej vrstve. Cieľom riedkosti skrytej vrstvy je zabezpečiť, aby väčšina neurónov skrytej vrstvy bola neaktívna. V takom prípade bude Riedky Autoenkodér stále objavovať zaujímavú štruktúru v dátach aj pri väčších počtoch neurónov v skrytej vrstve [29].

Príklad štruktúry Riedkeho Autoenkodéru reprezentuje obrázok 3.1, kde $h_{W,b}(x)$ je vektor funkcie výstupu Riedkeho Autoenkodéru pre vstupný vektor $x = \{x_1, \ldots, x_6\}$. Ako možno ďalej vidieť, tak neuróny majú špeciálny vstup, ktorým je bias, čo zodpovedá popisu Riedkych Autoenkodérov v práci od Ngho [29]. Výstup neurónu a s indexom i, vo vrstve j, pre vstup $x^{(i)}$ je potom definovaný ako $a_i^{(j)} = f(W_i^{(j-1)} x^{(i)} + b_i^{(j-1)}$, kde f je aktivačná funkcia. Ako aktivačnú funkciu Ng vo svojej práci používa sigmoidu, ale možno použiť aj hyperbolický tangens.

Aby platila požiadavka $h_{w,b}(x) \approx x$, je nutné váhy a biasy Riedkeho Autoenkodéru natrénovať pre zadanú trénovaciu sadu vstupných vektorov. Ng vo svojej práci [29] využíva k tomuto účelu metódu gradientneho zostupu, ktorá využíva pre aktualizáciu premenných

parciálne derivácie objektívnej funkcie podľa týchto premenných. Aktualizácia váh (W) a biasov (b) neurónov teda prebieha pomocou parciálnych derivácií príslušnej objektívnej funkcie. Ng vo svojej práci využíva objektívnu funkciu $J_{sparse}(W,b)$, ktorá je pre m trénovacích vzorov definovaná ako

$$J_{sparse}(W,b) = \left[\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\|h_{W,b}(x^{(i)}) - x^{(i)}\|^2\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}\left(W_{ji}^{(l)}\right)^2 + \beta\sum_{j=1}^{s_2} KL(\rho\|\hat{\rho_j}), \tag{3.1}$$

kde prvý term je priemerná chyba súčtu štvorcov rozdielu vstupných a výstupných vektorov Riedkeho Autoenkodéru a druhý term je regularizačný term, ktorý slúži k zníženiu veľkosti váh a snaží sa zabrániť pretrénovaniu. Tento proces je kontrolovaný pomocou parametru λ. Posledný term slúži k spomínanej kontrole riedkosti skrytej vrstvy (váha tohto termu je riadená parametrom β). Cieľom je, aby väčšina neurónov skrytej vrstvy bola neaktívna. Neurón je aktívny ak je jeho výstup blízky hodnote 1 a neaktívny ak je jeho výstup blízky hodnote 0 (pre hyperbolický tangens hodnote -1). Tento tretí term sa nazýva Kullback-Leiblerov (KL) rozdiel a vyjadruje rozdiel medzi Bernoulliho náhodnou premennou so strednou hodnotou ρ a Bernoulliho náhodnou premennou so strednou hodnotou $\hat{\rho_j}$. KL-divergencia je štandardná funkcia pre meranie, ako rozdielne sú dve rôzne distribúcie a spočíta sa ako

$$KL(\rho\|\hat{\rho_j}) = \rho\log\frac{\rho}{\hat{\rho_j}} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho_j}}, \tag{3.2}$$

kde ρ je parameter riedkosti (malá hodnota blízka 0) a $\hat{\rho_j}$ je priemerná aktivácia neurónu j cez všetky trénovacie vzorky ($\hat{\rho_j} = \frac{1}{m}\sum_{i=1}^{m}\left[a_j^{(2)}(x^{(i)})\right]$). Na miesto Kullback-Leiblerov (KL) rozdielu je však možné použiť aj inú variantu termu pre kontrolu riedkosti.

Aktualizácia váh a biasov potom prebieha pomocou gradientneho zostupu v jednom kroku nasledovne

$$W_{ij}^{(l)} = W_{ij}^{(l)} - \alpha\frac{\partial}{\partial W_{ij}^{(l)}}J\big(W,b\big) \tag{3.3}$$

$$b_i^{(l)} = b_i^{(l)} - \alpha\frac{\partial}{\partial b_i^{(l)}}J\big(W,b\big), \tag{3.4}$$

kde α je trénovacia konštanta. Pre výpočet týchto parciálnych derivácií objektívnej funkcie Ng používa algoritmus Backpropagation. Parciálne derivácie objektívnej funkcie podľa W a b potom možno ďalej rozpísať na

$$\frac{\partial}{\partial W_{ij}^{(l)}}J\big(W,b\big) = \left[\frac{1}{m}\sum_{i=1}^{m}a_j^{(l)}\delta_i^{(l+1)}\right] + \lambda W_{ij}^{(l)} \tag{3.5}$$

$$\frac{\partial}{\partial b_i^{(l)}}J\big(W,b\big) = \frac{1}{m}\sum_{i=1}^{m}\delta_i^{l+1}, \tag{3.6}$$

kde $\delta_i^{(l)}$ je miera ako veľmi je uzol i na vrstve l zodpovedný za chybu vo výstupe neurónovej siete. Pre výstupnú vrstvu sa $\delta_i^{(3)}$ spočíta nasledovne

$$\delta_i^{(3)} = \frac{\partial}{\partial z_i^{(3)}}\frac{1}{2}\|y - h_{W,b}(x)\|^2 = -(y_i - a_i^{(3)})\cdot f'(z_i^{(3)}) \tag{3.7}$$

a pre skrytú vrstvu bude $\delta_i^{(2)}$ vyzerať ako

$$\delta_i^{(2)} = \left(\left(\sum_{j=1}^{s_2} W_{ji}^{(2)} \delta_j^{(3)} \right) + \beta \left(- \frac{\rho}{\hat{\rho}_i} + \frac{1-\rho}{1-\hat{\rho}_i} \right) \right) f'(z_i^{(2)}), \tag{3.8}$$

kde prostredníctvom konštanty β je kontrolovaná úroveň penalizácie riedkosti.

Keďže nová reprezentácia vstupného vektoru sa získa ako vektor aktivácií neurónov skrytej vrstvy Riedkeho Autoenkodéru, možno veľkosť novej reprezentácie vstupu kontrolovať. Možno tak pre vstupný vektor vytvoriť jeho komprimovanú reprezentáciu. Avšak ak sú vstupné vektory z náhodného rozloženia a nie sú na sebe nijako závislé, tak kompresia dimenzií bude veľmi zložitá. Ak však medzi vstupnými vektormi existujú závislosti, tak Riedke Autoenkodére sú schopné objaviť niektoré z týchto závislostí [29].

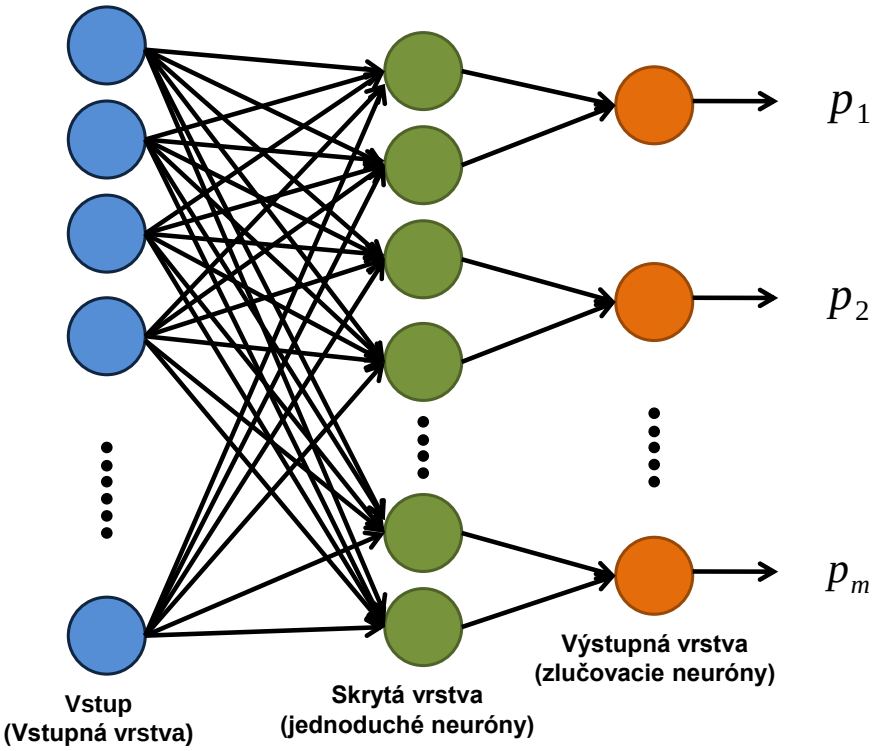
Riedke Autoenkodére však okrem komprimovanej reprezentácie dokážu získať aj high-dimenzionálnu reprezentáciu vstupného vektoru. To je možné vďaka kontrole riedkosti v skrytej vrstve. Pretože práve kontrola riedkosti zabezpečuje objavovanie zaujmavých štruktúr v dátach aj napriek väčšiemu počtu neurónov skrytej vrstvy než je dimenzionalita vstupu [29].

Nevýhodou vyššie popísaných Riedkych Autoenkodérov je obmedzenosť aktivačných funkcií vzhľadom na rozsah hodnôt. To znamená, že aby bolo možné sieť natrénovať správne, musia byť hodnoty vstupných vektorov v intervale od 0 do 1 pre sigmoidu alebo od -1 do 1 pre hyperbolický tangens. Toto obmedzenie však možno odstrániť zmenou aktivačných funkcií neurónov vo výstupnej vrstve. Pre tento účel sa používa lineárna aktivačná funkcia, v ktorej výstup je totožný s výstupom bázovej funkcie. Riedke Autoenkodére s lineárnou aktivačnou funkciou v neurónoch vo výstupnej vrstve sa nazývajú tiež Lineárne Dekodére [30].

Príkladom použitia Riedkych Autoenkodérov pri extrakcií príznakov je práca, ktorú vytvorili Coates et al. [9] a v ktorej sú porovnávané úspešnosti klasifikácie príznakov získaných pomocou rôznych metód učenia príznakov pri klasifikácií obsahu obrazu. Výsledky, ktoré Coates et al. dosiahli pomocou Riedkych Autoenkodérov, sú veľmi podobné výsledkom, ktoré dosiahli Riedke Obmedzené Boltzmanove stroje. V tejto práci je ďalej ukázané, že Riedke Autoenkodéri produkujú filtre podobné Gaborovým filtrom. Coates et al. skúmali aj vplyv ZCA vybielenia dát v preprocesingu na úspešnosť klasifikácie metód učenia príznakov a zistili, že ZCA vybielenie nemá vplyv na dosiahnutú úspešnosť klasifikácie príznakov vytvorených pomocou Riedkych Autoenkodérov.

3.2 Analýza nezávislých podpriestorov

Analýza nezávislých podpriestorov (ISA) je algoritmom učenia bez učiteľa, ktorý je popísaný neurónovou sieťou. Štruktúra ISA siete je zobrazená na obrázku 3.2. Ako možno na obrázku vidieť, tak ISA sieť je okrem vstupnej vrstvy tvorená aj skrytou a výstupnou vrstvou. Aktivačnou funkciou neurónov skrytej vrstvy je druhá mocnina a pre neuróny výstupnej vrstvy sa používa aktivačná funkcia reprezentovaná druhou odmocninou [20]. Neuróny skrytej vrstvy ISA siete sú tiež nazývané jednoduché neuróny a neuróny výstupnej vrstvy ISA siete sú nazývané zlučovacie. Váhy W vstupov neurónov skrytej vrstvy sa pri procese učenia menia, pričom váhy V vstupov neurónov výstupnej vrstvy sú určené náhodne a zostávajú fixné. Pomocou váhovej matice V reprezentujú neuróny výstupnej vrstvy štruktúru podpriestoru neurónov skrytej vrstvy. Každý neurón výstupnej vrstvy zoskupuje



Obrázek 3.2: Neurónová sieť reprezentujúca ISA sieť. Aktivačná funkcia neurónov skrytej vrstvy (zelené neuróny nazývané tiež jednoduché) je druhá mocnina a aktivačná funkcia neurónov výstupnej vrstvy (oranžové neuróny nazývané tiež zlučovacie) je druhá odmocnina. Veľkosť podpriestoru v tomto prípade je 2 (každý zlučovací neurón zlučuje 2 susedné jednoduché neuróny).

určitý počet susedných neurónov skrytej vrstvy, čím určuje veľkosť svojho podpriestoru. To je vidieť aj na obrázku 3.2, kde jeden podpriestor tvoria dva neuróny skrytej vrstvy.

Výstup ISA siete pre vstupný vektor x^t je potom pre každý neurón výstupnej vrstvy definovaný ako

$$p_i(x^t; W, V) = \sqrt{\sum_{k=1}^{m} V_{ik} \Big(\sum_{j=1}^{n} W_{kj} x_j^t \Big)^2}, \tag{3.9}$$

kde m značí počet neurónov výstupnej vrstvy a n značí počet dimenzií vstupného vektoru. Cieľom trénovania ISA je minimalizovať výstup siete získaný pre celú trénovaciu sadu $\sum_{t=1}^{T} \sum_{i=1}^{m} p_i(x^t; W, V)$, prostredníctvom zmeny váh W vstupov skrytej vrstvy, ale aby zároveň platilo, že $WW^T = I$, kde I je maticou identity. To platí len vtedy, ak matica váh W je ortonormálna. Toto obmedzenie na ortonormalitu matice váh W zabezpečuje, aby získané príznaky boli rôznorodé [20]. Objektívnou funkciou ISA je teda výstup siete pre celú trénovaciu sadu (vzhľadom na zachovanie ortonormality váh W).

Le et al. vo svojej práci využili na trénovanie ISA projekčný gradientny zostup, pracujúci s gradientom objektívnej funkcie ISA. Pre kroky gradientneho zostupu sú využité projekčné kroky na dosiahnutie obmedzenia na ortonormalitu. Jedna iterácia projekčného

gradientneho zostupu teda bude vyzerať nasledovne

$$1. \quad W \leftarrow \alpha \nabla_W \sum_{t=1}^{T} \sum_{i=1}^{m} p_i(x^t; W, V) \tag{3.10}$$

$$2. \quad W \leftarrow \text{proj}_U W \text{ kde U je priestor matíc spĺňajúci } WW^T = I. \tag{3.11}$$

Projekčný krok projekčného gradientneho zostupu je dosiahnutý nastavením

$$W \leftarrow (WW^T)^{-\frac{1}{2}} W, \tag{3.12}$$

čo môže byť videné ako ZCA vybielenie a zodpovedá to kubickej časovej náročnosti. Pre veľké dimenzionálne vstupy bude algoritmus pomalý. Avšak Le et al. používajú ISA pre malé temporálne výrezy videa, ktorých dimenzionalita bola zredukovaná PCA vybielením. Tým značne zrýchľujú natrénovanie ISA pri použití jednoduchého trénovania, pretože gradientny zostup nepotrebuje ladiť žiadne špeciálne trénovacie konštanty a pravidlá [20].

V ukážkach filtrov lokalizovaných pri trénovaní ISA na výrezoch prirodzeného obrazu, ktoré vo svojej práci Le et al. prezentovali, je vidieť, že tieto filtre pripomínajú Gaborové filtre. Ďalej je ukázaná vlastnosť zlučovacích neurónov, ktoré zlučujú podobné filtre do jedného podpriestoru, čo zabezpečuje získanie invariantnosti príznakov. Le et al. vykonali ohľadom tejto invariantnosti experiment s obrazovými dátami a zistili, že zlučovacie neuróny sú robustné voči translácií, ale selektívne voči frekvenčným a rotačným zmenám, čo robí získané príznaky vysoko invariantné.

Le et al. ukazujú tiež spôsob vytvorenia hierarchických príznakov, pre video, pomocou viacerých vrstiev ISA sieti, čím získali príznaky schopné detekovať high-level štruktúry vo videu. Úspešnosť klasifikácie videa týchto príznakov dosahuje, v porovnaní so štandardne používanými príznakmi videa, veľmi dobré výsledky.

Kapitola 4

Popis použitých príznakov v systéme pre klasifikáciu videa pomocou Bag of Words

Táto časť práce je určená k popisu metód extrakcie príznakov, ktoré som použil v tejto práci v systéme pre klasifikáciu videa pomocou Bag of Words. Táto časť teda okrem popisu použitých príznakov obsahuje aj popis vytvorenia BOW reprezentácie a popis použitého klasifikátora. Štruktúra použitého systému pre klasifikáciu videa je zobrazená na obrázku 4.1. V závere tejto kapitoly sa nachádza popis jednoduchej metódy Multiple Kernel Learning, ktorú som navrhol pre vylepšenie úspešnosti klasifikácie videa pomocou kombinácie predpočítaných jadier BOW reprezentácií viacerých typov príznakov.

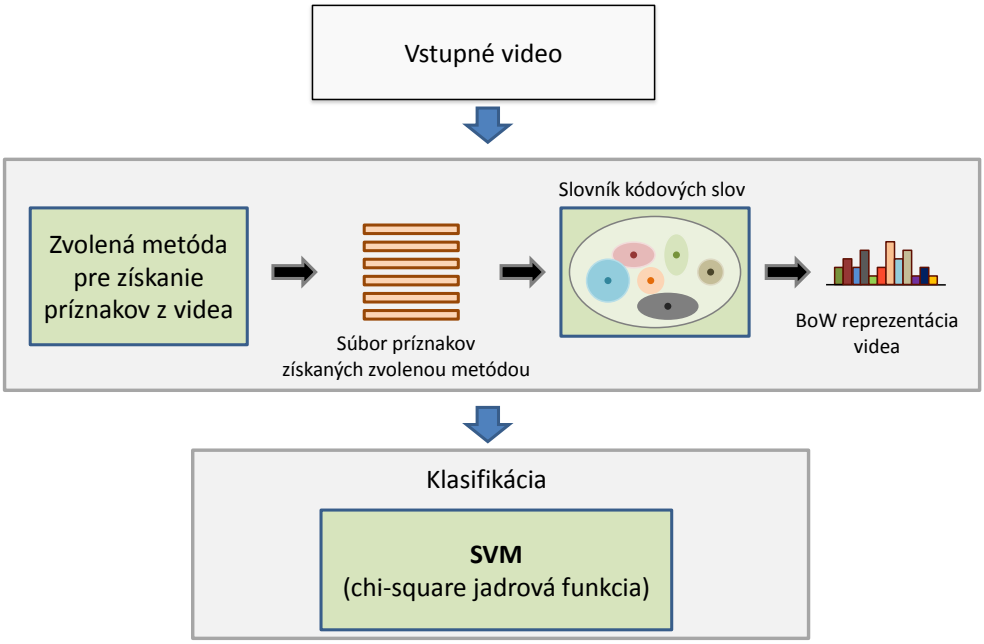
Ako už bolo v kapitole 2 uvedené, tak video obsahuje viacero informačných kanálov, ktoré sa používajú pre popis videa. Vo svojich experimentoch som zúžil extrahovanie príznakov na extrakciu príznakov z vizuálneho kanálu videa. Tento informačný kanál som zvolil preto, lebo je v ňom možné zachytiť informácie dôležité pre klasifikáciu obsahu videa ako napríklad pohyb, typ objektov, ich činnosť a podobne.

Mojim cieľom bolo použiť také príznaky z vizuálneho kanála videa, ktoré by prostredníctvom BOW reprezentácie popisovali video čo najlepšie a to nezávisle na úlohe klasifikácie videa. Pre túto úlohu som sa rozhodol vyskúšať extrahovať príznaky z vizuálneho kanála videa pomocou metód učenia príznakov, pretože výsledné príznaky sa tak dokážu prispôsobiť pre príslušnú dátovú sadu pomocou trénovania. Bližší popis použitia metód učenia príznakov pri popise priestorovo-časových výrezov videa obsahuje sekcia 4.2.

Ako referenčné príznaky som použil SIFT deskriptory extrahované z kľúčových oblastí vybratých snímok videa. Dosiahnuté výsledky klasifikácie je tak možné pre príznaky získané metódami učenia príznakov porovnať s ručne-navrhnutými príznakmi. Bližší popis extrakcie týchto SIFT deskripotorov sa nachádza v sekcií 4.1.

Ako už bolo vyššie uvedené, tak jednotlivé príznaky sú otestované so systémom pre klasifikáciu videa pomocou Bag of Words. Pre tento účel je najprv potrebné vytvoriť slovník z reprezentatívneho množstva vektorov príznakov videí trénovacej sady. K vyhľadaniu prototypov tvoriacich slovník, nazývaných tiež kódové slová, som použil algoritmus K-means s Euklidovskou vzdialenosťou. Takýmto spôsobom bol slovník vytvorený pre každý typ príznakov.

Vytvorený slovník je potom možné použiť na vytvorenie BOW reprezentácie zo súboru vektorov príznakov videa. Aby som zmiernil kvantizačnú chybu, ktorá vzniká jednoduchým



Obrázek 4.1: Štruktúra systému pre klasifikáciu videa, s ktorým sú testované použité typy príznakov.

priradením vektora ku kódovému slovu a ktorá pri viac-dimenzionálnych vstupných vektoroch ešte viac narastá, rozhodol som sa v tejto práci použiť pri tvorbe BOW postup neurčitého kódového slova (codeword uncertainty) [12]. Tento spôsob zaraďuje príslušný vstupný vektor s určitou váhou ku každému kódovému slovu pričom súčet váh je jedna. Takže BOW reprezentácia pomocou neurčitého kódového slova pre sadu vektorov príznakov P, pre každé kódové slovo w, zo slovníku B vyzerala nasledovne

$$UNC(w) = \sum_{p \in P} \frac{K(w,p)}{\sum_{b \in B} K(b,p)}, \tag{4.1}$$

kde K je jadrová funkcia. V tejto práci je použitá Gaussová jadrová funkcia definovaná ako

$$K(w,w') = \mathrm{e}^{-\frac{\|w-w'\|_2^2}{2\sigma^2}}, \tag{4.2}$$

kde σ je veľkosť jadra, v tomto prípade nastavená na priemernú vzdialenosť medzi kódovými slovami v slovníku. Výslednú BOW reprezentáciu som pred vstupom do klasifikátora normalizoval tak, aby súčet hodnôt BOW reprezentácie bol 1.

Pre klasifikáciu bol použitý Support Vector Machines s χ^2 jadrovou funkciou. Tento klasifikátor s χ^2 jadrovou funkciou som zvolil kvôli dobrej úspešnosti klasifikácie, ktorú pri rozpoznávaní videa dosahujú (viď. kapitola 2). Použitá χ^2 jadrová funkcia je definovaná ako

$$K(H,H') = \mathrm{e}^{-\gamma \sum_{n=1}^{V} \frac{(h_n - h'_n)^2}{h_n + h'_n}}, \tag{4.3}$$

kde $H = \{h_n\}$ a $H = \{h'_n\}$ sú dve Bag of Words reprezentácie a V je počet kódových slov v slovníku. Optimálne hodnoty SVM regulačného parametru C a škalovací parameter jadra γ som hľadal pomocou grid search a 6-stupňovej crossvalidácie.

Takto zostavený systém pre klasifikáciu videa bol využitý pre otestovanie každého typu použitých príznakov na príslušnej dátovej sade. Pre vylepšenie úspešnosti klasifikácie videa som sa rozhodol inšpirovať metódou Multiple Kernel Learning (MKL) [13]. Táto metóda používa predpočítané jadrá pre rôzne typy príznakov k vytvoreniu novej jadrovej funkcie získanej kombináciou týchto predpočítaných jadier. Geonen et al. vo svojej práci popisujú rôzne metódy ako nové jadro vytvoriť pomocou rôzneho váhovania predpočítaných jadier. Vo svojej práci som sa ale rozhodol použiť nasledujúci postup, ktorý výslednú jadrovú funkciu získaval priemerovaním predpočítaných jadier (rôznych BOW reprezentácií videí) a váha jadier bola určená iba škálovacím parametrom jadra γ pri predpočítavaní príslušného jadra. Jadrá boli teda predpočítané z BOW reprezentácií videí príslušnej dátovej sady podľa vzťahu 4.3 a výsledné jadro $K_{mult}(H, H')$, ktoré som v SVM potom používal bolo získané ako

$$K_{mult}(H, H') = \sum_{i=1}^{N} K_i(H, H')/N, \tag{4.4}$$

kde N je počet predpočítaných jadier a $K_i(H, H')$ je jedno predpočítané jadro spočítané pre BOW reprezentácie zvolených typov príznakov. Optimálne hodnoty SVM regulačného parametru C som hľadal pomocou grid search a 6-stupňovej crossvalidácie (škálovací paramater γ jadra už nebolo nutné hľadať).

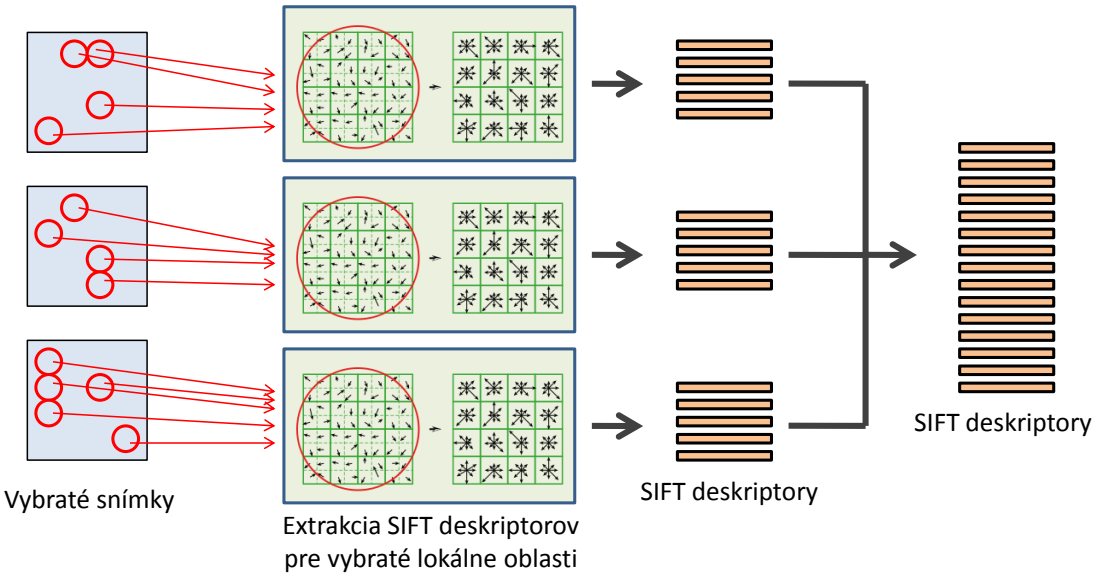
4.1 Extrakcia SIFT deskritorov z kľúčových oblastí snímok

Ako referenčné príznaky videa som zvolil SIFT deskriptory extrahované z kľúčových oblastí reprezentatívnych snímok videa. Výpočet týchto príznakov reprezentuje obrázok 4.2, kde je vidieť, že výsledné SIFT deskriptory kľúčových oblastí snímok sú zlúčené do jedného súboru a stráca sa teda informácia o tom, z ktorej snímky boli SIFT deskriptory získané. Výber snímok z videa, pre ktoré boli SIFT deskriptory z kľúčových oblastí extrahované som z dôvodu jednoduchosti (keďže ide o referenčné príznaky) vykonával v malých pravidelných intervaloch. SIFT deskriptory som zvolil preto, lebo zabezpečujú invariantnosť výsledných príznakov voči rotácií a patria medzi najpoužívanejších zástupcov triedy lokálnych príznakov. Pre lokálne príznaky som sa rozhodol, pretože sú oproti globálnym príznakom menej citlivé na zmenu okolia a oklúziu.

Pre výber kľúčových oblastí som sa rozhodol vyskúšať dve metódy. Prvou metódou výberu kľúčových oblastí zo snímky bol hustý výber kľúčových oblastí prostredníctvom homogénnej mriežky a druhou metódou určovania kľúčových oblastí v snímke bolo použitie Harris-Laplaceovho detektora kľúčových oblastí.

Pre popis kľúčových oblastí boli použité dve varianty výpočtu SIFT deskriptorov na základe toho, či bola snímka v stupni šedi alebo bola farebná. Pre kľúčovú oblasť snímky v stupni šedi sa SIFT deskriptor spočítal pre zadnú veľkosť oblasti a výsledkom bol 128-rozmerný vektor reprezentujúci SIFT deskriptor. Pre kľúčovú oblasť farebnej snímky sa najprv spočítali SIFT deskriptory pre každú farebnú zložku RGB zvlášť a výsledné SIFT deskriptory sa skonkatenovali do jedného vektoru. Výsledný vektor príznakov sa potom získal PCA redukciou tohto vektora SIFT deskriptorov na 128-rozmerný vektor.

Kombináciou dvoch spôsobov určovania kľúčových oblastí snímky a dvoch spôsobov výpočtu SIFT deskriptorov sú získané štyri varianty výpočtu príznakov, ktoré boli pri experimentoch testované.

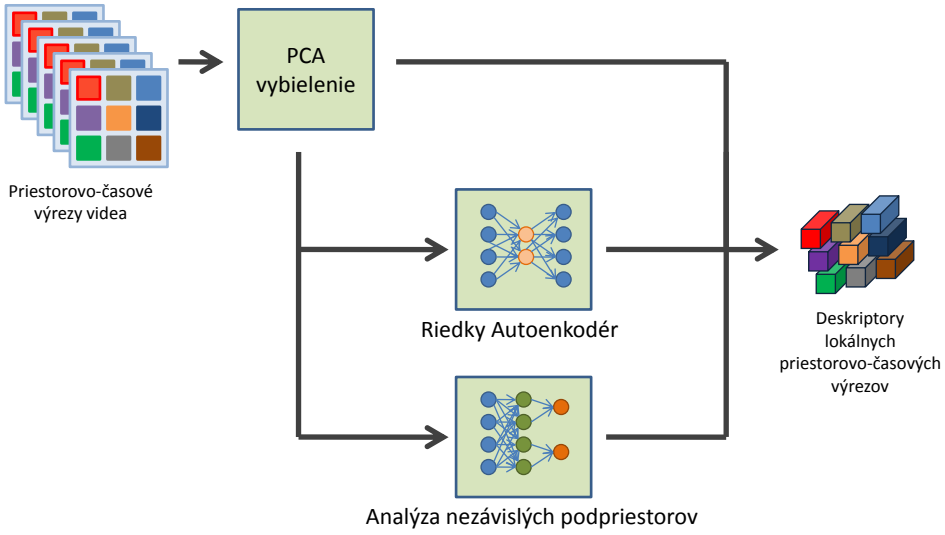


Obrázek 4.2: Popis princípu extrakcie SIFT deskriptorov pre celé video.

4.2 Popis priestorovo-časových výrezov videa metódami učenia príznakov

Táto sekcia popisuje postup pri získavaní príznakov z vizuálneho kanála videa prostredníctvom vybratých metód učenia príznakov. Ako už bolo v úvode tejto kapitoly uvedené, tak mojim cieľom bolo použiť také príznaky z videa, ktoré by reprezentovali vstupné video prostredníctvom BOW rovnako dobre nezávisle na úlohe klasifikácie videa. Kapitola 3 však uvádza, že sa ukazuje, že neexistujú univerzálne štandardne príznaky videa, ktoré by rovnako dobre fungovali pre rôzne dátové sady a rôzne typy úloh klasifikácie videa. Práve preto som sa rozhodol vyskúšať pre extrakciu príznakov z videa metódy učenia príznakov, pri ktorých sa výsledné príznaky prostredníctvom učenia dokážu prispôsobiť príslušnej sade videí. Mojim cieľom teda bolo využiť metódy učenia príznakov tak, aby nová získaná reprezentácia čo najlepšie reprezentovala vstupné dáta videa bez ohľadu na riešenú klasifikačnú úlohu. Pre dosiahnutie tohto cieľa som vyskúšal použiť tri metódy učenia príznakov, ktorými sú PCA vybielenie, ISA a Riedke Autoenkodére. Zisk deskriptorov prostredníctvom týchto metód zobrazuje obrázok 4.3.

Ako možno vidieť na tomto obrázku, tak vstupom použitých metód učenia príznakov boli malé priestorovo-časové (3D) výrezy videa. Popis 3D výrezov videa som zvolil, pretože na rozdiel od príznakov snímok sú výsledné príznaky schopné pri 3D výrezoch videa zachytiť pohyb hrán, objektov a podobne. Pre výber týchto výrezov som použil hustý výber výrezov pomocou 3D homogénnej mriežky. Hustý výber výrezov som zvolil preto, lebo ako je uvedené v sekcií 2.2, tak pri klasifikácií obsahu reálneho videa dosahuje tento spôsob dokonca lepšiu úspešnosť než pri použití detektorov 3D oblastí. Výsledkom je súbor 3D oblastí, na ktoré sa aplikujú metódy učenia príznakov. Predtým však ako bolo možné metódy učenia príznakov použiť pre popis 3D výrezov, bolo nutné tieto metódy natrénovať pre zvolenú trénovaciu sadu 3D výrezov. Pre tento účel som použil z každého videa trénovacej sady rovnaký počet náhodné lokalizovaných 3D výrezov videa.



Obrázek 4.3: Postup získania deskriptorov priestorovo-časových výrezov videa vybratými metódami učenia príznakov.

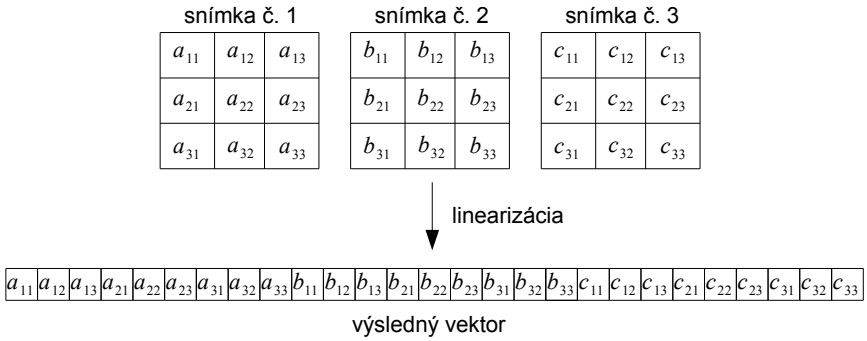
Keďže vstupmi PCA vybielenia, ISA a aj Riedkych Autoenkodérov sú vektory, tak jednotlivé 3D výrezy boli transformované na vektory (linearizované). Linearizácia 3D výrezu 3×3 pixelov troch snímok (v stupni šedi) je ukázaná na obrázku 4.4. Ako možno na obrázku vidieť, tak linearizácia 3D výrezu (pre snímky videa v stupni šedi) pozostáva z naskladania riadkov všetkých snímkov za seba do jedného vektora. Pri farebných snímkach boli prietorovo-časové výrezy linearizované po farbených zložkách RGB a skonkatenované do jedného vektora.

Prvú metódu učenia príznakov, ktorú som pre popis 3D výrezov použil bola Analýza hlavných komponentov (PCA), presnejšie PCA vybielenie vstupných dát. Túto metódu som použil ako referenčnú metódu učenia príznakov k Riedkym Autoenkodérom a ISA. Prvým krokom pri tejto metóde je normalizácia intenzity linearizovaných vstupných 3D výrezov. Tento krok bol vykonaný odčítaním priemernej hodnoty pixelov 3D výrezu od každého pixlu a to zvlášť pre každý linearizovaný 3D výrez. Pre trénovaciu sadu normalizovaných výrezov bola spočítaná kovariančná matica a z nej získané vlastné vektory. Tieto vlastné vektory sú potom ďalej použité pri transformácií normalizovaných výrezov videa na nové reprezentácie. Okrem prenásobenia normalizovaných výrezov boli výrezy aj vybielené. Vybielenie zabezpečuje aby sa vstupné výrezy dekorelovali. Vybielenie bolo vykonané predelením príznakov získaných PCA transformáciou odmocninami príslušných vlastných čísiel, teda

$$x_{PCAbiel,i} = \frac{x_{PCArot,i}}{\sqrt{\lambda_i + \epsilon}}, \tag{4.5}$$

kde ϵ bola malá konštanta (napr. 0,1), ktorá zabezpečovala aby výsledný vektor príznakov neobsahoval príliš veľké hodnoty, ktoré mohli vzniknúť pri veľmi malých hodnotách vlastných čísiel. Výsledkom tejto metódy bola nová dekorelovaná reprezentácia pre linearizovaný výrez s rovnakou alebo redukovanou dimenzionalitou v závislosti na počte použitých vlastných vektorov pri transformácií.

Druhou metódou učenia príznakov, ktorú som použil pre popis 3D výrezov videa bola ISA. Túto metódu som vyskúšal použiť pre jej schopnosť produkovať invariantné príznaky



Obrázek 4.4: Linearizáca priestorovo-časových výrezov videa pre snímky v stupni šedi.

vďaka združovaniu podobných filtrov do jedného podpriestoru a pretože pri použití v hierarchických príznakoch dosahovala pri klasifikácií videa veľmi dobré výsledky [20]. V tejto práci som sa obmedzil na použitie jednej ISA siete, ktorá bola použitá pre transformáciu všetkých výrezov. Pri trénovaní ISA bol použitý postup popísaný v sekcii 3.2, ktorý využíva pre natrénovanie ISA siete projekčný gradientný zostup. Váhy teda boli v každej trénovacej iterácií najprv upravené pomocou gradientu objektívnej funkcie a následne ortogonalizované. Keďže tento algoritmus je pomalý pri veľkej dimenzionalite vstupných dát, tak som použil ako vstup ISA siete PCA vybielené 3D výrezy (pri ich získaní bol použitý rovnaký postup popísaný vyššie). Novú reprezentáciu vstupných PCA vybielených výrezov som teda pomocou ISA získaval ako vektor výstupov neurónov výstupnej vrstvy.

Z definície ISA je zrejme, že nová reprezentácia vstupných dát obsahuje menší alebo maximálne rovnaký počet dimenzií ako vstup. Preto som ako tretiu metódu učenia príznakov použil Riedke Autoenkodére, ktoré umožňujú vytvoriť novú reprezentáciu s dimenzionalitou väčšou než má vstup. Podobnú vlastnosť poskytuje aj Riedky Obmedzený Boltzmanov stroj, ale pre Riedke Autoenkodére som sa rozhodol pre ich jednoduchosť. Ako už bolo uvedené v sekcií 3.1, tak nevýhodou Riedkych Autoenkodérov je, že vstupné hodnoty musia byť v rozsahu 0 až 1, pretože aktivačná funkcia neurónov výstupnej vrstvy je sigmoida. Použil som teda Riedky Autonkodér s lineárnou aktivačnou funkciou vo výstupnej vrstve, nazývaný tiež Lineárny Kodér. Mohol som tak znížiť dimenzionalitu vstupných linearizovaných výrezov prostredníctvom PCA vybielenia a zrýchliť tak trénovanie pomocou gradientneho zostupu. Tým sa zmenil výstup siete na výstup bázovej funkcie výstupnej vrstvy. Výpočet parciálnych derivácií objektívnej funkcie pomocou algoritmu Backpropagation zodpovedal až na jeden vzťah presne popisu v sekcií 3.1. Jediným zmeneným vzťahom bol vzťah 3.7 pre výpočet $\delta_i^{(3)}$, ktorý sa zmenil na

$$\delta_i^{(3)} = -(y_i - a_i^{(3)}). \tag{4.6}$$

Pomocou natrénovaného Riedkeho Autoenkodéru (s lineárnou aktivačnou funkciou vo výstupnej vrstve) som novú reprezentáciu vstupného PCA vybieleného výrezu videa získal ako vektor aktivácií neurónov skrytej vrstvy.

Kapitola 5

Implementácia častí použitého systému pre klasifikáciu videa pomocou Bag of Words

V predchádzajúcej kapitole boli popísané metódy extrakcie príznakov, ktoré som použil spolu so systémom využívajúcim BOW pri klasifikácií videa. V tejto kapitole sa nachádzajú informácie o tom, aké existujúce nástroje a techniky implementácie boli pre jednotlivé metódy extrakcie a časti systému použité.

Extrakciu príznakov som sa rozhodol rozdeliť na dva nezávisle pracujúce celky. V skripte vykonávajúcom extrakciu SIFT deskriptorov z kľúčových oblastí vybratých snímok videa som použil dva existujúce nástroje. Prvým z týchto nástrojov bol nástroj `ffmpeg`, ktorý som použil pre rozloženie videa na snímky pomocou prikazu `ffmpeg -i <nazov_videa> <nazov_videa>%d.jpg`. Z týchto snímok som následne vybral reprezentatívne snímky v pravidelných intervaloch. Pre určenie kľúčových oblastí v snímkach a výpočet SIFT deskriptorov z kľúčových oblastí som použil nástroj vyvinutý na UPGM FIT VUT, pre účely sémantického indexovania videa a rozpoznávania žánru videa, ktorý sa používa primárne pre účely súťaží PASCAL, TRECVID [5] a MediaEval [15]. Výsledné SIFT deskriptory snímok som zlúčil do jedného súboru, ktorý obsahoval SIFT deskriptory pre celé video.

Pre popis 3D výrezov som vytvoril sadu funkcií v jazyku MATLAB. Bližší popis tejto implementácie sa nachádza v sekcií 5.1. Výstup je však rovnaký ako pri extrakcií SIFT deskriptorov. Pre vstupné video je výsledkom súbor obsahujúci príznakové vektory reprezentujúce celé video.

Pre vytvorenie BOW reprezentácie videa som použil sadu nástrojov vyvinutú na UPGM FIT VUT, ktoré sú podobne ako nástroj pre extrakciu SIFT deskriptorov z obrazu vyvinuté pre účely sémantického indexovania videa a rozpoznávania žánru videa a primárne sú používané pre účely súťaží PASCAL, TRECVID [5] a MediaEval [15]. Tieto nástroje umožňujú zo súboru vektorov príznakov vytvoriť BOW reprezentáciu postupom popísaným v kapitole 4. Pre vytvorenie slovníka kódových slov som trénoval K-means pre reprezentatívny počet náhodne zvolených vektorov príznakov vybratých zo súborov príznakových vektorov celej trénovacej sady videí. Slovník bol vytvorený pre každú metódu extrakcie príznakov zvlášť a nový slovník som vytváral aj pri zmene konfigurácie nejakej metódy extrakcie príznakov.

Bližší popis implementácie klasifikátoru Support Vector Machine a popis implementácie metódy Multiple Kernel Learning sa nachádza v sekcií 5.2.

Všetky tieto časti, ktoré sú tu popísané sú schopné pracovať samostatne. Pri svojich

experimentoch som ich podľa potreby prepojil pomocou dočasných súborov obsahujúcich príznakové vektory a pomocných skriptov, ktoré som púšťal prostredníctvom Sun Grid Engine (SGE) na výpočetnom gride FIT VUT [1]. Výpočetný grid FIT VUT som použil z dôvodu vykonávania množstva experimentov, ktoré boli pre použité dátové sady výpočetne náročné a použitie jedného počítača sa ukázalo ako nedostačujúce.

5.1 Implementácia popisu priestorovo-časových výrezov videa učením príznakov

Popis 3D výrezov videa zvolenými metódami učenia príznakov som sa rozhodol implementovať v jazyku MATLAB. MATLAB som použil preto, lebo výpočetné kroky popísané v sekcií 4.2 pri trénovaní použitých metód učenia príznakov, ako aj získavaní nových reprezentácií možno vektorizovať. To umožní násobiť vstupný vektor pomocou viacerých váhových vektorov súčasne, či počítať súčasne výstupy pre viacero vstupných vektorov. Výpočty sa tým menia na operácie nad maticami a práve preto som sa rozhodol použiť MATLAB, ktorý maticové operácie vykonáva rýchlo a efektívne.

Pre trénovanie a získanie novej reprezentácie videa pomocou zvolenej metódy učenia príznakov som v MATLABe vytvoril sadu funkcií. Tieto funkcie sú zastrešované funkciou `trainFLMethod` určenou pre trénovanie zvolenej metódy učenia príznakov a funkciou `newRepresByFLM` určenou pre získanie novej reprezentácie vstupného videa. Na základe vstupného zoznamu videí určených pre trénovanie a výberu metódy učenia príznakov funkcia `trainFLMethod` vyberie náhodne z každého videa rovnaký počet priestorovo-časových výrezov a pre získanú sadu trénovacích vzorkov zavolá príslušnú funkciu pre natrénovanie vybratej metódy učenia príznakov. Výsledné lokalizované filtre spolu s ďalšími informáciami sú uložené do špeciálneho konfiguračného súboru. Cesta k tomuto konfiguračnému súboru sa potom použije ako parameter funkcie `newRepresByFLM`, ktorá zo zadaného video vyberie 3D homogénnou mriežkou 3D výrezy. Zo získanej matice linearizovaných výrezov sa potom spočítajú pomocou konfiguračného súboru príznakové vektory (nové reprezentácie výrezov) pre celé video. Výsledkom je súbor vektorov príznakov, ktorý reprezentuje vstupné video.

V nasledujúcom texte sa nachádza niekoľko informácií o tom, na aké problémy som pri týchto jednotlivých fázach implementácie narazil a aké techniky, prípadne nástroje som pri ich vyriešení použil.

Prvou úlohou, ktorú som riešil bolo rýchle načítanie snímok videa do matice snímok. Pre túto úlohu volám v príslušnej funkcií nástroj `ffmpeg`, pomocou ktorého je video rozložené na snímky a tie sú dočasne uložené do zadaného adresára. Tieto snímky sú následne z tohto adresára načítavané ako obrázky (podľa príslušného parametra sú ďalej prevedené do stupňov šedi) a ukladané do matice snímok.

Ďalším problémom sa pri trénovaní a vytváraní novej reprezentácie vstupného videa ukázal byť výber 3D oblastí z matice snímok a ich linearizácia do vektorov. Počiatočná implementácia v MATLABe, ktorú som vytvoril bola príliš pomalá, čo sa pre dlhé videá a veľké dátové sady ukázalo ako problém. Preto som sa rozhodol výber 3D výrezov videa a ich linearizovanie do vektorov implementovať v jazyku C pomocou MEX-súborov, ktoré po kompilácií možno používať ako funkcie MATLABu. Tento krok výpočet značne zrýchlil a preto som v jazyku C implementoval nie len výber 3D výrezov pomocou homogénnej mriežky (používaný pri popise videa), ale aj náhodný výber výrezov z videí používaný pri trénovaní. Vstupom vytvorených funkcií je matica obsahujúca snímky videa a výstupom je matica obsahujúca linearizované 3D výrezy, ktoré zodpovedajú zadaným pa-

rametrom (veľkosť výrezov, veľkosť homogénnej mriežky, počet náhodne vybratých výrezov a podobne).

Pre učenie jednotlivých metód učenia príznakov bol použitý postup popísaný v sekciach 4.2 a v kapitole 3. Pri niektorých metódach som pre proces trénovania využil aj voľne dostupné časti kódu vykonávajúce popísaný postup efektívnejšie. Konkrétne pri ISA som využil voľne dostupnú trénovaciu funkciu v MATLABe, ktorá bola súčasťou implementácie pre extrakciu hierarchických príznakov videa [20]. Pre účely svojej práce som túto funkciu upravil a použil vo svojej implementácií.

Pre získanie vlastných vektorov z kovariančnej matice pre metódu PCA vybielenia dát som využil vstavanú funkciu MATLABu pre singulárny rozklad. Pomocou tejto funkcie som z kovariančnej matice trénovacích vektorov získal maticu usporiadaných vlastných vektorov a príslušný vektor s vlastnými hodnotami, ktoré som pri PCA vybielení ďalej používal.

Implementáciu výpočtu parciálnych derivácií pre trénovanie Riedkych Autoenkodérov s lineárnou aktivačnou funkciou vo výstupnej vrstve som vytvoril podľa popisu v sekcií 3.1 a 4.2. Korektnosť implementácie som overil na základe kontroly produkovaného gradientu [29], pretože z definície gradientu vyplýva, že musí platiť vzťah

$$g(\theta) \approx \frac{J(\theta + \epsilon) - J(\theta - \epsilon)}{2\epsilon}, \tag{5.1}$$

kde $g(\theta) = \frac{d}{d\theta} J(\theta)$ a ϵ je malá hodnota blízka nule (napr. 10^{-4}). V mojom prípade bol výsledkom funkcie g vektor obsahujúci parciálne derivácie váh a biasov získaných pomocou Backpropagation a funkcia J bola objektívna funkcia Riedkeho Autoenkodéru s lineárnou aktivačnou funkciou vo výstupnej vrstve. Hodnota θ bol tiež vektor a konkrétne vektor získaný konkatenáciou váh a biasov. To znamená, že rovnica 5.1 potom vyzerala ako

$$g(\theta)_i \approx \frac{J(\theta + e_i\epsilon) - J(\theta - e_i\epsilon)}{2\epsilon}, \tag{5.2}$$

kde e_i bol vektor obsahujúci na i-tej pozícií hodnotu 1 a na ostatných pozíciách 0.

Gradientny zostup popísaný v sekcií 3.1 však pri trénovaní na vybratých dátových sadách videí veľmi pomaly konvergoval. Z toho dôvodu som sa rozhodol podobne ako to navrhuje Ng vo svojej práci [29] využiť sofistikovajší spôsob minimalizácie objektívnej funkcie pomocou gradientneho zostupu, ktorým môže byť napríklad metóda optimalizujúca hodnotu α alebo stochastický gradientny zostup. Práca, ktorú vytvorili Ng et al. [21] ukazuje experimenty s rôznymi metódami minimalizácie objektívnej funkcie pomocou gradientneho zostupu a práve aj pre Riedke Autoenkdére. Z výsledkov, ktoré táto práca ukazuje sú najlepšie výsledky dosiahnuté pri použití Limited memory Broyden-Fletcher-Goldfarb-Shanno algoritmu (L-BFGS), ktorý prispôsobuje aktualizácie s cieľom aproximovať Hessovu maticu. Z toho dôvodu som pre urýchlenie procesu učenia Riedkych Autoenkodérov použil voľne dostupnú implementáciu algoritmu L-BFGS v MATLABe [2].

5.2 Support Vector Machine a Multiple Kernel Learning

Táto sekcia uvádza bližšie informácie o vytvorenej implementácií a existujúcich nástrojoch, ktoré používam v procese klasifikácie BOW reprezentácií videí pomocou klasifikátora Support Vector Machine. V závere tejto sekcie sa nachádza popis implementácie navrhnutej metódy Muliple Kernel Learning, ktorá je popísaná v sekcií 4.2.

Pôvodne som vo svojich experimentoch pri klasifikácií využíval implementáciu SVM v programe RapidMiner [3]. Avšak pre použité dátové sady videí boli experimenty extrémne

pomalé. Bolo to spôsobené tým, že pri hľadaní optimálnej hodnoty SVM regulačného parametru C a škalovacieho parametru jadra γ pomocou grid search a 6-stupňovej crossvalidácie sa jadro pre SVM počítalo pre každú kombináciu parametrov znova. V RapidMineri však zatiaľ efektívne riešenie tohto problému neexistuje. Preto som sa rozhodol využiť implementáciu SVM vo voľne dostupnej knižnici LIBSVM [8], ktorá mi umožnila tento problém odstrániť pomocou predpočítania časti jadra, pretože funkcie trénovania modelu SVM a predikcie umožňujú vložiť predpočítané jadro SVM ako parameter funkcie.

Funkcie tejto knižnice sú dostupné v rozhraniach viacerých programovacích jazykov. Vo svojej implementácií som využil funkcie knižnice LIBSVM s rozhraním vytvoreným pre MATLAB. Pre MATLAB som sa rozhodol preto, lebo predpočítavanie jadra možno vykonať pomocou maticových operácií, ktoré MATLAB vykonáva efektívne a rýchlo.

Výsledná implementácia pracujúca s týmito funkciami pri procese klasifikácie vstupného súboru BOW reprezentácií sa nachádza vo funkcií `svmClassification`. Na začiatku tejto funkcie je pre celú dátovú sadu predpočítaná časť jadra 4.3, ktorou je

$$K_{part}(H, H') = \sum_{n=1}^{V} \frac{(h_n - h'_n)^2}{h_n + h'_n}. \tag{5.3}$$

Práve táto časť sa nemení počas zmeny škálovacieho parametru jadra γ a jej výpočet trvá najdlhšie. Keďže sa nemení, stačí ju vypočítať iba raz a pri hľadaní parametrov γ a C sa potom jadro dopočíta pre daný parameter γ do kompletnej podoby. Takýmto spôsobom som značne zrýchlil výslednú klasifikáciu pomocou SVM, ktorá predtým trvala aj niekoľko dní a po tejto úprave maximálne niekoľko hodín.

Pre účely navrhnutej metódy Multiple Kernel Learning ukladám vo funkcií `svmClassification` okrem úspešnosti klasifikácie aj predpočítané jadro (podľa vzťahu 4.3) pre celú sadu (trénovacia + testovacia sada) a nájdený parameter γ. Práve pre navrhnutú metódu Multiple Kernel Learning som vytvoril funkciu `svmMKL` v MATLABe, ktorá pre vstupný zoznam takto získaných jadier spočíta nové jadro podľa vzťahu 4.4. Toto jadro sa následné delí podľa anotácií na trénovaciu časť a testovaciu časť. Príslušné časti sú potom použité pre natrénovanie modelu SVM a testovanie úspešností klasifikácie tohto modelu.

Kapitola 6

Použité dátové sady

V tejto kapitole sa nachádza popis dátových sád, na ktorých boli vykonané experimenty výsledným systémom pre klasifikáciu videa. Cieľom bolo vybrať také dátové sady, ktorých úlohy klasifikácie neboli úplne rovnaké a obsahovali reálne videá s rôznymi vlastnosťami tak, aby tieto datasety neboli príliš triviálne.

Okrem týchto kritérií som pri výbere dátových sád zohľadnil aj ďalšie dve veľmi dôležité kritéria a to dostupnosť dátovej sady a existenciu publikovaných výsledkov pre vybratú dátovú sadu. Dostupnosťou dátovej sady sa myslí možnosť voľného stiahnutia dátovej sady z Webu a existenciou publikovaných výsledkov pre dátovú sadu sa myslí existencia a dostupnosť výsledkov experimentov iných prístupov pre klasifikáciu videa na danej dátovej sade.

Na základe vyššie uvedených kritérií som vybral dve dátové sady videí, ktorými sú dátová sada A Large Video Database for Human Motion Recognition (HMDB) [19] a YouTube Action Data Set (UCF11) [24]. HMDB je dataset určený pre klasifikáciu ľudskej činnosti a bližší popis tohto datasetu sa nachádza v sekcií 6.1. UCF11 je datasetom určeným pre klasifikáciu akcie vo videu a bližší popis tohto datasetu sa nachádza v sekcií 6.2.

6.1 A Large Video Database for Human Motion Recognition

A Large Video Database for Human Motion Recognition[1] (HMDB) je dataset videí určený pre klasifikáciu ľudských činností. Táto sada obsahuje videá každodennej ľudskej činnosti rozdelené do 51 kategórií. Každé video je rozdelené do klipov tak, aby každý klip obsahoval jednu konkrétnu ľudskú činnosť po celú dĺžku trvania klipu. Ukážka snímok videí tejto dátovej sady sa nachádza na obrázku 6.1. Pri tvorbe tohto datasetu boli použité digitalizované filmy, verejné databáze ako je Prelinger archív, videá dostupné na internete, YouTube a Google videá. HMDB teda obsahuje videá, ktoré vznikli za rôznych podmienok, rôznymi autormi a neboli vytvorené jednou kamerou. To znamená, že videá obsahujú rôzne pozadia, rôzne typy osvetlenia, rôzne pohyby kamery a podobne. Minimálna dĺžka videa je 1 sekunda. Všetky klipy videí boli pomocou knižnice `ffmpeg` prevedené na klipy s 30 fps a skomprimované použitím kodeku DivX 5.0. Všetky klipy sú vo formáte AVI.

Kategorie, do ktorých sú klipy videí roztriedené možno rozdeliť na 5 typov:

- **Všeobecné pohyby tváre:** úsmev, smiech, žuvanie, rozprávanie

[1] Dataset HMDB je dostupný na: http://serre-lab.clps.brown.edu/resources/HMDB/

Obrázek 6.1: Ukážka snímok videí dátovej sady A Large Video Database for Human Motion Recognition

- **Akcie tváre pri manipulácií s objektom:** fajčenie, jedenie, pitie
- **Všeobecné pohyby tela:** mlynské koleso, tlieskanie rukami, lezenie, kráčanie po schodoch, padnutie na zem, prudký pohyb chrbtom ruky, stojka, skok, zastavenie, tlačenie, beh, sadnutie, sadnutie (z ľahu), kotrmelec, postavenie sa, otočenie, kráčanie, mávanie
- **Pohyb tela pre interakciu s objektom:** česanie vlasov, chytanie, tasenie meča, driblovanie, golf, udrenie niečoho, kopnutie lopty, vybrať niečo, vyliať niečo, tlačiť niečo, jazdenie na bicykli, jazdenie na koni, odpálenie loptičky, vystrelenie šípu, vystrelenie so zbraňou, odpaľovanie baseballovou pálkou, cvičenie s mečom, hádzanie
- **Pohyb tela pre interakciu s inými ľuďmi:** šermovanie, objímanie, kopnutie niekoho, bozkávanie, udretie niekoho, podávanie rúk, bitka mečmi

Pre HMDB existujú tri rozdelenia klipov videí do trénovacej a testovacej sady. Pre všetky tri tieto rozdelenia platí, že klipy rovnakého videa sú buď v trénovacej alebo v testovacej sade. Nenachádza sa teda súčasne časť klipov rovnakého videa v trénovacej množine a časť v testovacej sade. Pre všetky tri rozdelenia tiež platí, že pre každú činnosť sa v trénovacej množine nachádza 70 klipov a v testovacej množine 30 klipov. To znamená, že pre všetky tri rozdelenia obsahuje trénovacia množina spolu 3 570 klipov a testovacia množina 1 530 klipov.

6.2 YouTube Action Data Set

YouTube Action Data Set[2] je dataset určený pre klasifikáciu akcie vo videu. Tento dataset je vytvorený z YouTube videí, ktoré Liu et al. [24] doplnili vlastnými videami. Pre UCF11 podobne ako pre HMDB platí, že obsahuje videá, ktoré majú rôzny pohyb kamery, rôzne

[2]Dataset UFC11 je dostupný na: http://crcv.ucf.edu/data/UCF_YouTube_Action.php

Obrázek 6.2: Ukážka snímok videí dátovej sady YouTube Action Data Set

pozadie, rôzne zmeny veľkosti objektov, rôzne body pohľadu, rôzne osvetlenie a nízke rozlíšenie. Všetky videá sú vo formáte mpeg4 a všetky majú 29.97 fps. Ukážka snímok videí tejto dátovej sady sa nachádza na obrázku 6.2.

Videá UFC11 sú rozdelené do 11 akčných kategórií, ktorými sú: strieľanie na kôš v basketbale, smečovanie vo volejbale, skákanie na trampolíne, žonglovanie s loptou vo futbale, jazdenie na koni, bicyklovanie, potápanie, plávanie, zakončovanie v golfe, odrazenie loptičky v tenise a prechádzanie sa (so psom).

Okrem toho sú videá v každej kategórií rozdelené do 25 podobnostných skupín, kde každá skupina obsahuje minimálne 6 videí, ktoré sú navzájom závislé či už pozadím, rovnakým hercom alebo podobne. Videá medzi týmito skupinami sú však nezávisle a navzájom sa líšia.

UFC11 obsahuje celkovo 1 168 videí, kde každé video obsahuje jednu konkrétnu činnosť počas celej dĺžky trvania videa.

Pre rozdelenie na trénovaciu a testovaciu sadu používajú autori tohto datasetu 5-stupňovú crossvalidáciu.

Kapitola 7

Dosiahnuté výsledky experimentov

V tejto kapitole sa nachádza popis výsledkov experimentov, ktoré som dosiahol implementovaným systémom pre klasifikáciu videa pomocou BOW popísaného v kapitole 4. Cieľom experimentov bolo otestovať úspešnosť klasifikácie videa pri použití jednotlivých príznakov a výsledky medzi porovnať. Okrem toho sú tu uvedené aj výsledky experimentov dosiahnuté navrhnutou metódou Multiple Kernel Learning. Všetky tieto experimenty som vykonával pre dátové sady videí HMDB a UCF11, ktoré sú popísané v kapitole 6.

Keďže pre dátovú sadu HMDB existujú tri rozdelenia klipov na trénovaciu a testovaciu sadu, vykonával som experimenty pre všetky tri tieto rozdelenia a výsledná úspešnosť klasifikácie bola potom získaná spriemerovaním týchto troch výsledkov. Ako referenčné výsledky klasifikácie videa na tejto dátovej sade som použil výsledky klasifikácie dosiahnuté autormi dátovej sady HMDB [19]. Tieto výsledky reprezentuje tabuľka 7.1. Ako možno v tabuľke vidieť, tak Kuehne et al. použili pri klasifikácií dva prístupy. Prvý z nich bol založený na použití lokálnych príznakov HOG/HOF popisujúcich kľúčové 3D oblastí videa, ktoré boli detekované pomocou Harrisovho detektora 3D oblastí. Z týchto príznakov ďalej Kuehne et al. vytvorili BOW reprezentáciu celého videa a pre klasifikáciu použili Support Vector Machine. Druhý prístup využíva C2 príznaky popisujúce vizuálny kortex videa.

Ako už bolo uvedené v sekcií 6.2, tak autori dátovej sady UCF11 používajú pre rozdelenie tejto sady na trénovaciu a testovaciu časť 5-stupňovú crossvalidáciu. Pri tejto voľbe rozdelenia sady by však bolo nutné trénovať metódy učenia príznakov vždy 5-krát pre rôzne kombinácie častí sady v crossvalidácií. Preto som sa rozhodol tento postup zjednodušiť. UCF11 som rozdelil na 5 častí, z ktorých som náhodne vybral 4 časti a tie používal ako trénovaciu dátovú sadu. Zvyšnú časť som potom používal ako testovaciu sadu. Vykonané experimenty tak nezodpovedajú popisu experimentov autormi datasetu a teda porovnanie mojich výsledkov s výsledkami iných prác beriem len ako orientačné. Výsledky klasifikácie dosiahnuté inými prácami pre UCF11 dátovú sadu reprezentuje tabuľka 7.2. Prvé dva prístupy v tejto tabuľke sú výsledky dosiahnuté autormi dátovej sady UCF11 [24] pomocou popisu kľúčových oblastí SIFT deskriptormi a popisu 3D výrezov pomocou 3D kuboidov. Tretí prístup používa pre popis kľúčových 3D oblastí príznaky HOG/HOF. Tento prístup zodpovedá popisu vyššie pri dátovej sade HMDB.

Pri testovaní úspešností klasifikácie použitých príznakov som systém, s ktorým boli príznaky testované nastavil vždy rovnako. Medzi tieto nastavenia systému patrí aj veľkosť slovníka kódových slov, ktorú som pre všetky typy príznakov nastavil na 4096 slov. Jednotné nastavenie systému umožnilo porovnať úspešnosti klasifikácie jednotlivých príznakov medzi sebou.

Pri metódach učenia príznakov som používal rovnako veľké 3D výrezy videa ($16 \times$

Použitý prístup	Úspešnosť klasifikácie
HOG/HOF + BOW + SVM [19]	20.44 %
C2 + SVM [19]	22.83 %

Tabulka 7.1: Výsledky experimentov, ktoré vykonali autori dátovej sady HMDB na dátovej sade HMDB. Prvý prístup používa rovnakú štruktúru systému pre klasifikáciu videa ako bola použitá v tejto práci, ale pre popis priestorovo-časových výrezov sú použité príznaky HOG/HOF.

Použitý prístup	Úspešnosť klasifikácie
SIFT + BOW + Adaboost [24]	63.0 %
3D Kuboid + BOW + SVM [24]	65.4 %
HOG/HOF + BOW + SVM [19]	58.9 %

Tabulka 7.2: Výsledky experimentov, ktoré vykonali autori dátovej sady HMDB a UCF11 na dátovej sade UCF11. Druhý a tretí prístup používa príznaky popisujúce priestorovo-časové výrezy videa. Prvý používa príznaky založené na snímkach.

16 pixlov × 11 snímiek), ktoré som z videa vyberal jednotnou 3D homogénnou mriežkou (vzdialenosť stredov oblastí v x-ovej a y-ovej osi snímok bola 6 pixlov a v časovej osi 10 snímok). Takto potom bolo možné porovnať, ktorá metóda učenia príznakov produkuje príznaky s najlepšou úspešnosťou klasifikácie videa.

Výsledky experimentov pre SIFT deskripotory kľúčových oblastí reprezentatívnych snímok popisuje pre obidve dátové sady sekcia 7.1. Výsledky experimentov pre naučené príznaky 3D výrezov videa pomocou PCA vybielenia obsahuje pre obidve dátové sady sekcia 7.2, pre ISA sekcia 7.3 a pre Riedke Autoenkodére sekcia 7.4. Porovnanie dosiahnutých výsledkov použitých príznakov, ako aj porovnanie výsledkov s publikovanými výsledkami obsahuje sekcia 7.5.

Popis dosiahnutých úspešností klasifikácie pri kombinácií predpočítaných jadier pomocou navrhnutej metódy Multiple Kernel learning sa nachádza v sekcií 7.6.

7.1 Výsledky pre SIFT deskriptory

V tejto sekcií sa nachádzajú uvedené výsledky experimentov, ktoré som dosiahol pre referenčné príznaky, ktorými sú SIFT deskriptory. Ako už bolo uvedené v sekcií 4.1, tak pri týchto príznakoch som sa rozhodol vyskúšať 4 varianty extrakcie SIFT deskriptorov z videa a to na základe použitých SIFT deskripotorov a určovania kľúčových oblastí v snímke. Pre všetky varianty boli vo videách vyberané rovnaké reprezentatívne snímky (každý 5 snímok videa). Pri experimentoch s výberom kľúčových oblastí prostredníctvom homogénnej mriežky bola použitá mriežka so vzdialenosťou stredov oblastí 8 pixlov v obidvoch osiach snímky.

Najlepšie výsledky experimentov (spriemerované výsledky troch výsledkov získaných z troch rozdelení tejto sady na trénovaciu a testovaciu časť) pre dátovú sadu HMDB reprezentuje tabuľka 7.3a. DENSE označuje použitie homogénnej mriežky pri určovaní kľúčových oblastí a číslo nasledovné za DENSE označuje polomer kľúčovej oblasti, z ktorej sa SIFT deskriptor počítal. HARLAP označuje použitie Harris-Laplaceoho detektora kľúčových oblastí. SIFT označuje použitie SIFT deskriptorov počítaných zo snímok v stupňoch šedi

Extrakcia príznakov zo snímok	Úspešnosť klasifikácie
DENSE16 + CSIFT	19.17 %
DENSE16 + SIFT	17.89 %
DENSE8 + CSIFT	18.52 %
DENSE8 + SIFT	16.8 %
HARLAP + CSIFT	**19.26 %**
HARLAP + SIFT	17.71 %

(a) Výsledky pre HMDB

Extrakcia príznakov zo snímok	Úspešnosť klasifikácie
DENSE16 + CSIFT	**73.92 %**
DENSE16 + SIFT	72.21 %
DENSE8 + CSIFT	69.18 %
DENSE8 + SIFT	63.44 %
HARLAP + CSIFT	73.11 %
HARLAP + SIFT	69.79 %

(b) Výsledky pre UCF11

Tabulka 7.3: Výsledky klasifikácie videa pre klasifikačný systém pomocou Bag of Words, ktorý používal pre popis videa SIFT deskriptory. DENSE označuje použitie homogénnej mriežky pre určovanie kľúčových oblastí a HARLAP označuje použitie Harris-Laplaceovho detektora. SIFT označuje použitie extrakcie SIFT deskriptorov zo snímok v stupňoch šedi a CSIFT označuje použitie extrakcie SIFT deskripotov z farebných snímok.

a CSIF označuje použitie SIFT deskriptorov počítaných z farebných snímiek.

Ako je teda v tabuľke 7.3a vidieť, tak pre dátovú sadu HMDB dosiahla najlepšiu priemernú úspešnosť klasifikácie kombinácia Harris-Laplaceovho detektora kľúčových oblastí a SIFT deskriptorov farebných snímok. Táto úspešnosť klasifikácie bola 19.26 %.

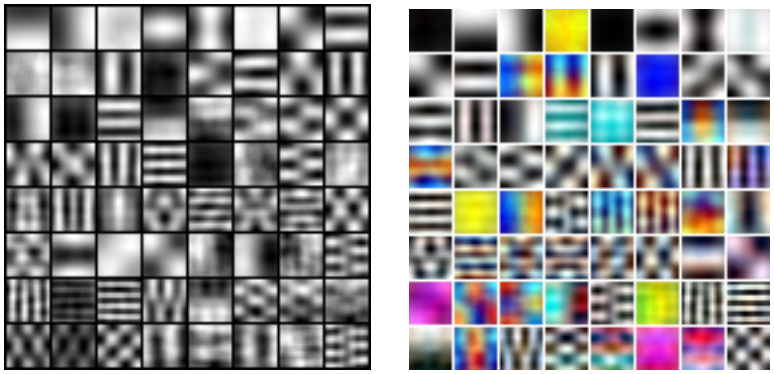
Najlepšie výsledky dosiahnuté pre dátovú sadu UCF11 reprezentuje tabuľka 7.3b. Táto tabuľka ukazuje, že najlepšia kombinácia pre túto dátovú sadu je výber oblastí pomocou homogénnej mriežky a použitie farebných SIFT deskriptorov kľúčových oblastí s polomerom 16 pixlov. Táto úspešnosť klasifikácie bola 73.11 %.

7.2 Výsledky pre Analýzu hlavných komponentov

Experimenty pri príznakoch 3D výrezov videa získaných pomocou PCA vybielenia som zameral na hľadanie optimálneho počtu použitých vlastných vektorov pri PCA vybielení. Keďže vstupom metódy PCA vybielenia boli linearizované 3D výrezy (viď. obrázok 4.4), tak vlastné vektory bolo možné vizualizovať ako 3D filtre (inverznou operáciou k linearizácií). Preto okrem výsledkov uvádzam v tejto časti aj niektoré ukážky získaných vlastných vektorov vo forme vizualizovaných priestorovo-časových filtrov.

Pri hľadaní optimálneho počtu vlastných vektorov, pomocou ktorých boli nové reprezentácie 3D výrezov vytvárané som vyskúšal viacero stratégií ako napríklad percentuálny pomer súčtu všetkých vlastných čísiel voči súčtu vlastných čísiel použitých vlastných vektorov. Najlepšie výsledky som však dosiahol pri systematickom hľadaní optimálneho počtu vlastných vektorov pomocou mocnín čísla 2.

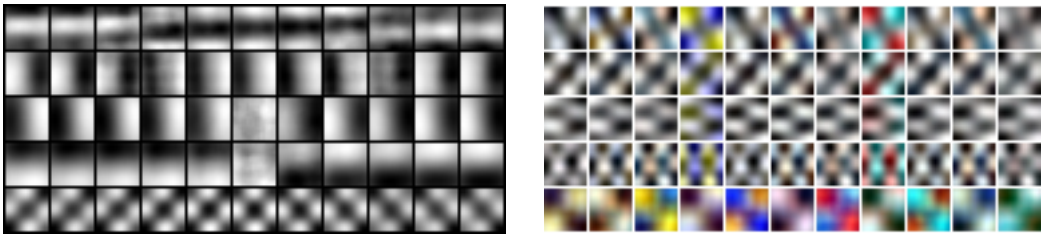
Výsledky na dátovej sade HMDB pre príznaky získané pomocou tejto stratégie reprezentuje tabuľka 7.4a. Ako je v tabuľke vidieť, tak experimenty som vykonal pre farebné a aj pre 3D výrezy v stupňoch šedi. Najlepšiu priemernú úspešnosť klasifikácie som pre výrezy v stupňoch šedi dosiahol pre príznaky vytvorené PCA vybielením pomocou 64 vlastných vektorov. Táto úspešnosť bola 22.9 %. Ukážka priestorových filtrov prostredných výrezov snímok je pre týchto 64 vlastných vektorov zobrazená na obrázku 7.1a. Tieto filtre sú na obrázku 7.1a usporiadané podľa hodnôt vlastných čísiel vektorov, ktorým patria a je vidieť ako sa so znižujúcou hodnotou vlastného čísla zvyšuje priestorová frekvencia v týchto filtroch



(a) Pre snímky videa v stupňoch šedi

(b) Pre farebné snímky videa

Obrázek 7.1: Priestorové filtre pre prostredné výrezy snímok patriace 64 vlastným vektorom. Tieto filtre sú usporiadané zostupne podľa hodnôt vlastných čísiel príslušných vlastných vektorov, ktorým filtre patria.



(a) Pre snímky videa v stupňoch šedi

(b) Pre farebné snímky videa

Obrázek 7.2: Ukážka piatich vlastných vektorov prevedených z linearizovanej formy na ich priestorovo-časovú reprezentáciu (pre výrezy videa 16×16 pixlov $\times 11$ snímok). Je vidieť ako sa jednotlivé filtre výrezov snímok menia v čase. To je spôsobené tým, ako sa menili snímky videí trénovacej sady.

vlastných vektorov. Ukážka niektorých celých vlastných vektorov je pre výrezy v stupňoch šedi zobrazená na obrázku 7.2a. Na tomto obrázku je vidieť ako sa menia jednotlivé priestorové filtre vlastných vektorov v čase. To je spôsobené tým, ako sa v čase menili snímky vo videách trénovacej sady.

Z výsledkov pre farebné 3D výrezy (viď. tabuľka 7.4a) je vidieť, že najlepšia úspešnosť klasifikácie bola tiež dosiahnutá pre príznaky vytvorené PCA vybielením pomocou 64 vlastných vektorov. Táto úspešnosť bola 21.55 %. Ukážka priestorových filtrov prostredných výrezov snímok je pre týchto 64 vlastných vektorov zobrazená na obrázku 7.1b a ukážka piatich celých vlastných vektorov je zobrazená na obrázku 7.2b. Ako je na týchto obrázkoch vidieť, tak tieto vlastné vektory vykazujú podobné chovanie ako je popísané vyššie pri vlastných vektoroch 3D výrezov v stupňoch šedi.

Výsledky experimentov pre dátovú sadu UCF11 reprezentuje tabuľka 7.3b. Z tejto tabuľky je vidieť, že najlepšiu úspešnosť klasifikácie som dosiahol pre príznaky 3D výrezov v stupňoch šedi vytvorené PCA vybielením pomocou 32 vlastných vektorov. Táto úspešnosť

Farba výrezov	Počet vlastných vek.	Úspešnosť klasifikácie
GRAY	8	18.67 %
GRAY	16	21.29 %
GRAY	32	22.46 %
GRAY	**64**	**22.9 %**
GRAY	128	20.87 %
GRAY	256	19.83 %
RGB	8	16.88 %
RGB	16	19.46 %
RGB	32	20.78 %
RGB	64	21.55 %
RGB	128	21.53 %
RGB	256	20.04 %

(a) Výsledky pre HMDB

Farba Farba	Počet vlastných vek.	Úspešnosť klasifikácie
GRAY	8	68.58 %
GRAY	16	68.28 %
GRAY	32	71.3 %
GRAY	64	70.39 %
GRAY	128	67.98 %
GRAY	256	66.77 %
RGB	8	63.75 %
RGB	16	66.16 %
RGB	32	68.58 %
RGB	64	71 %
RGB	128	73.11 %
RGB	**256**	**74.92 %**
RGB	512	72.5 %

(b) Výsledky pre UCF11

Tabulka 7.4: Výsledky klasifikácie videa pri použití systému pre klasifikáciu videa pomocou Bag of Words a použití príznakov vytvorených PCA vybielením priestorovo-časových výrezov videa pri použití rôznych počtov vlastných vektorov.

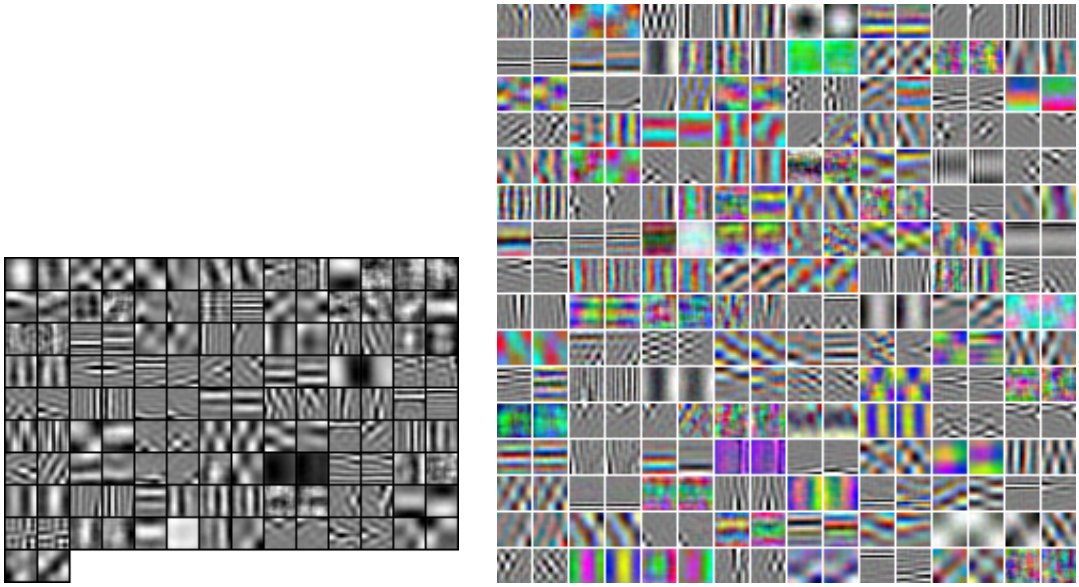
bola 71.3 %. Z tabuľky 7.3b je ďalej vidieť, že najlepšia úspešnosť klasifikácie pre farebné 3D výrezy bola dosiahnutá pre príznaky vytvorené PCA vybielením pomocou 256 vlastných vektorov. Táto úspešnosť bola 74.92 %. Vizualizované vlastné vektory mali rovnaké vlastnosti ako vlastné vektory popísané pri dátovej sade HMDB.

Výhodou príznakov vytvorených PCA vybielením 3D výrezov videa bolo, že pri experimentoch som počas trénovania nemusel nastavovať žiadne špeciálne trénovacie konštanty a jediná vec, ktorú bolo treba hľadať bol počet vlastných vektorov, ktoré sa pri transformácií výrezov používali. Z toho vyplýva ďalšia výhoda, keďže trénovanie trvalo rovnako dlho pre ľubovoľný počet vlastných vektorov, ktoré sa pri transformácií dát používali. Rýchlosť trénovania PCA vybielenia závisela iba od počtu trénovacích vektorov a ich dimenzionality. Avšak pre použité dátové sady trvalo trénovanie maximálne niekoľko minút.

7.3 Výsledky pre Analýzu nezávislých podpriestorov

Experimenty pri príznakoch 3D výrezov videa vytvorených pomocou ISA som zameral na hľadanie optimálnej redukcie dimenzionality vstupných 3D výrezov (pomocou PCA vybielenia) a hľadanie optimálnej veľkosti podpriestoru ISA. Okrem výsledkov uvádzam aj ukážky váh ISA vo forme vizualizovaných 3D filtrov, ktoré sa pri procese trénovania v skrytej vrstve ISA naučili z dát. Keďže vstupom ISA boli PCA vybielené výrezy videa, tak bolo nutné váhy pred vizualizáciou previesť pomocou vlastných vektorov do priestoru linearizovaných 3D výrezov (pred PCA vybielením). Po tomto kroku mohli byť váhy inverznou operáciou k linearizácií vizualizované.

Pri hľadaní optimálneho počtu vlastných vektorov pre PCA vybielenie a pri hľadaní vhodného počtu zlučovacích neurónov som vyskúšal viacero stratégií. Avšak najlepšie výsledky som dosiahol pri systematickom hľadaní týchto počtov, keď počet vlastných vektorov bol hľadaný pomocou mocnín čísla 2 a veľkosť podpriestoru bola potom hľadaná pre kon-



(a) Pre snímky videa v stupňoch šedi

(b) Pre farebné snímky videa

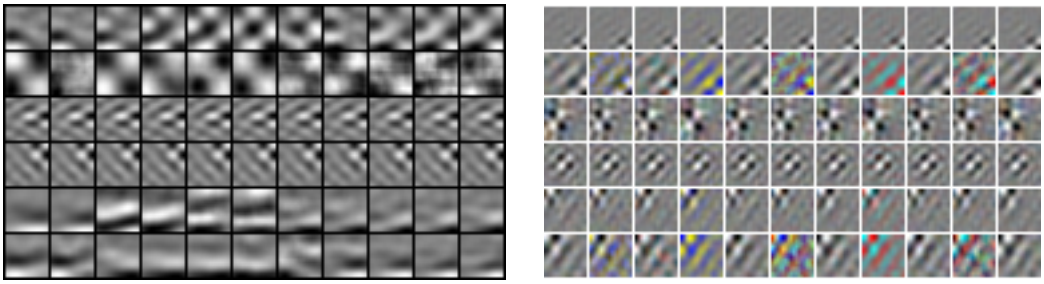
Obrázek 7.3: Priestorové filtre pre prostredné výrezy snímok, ktoré patria váham skrytej vrstvy ISA. Veľkosť podpriestoru je 2. To je vidno aj obrázkoch, pretože sa za sebou nachádzajú vždy dva podobné filtre.

krétnu variantu PCA vybielených výrezov.

Výsledky pri príznakoch získaných pomocou ISA pre dátovú sadu HMDB reprezentuje tabuľka 7.5a a tabuľka 7.6a. Podobne ako pri príznakoch získaných PCA vybielením, tak aj tu som príznaky vytváral pre farebné a aj pre 3D výrezy v stupňoch šedi. Ako je vidieť v tabuľke 7.5a, tak najlepšiu priemernú úspešnosť klasifikácie pre výrezy v stupňoch šedi som dosiahol pre príznaky vytvorené PCA vybielením pomocou 64 vlastných vektorov a ISA sieťou s podpriestorom veľkosti 2. Táto úspešnosť bola 23.46 %. Ukážka priestorových filtrov prostredných výrezov snímok je pre váhy skrytej vrstvy ISA zobrazená na obrázku 7.3a. Na tomto obrázku je dobre vidieť, ako vždy dva filtre patriace neurónom skrytej vrstvy, ktoré sú zlučované do jedného podprietoru sú podobné. To práve zabezpečuje čiastočnú invariantnosť výsledných príznakov voči týmto podobnostným zmenám.

Ako je z tabuľky 7.6a vidieť, tak pre farebné 3D výrezy bola najlepšia úspešnosť klasifikácie dosiahnutá pre príznaky vytvorené PCA vybielením pomocou 256 vlastných vektorov a ISA sieťou s podpriestorom veľkosti 2. Táto úspešnosť bola 22.9 %. Ukážka priestorových filtrov prostredných výrezov snímok je pre váhy neurónov skrytej vrstvy ISA zobrazená na obrázku 7.3b a ukážka niektorých celých váh je zobrazená na obrázku 7.4b. Na obrázkoch je možné vidieť podobné vlastnosti ako sú popísané vyššie pri váhach skrytej vrstvy ISA pre 3D výrezy v stupňoch šedi.

Výsledky experimentov vykonaných na dátovej sade UCF1 reprezentuje tabuľka 7.5b a tabuľka 7.6b. Ako ukazuje táto tabuľka 7.5b, tak najlepšia úspešnosť klasifikácie pre priestorovo-časové výrezy videa v stupňoch šedi bola dosiahnutá pre príznaky vytvorené PCA vybielením pomocou 256 vlastných vektorov a ISA sieťou s podpriestorom veľkosti 8. Táto úspešnosť bola 69.18 %. Z tabuľky 7.6b je zasa vidieť, že najlepšia úspešnosť klasifikácie pre farebné 3D výrezy bola dosiahnutá pre 128 vlastných vektorov a veľkosť podpriestoru 2.



(a) Pre snímky videa v stupňoch šedi

(b) Pre farebné snímky videa

Obrázek 7.4: Ukážka váh šiestich neurónov skrytej vrstvy ISA (pre výrezy videa 16×16 pixlov $\times 11$ snímok). Uvedené sú tri dvojice váh neurónov, ktoré sú v rámci dvojice podobné. Je to preto, lebo tieto dvojice tvoria jeden podpriestor v ISA. Je tiež vidieť ako sa filtre váh v čase menia.

Táto úspešnosť bola 71.9 %.

Výhoda príznakov naučených pomocou ISA bola, že pri experimentoch som pri trénovaní nemusel nastavovať žiadne špeciálne trénovacie konštanty. Jediná vec, ktorú teda bolo nutné hľadať bol optimálny počet vlastných vektorov použitý pri PCA vybielení dát a príslušná veľkosť podpriestoru. Podobne ako je uvedené v sekcií 3.2, tak trénovanie ISA bolo rýchle pre malé rozmery vstupných vektorov (64,128) a trvalo maximálne niekoľko desiatok minút až hodinu. Pri väčších dimenziách vstupných vektorov trvalo trénovanie dlhšie a pri vstupoch s 512 dimenziami to bolo 5 až 8 hodín.

7.4 Výsledky pre Riedke Autoenkodére

Experimenty pri príznakoch 3D výrezov videa získaných pomocou Riedkych Autoenkodérov som zameral na hľadanie optimálnej redukcie dimenzionality vstupných 3D výrezov (pomocou PCA vybielenia) a hľadanie optimálneho počtu neurónov skrytej vrstvy Riedkych Autoenkodérov. Opäť uvádzam aj ukážky váh skrytých neurónov Riedkych Autoenkodérov, ktoré sa pri procese trénovania z dát naučili. Keďže vstupom Riedkych Autoenodérov boli podobne ako pri ISA PCA vybielené výrezy postupoval som pri vizulizácií váh rovnako ako pri ISA (viď. úvod sekcie 7.3).

Na základe experimentov zo sekcie 7.2 a sekcie 7.3 som pri hľadaní optimálneho počtu vlastných vektorov a počtu neurónov skrytej vrstvy použil systematické hľadanie pomocou druhých mocnín čísla 2.

Výsledky experimentov pre dátovú sadu HMDB reprezentuje tabuľka 7.7a. Ako možno v tabuľke vidieť, tak experimenty som vykonával len pre príznaky 3D výrezov v stupňoch šedi. Najlepšiu priemernú úspešnosť klasifikácie som dosiahol použitím príznakov vytvorených pomocou PCA vybielenia so 128 vlastnými vektormi a Riedkych Autoenkodérov so 64 neurónmi v skrytej vrstve. Táto úspešnosť bola 22.88 %. Výsledky ma prekvapili, keďže som očakával, že najlepšie výsledky budú dosiahnuté pre príznaky vytvorené Riedkymi Autoenkodérmi s väčším počtom neurónov skrytej vrstvy. Pretože pri väčších počtoch neurónov skrytej vrstvy je dimenzionalita výstupného vektoru príznakov vyššia.

Ukážka priestorových filtrov prostredných výrezov snímok váh Reidkych Autoenkodérov je pre kombináciu 64 vlastných vektorov a 512 váh skrytej vrstvy zobrazená na obrázku 7.5.

Počet vlastných vek.	Veľkosť podpriestoru	Úspešnosť klasifikácie
64	2	21.53 %
64	4	19.15 %
64	8	17.15 %
128	**2**	**23.46 %**
128	4	20.98 %
128	8	19.39 %
256	2	21.92 %
256	4	21.31 %
256	8	19.72 %
512	2	20.94 %
512	4	20.48 %
512	8	19.39 %

(a) Výsledky pre HMDB a snímky v stupňoch šedi

Počet vlastných vek.	Veľkosť podpriestoru	Úspešnosť klasifikácie
64	2	68.88 %
64	4	63.14 %
64	8	57.40 %
128	2	68.88 %
128	4	65.26 %
128	8	66.16 %
256	2	67.07 %
256	4	68.88 %
256	**8**	**69.18 %**
512	2	66.77 %
512	4	67.67 %
512	8	66.47 %

(b) Výsledky pre UCF11 a snímky v stupňoch šedi

Tabulka 7.5: Výsledky klasifikácie videa pri použití systému pre klasifikáciu videa pomocou Bag of Words a použití príznakov priestorovo-časových výrezov videa v stupňoch šedi, ktoré som vytváral pomocou ISA pre rôzne nastavenia PCA redukcie a rôzne nastavenie veľkostí podpriestoru.

Ako je na obrázku vidieť, tak filtre neurónov skrytej vrstvy pripomínajú Gaborové filtre, čo zodpovedá informáciám z kapitoly 3. Ukážka niektorých celých váh je zobrazená na obrázku 7.6. Opäť je vidieť ako sa menia jednotlivé priestorové filtre váh v čase.

Výsledky experimentov vykonaných na dátovej sade UCF11 reprezentuje tabuľka 7.7b. Podobne ako pri experimentoch na HMDB som vykonal experimenty iba pre 3D výrezy videa v stupňoch šedi. Ako ukazuje táto tabuľka, tak najlepšia úspešnosť klasifikácie bola dosiahnutá pri príznakoch vytvorených pomocou PCA vybielenia so 128 vlastnými vektormi a Riedkych Autoenkodérov s 256 neurónmi v skrytej vrstve. Táto úspešnosť bola 74.02 %.

Pri experimentoch s naučenými príznakmi pomocou Riedkych Autoenkodérov som narazil na niekoľko problémov. Ide o problémy spojené s trénovaním Riedkych Autoenkodérov a to konkrétne o problémy s nastavením viacerých trénovacích konštánt (riedkosť skrytej vrstvy, váha penalizácie pri riedkosti, váha termu zabraňujúcemu pretrénovaniu siete), ktoré bolo nutné správne nastaviť pri trénovaní. Pri zlej voľbe parametrov trvalo trénovanie extrémne dlho (niekoľko týždňov) a filtre produkované Autoenkódermi pripomínali šum. Práve kvôli extrémnej dĺžke trvania trénovania bol problém vyhľadať správnu kombináciu týchto parametrov a vyžiadalo sa to časovo náročné experimenty. Uspokojivé výsledky som dosiahol pri nasledovnom nastavení

$$\beta = 5, \lambda = 3e^{-3}, \rho = 0.035. \tag{7.1}$$

Avšak aj pri správnych hodnotách týchto parametrov a použití algoritmu L-BFGS bolo trénovanie pomalé. Problém nebol ani tak pri malých dimenziách vstupných vektorov (64) a malom počte neurónov skrytej vrstvy (32, 64), kedy trénovanie trvalo niekoľko hodín. Ale problém nastával pri väčšom počte neurónov skrytej vrstvy (256, 512) a väčších dimenziách vstupov (256, 512), kedy trénovanie trvalo od niekoľkých dní až viac než týždeň. Preto som sa rozhodol, že pre príznaky farebných 3D výrezov nebudem experimenty vykonávať.

Počet vlastných vek.	Veľkosť podpriestoru	Úspešnosť klasifikácie
64	2	21.05 %
64	4	18.52 %
64	8	16.06 %
128	2	22.2 %
128	4	20.02 %
128	8	18.24 %
256	**2**	**22.9 %**
256	4	20.96 %
256	8	19.87 %
512	2	21.26 %
512	4	20.81 %
512	8	20.31 %

(a) Výsledky pre HMDB a farebné snímky

Počet vlastných vek.	Veľkosť podpriestoru	Úspešnosť klasifikácie
64	2	67.98 %
64	4	68.58 %
64	8	63.14 %
128	**2**	**71.9 %**
128	4	70.09 %
128	8	60.42 %
256	2	70.39 %
256	4	70.39 %
256	8	69.78 %
512	2	69.49 %
512	4	69.49 %
512	8	66.47 %

(b) Výsledky pre UCF11 a farebné snímky

Tabulka 7.6: Výsledky klasifikácie videa pri použití systému pre klasifikáciu videa pomocou Bag of Words a použití príznakov farebných priestorovo-časových výrezov videa, ktoré som vytváral pomocou ISA pre rôzne nastavenia PCA redukcie a rôzne nastavenie veľkostí podpriestoru.

7.5 Porovnanie dosiahnutých výsledkov

V tejto sekcii sa nachádza porovnanie dosiahnutých výsledkov z predchádzajúcich sekcií. Pre lepší prehľad som najlepšie výsledky jednotlivých príznakov umiestnil do samostatných tabuliek. Pre dátovú sadu HMDB sú najlepšie výsledky umiestnené v tabuľke 7.8a a pre dátovú sadu UCF11 v tabuľke 7.8b.

Ako je vidieť v tabuľke 7.8a, tak výsledky pre dátovú sadu HMDB ukazujú, že všetky naučené príznaky dosiahli lepšiu úspešnosť klasifikácie než referenčné SIFT deskriptory. Z týchto výsledkov najlepšiu úspešnosť klasifikácie 23.46 % dosiahli príznaky získané Analýzou nezávislých podpriestorov, ktorá bola o 0.56 % lepšia než úspešnosť dosiahnutá pri príznakoch naučených pomocou použití PCA vybielenia. Pri porovnaní s výsledkami dosiahnutými autormi datasetu HMDB (viď. tabuľka 7.1) dosahujú naučené príznaky porovnateľné výsledky, ktoré sú o niečo lepšie než sú obidva výsledky dosiahnuté autormi datasetu HMDB pre ručne-navrhnuté príznaky. A to je významná informácia hlavne pre prístup, v ktorom Kuehne et al. použili rovnakú štruktúru systému pre klasifikáciu videa pomocou BOW ako bola použitá v tejto práci. S tým rozdielom, že pre popis 3D výrezov použili HOG/HOF príznaky. Z toho teda vyplýva, že použitie naučených príznakov získaných metódami učenia príznakov pri klasifikácii videa dosahuje lepšie výsledky.

Z výsledkov naučených príznakov je zaujímavý výsledok dosiahnutý pre príznaky získané pomocou PCA vybielenia, pretože ide o jednoduchú a rýchlu metódu a ukazuje sa, že aj táto jednoduchá metóda môže byť použitá úspešne pri klasifikácií videa.

Z výsledkov pre dátovú sadu UCF11 dosiahli najlepšiu úspešnosť klasifikácie naučené príznaky pomocou PCA vybielenia pred naučenými príznakmi pomocou Riedkych Autoenkodérov o 0.9 %. Ako je vidieť v tabuľke 7.8b, tak naučené príznaky pomocou PCA vybielenia a Riedkych Autoenkodérov dosiahli mierne lepšie výsledky než referenčné SIFT deskriptory (o 1.81 % a o 0.91 %). Pri porovnaní dosiahnutých výsledkov s publikovanými výsledkami k tejto dátovej sade dosiali moje výsledky veľmi dobrú úspešnosť. Avšak pod vplyvom

Počet vlastných vek.	Počet neurónov skrytej vrstvy	Úspešnosť klasifikácie
64	32	21.61 %
64	64	22.75 %
64	128	22.68 %
64	256	22.4 %
64	512	21.7 %
128	32	22.27 %
128	**64**	**22.88 %**
128	128	21.61 %
128	256	21.50 %
128	512	20.96 %
256	32	21.66 %
256	64	22.29 %
256	128	21.26 %
256	256	21.26 %
512	32	21.98 %
512	64	22.81 %
512	128	20.92 %

(a) Výsledky pre HMDB a snímky v stupňoch šedi

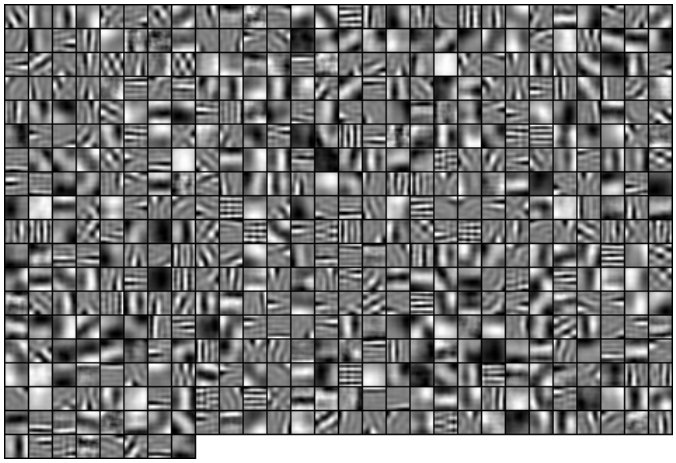
Počet vlastných vek.	Počet neurónov skrytej vrstvy	Úspešnosť klasifikácie
64	32	71.3 %
64	64	71.4 %
64	128	70.69 %
64	256	71 %
64	512	69.79 %
128	32	70.09 %
128	64	72.51 %
128	128	71.3 %
128	**256**	**74.02 %**
128	512	71 %
256	32	71.3 %
256	64	71 %
256	128	72.21 %
256	256	68.28 %
512	32	68.58 %
512	64	71 %
512	128	69.18 %

(b) Výsledky pre UCF11 a snímky v stupňoch šedi

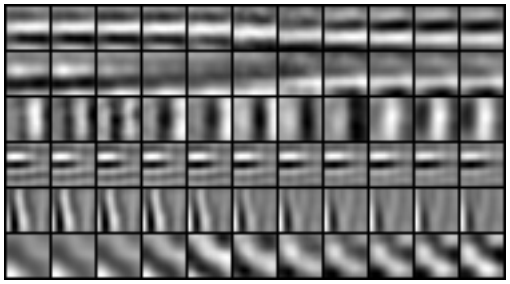
Tabulka 7.7: Výsledky klasifikácie videa pri použití systému pre klasifikáciu videa pomocou Bag of Words a použití príznakov priestorovo-časových výrezov videa v stupňoch šedi, ktoré som vytváral pomocou Riedkych Autoenkodérov pre rôzne nastavenia PCA redukcie a rôzne nastavenie počty neurónov skrytej vrstvy.

spôsobu vykonania experimentov by bolo nutné tieto výsledky potvrdiť vykonaním experimentov pre celú 5-stupňovú crossvalidáciu. Naučené príznaky pomocou PCA vybielenia a Riedkych Autoenkodérov však obstáli pri porovnaní s referenčnými SIFT deksriptormi a teda dosiahli opäť dobré výsledky. Zaujímavé výsledky dosiahli naučené príznaky pomocou ISA, pretože pri dátovej sade HMDB dosiahli najlepšiu úspešnosť klasifikácie, avšak ako je vidno v tabuľke 7.8b, tak pri dátovej sade UFC11 dosiahli tieto príznaky najhoršie výsledky.

Keďže pre dátovú sadu HMDB dosiahli všetky naučené príznaky lepšie výsledky než boli výsledky referenčných SIFT deskriptorov a publikované výsledky autormi tejto dátovej sady pre ručne-navrhnuté príznaky. A pre dátovú sadu UCF11 dosiahli naučené príznaky pomocou PCA vybielenia a Riedkych Autoenkodérov lepšie výsledky než referenčné SIFT deskriptory, tak sa ukazuje teda, že naučené príznaky pomocou PCA vybielenia a Riedkych Autoenkodérov produkujú stabilné výsledky pri rôznych úlohách klasifikácie a pre rôzne dátové sady. Tento fakt by bolo vhodné potvrdiť ešte na viacerých dátových sadách, ale možno na základe týchto výsledkov povedať, že príznaky získané naučením pomocou metód učenia príznakov sú vhodným riešením pre popis videa pre rôzne úlohy klasifikácie a rôzne dátové sady a dosahujú lepšie výsledky než ručne-navrhnuté príznaky, s ktorými boli v tejto práci porovnávané. Tu by som zdôraznil práve výsledok dosiahnutý pri naučených príznakoch pomocou PCA vybielenia, pretože PCA vybielenie je rýchla a jednoduchá metóda učenia príznakov a práve táto metóda vyprodukovala príznaky, ktoré dosiahli pre dátovú sadu HMDB druhý najlepší výsledok a pre dátovu sadu UCF11 najlepší výsledok.



Obrázek 7.5: Priestorové filtre pre prostredné výrezy snímok, ktoré patria váham skrytej vrstvy Riedkeho Autoenkodéru. Tieto váhy boli naučené na priestorovo-časových výrezoch videa v stupni šedi.



Obrázek 7.6: Ukážka šiestich váh skrytej vrstvy Riedkeho Autoenkodéru (pre výrezy videa 16×16 pixlov $\times11$ snímiek). Tieto váhy boli naučené na priestorovo-časových výrezoch videa v stupni šedi a je vidieť, ako sa naučené filtre týchto váh pre jednotlivé výrezy snímok menia v čase.

7.6 Výsledky navrhnutej metódy Multiple Kernel Learning

Táto sekcia obsahuje výsledky dosiahnuté navrhnutou metódou Multiple Kernel Learning, ktorá kombinovala predpočítané jadrá BOW (reprezentácií videí) jednotlivých typov príznakov s cieľom zvýšiť dosiahnutú úspešnosť klasifikácie videa. Pre tento účel som využíval predpočítané jadrá z experimentov popísaných v predchádzajúcich sekciách. Keďže však týchto jadier bolo relatívne dosť, tak vyskúšanie všetkých možných kombinácií jadier (dvoch, troch, štyroch, ...) by bolo extrémne časovo náročné. Preto som sa zameral na otestovanie zaujímavých kombinácií týchto predpočítaných jadier.

Dosiahnuté výsledky experimentov pre dátovú sadu HMDB obsahuje tabuľka 7.9a. Označenie ALL reprezentuje použitie všetkých predpočítaných jadier príznakov daného typu (typ uvedený v stĺpci Zvolená kombinácia), ktoré som pre tento typ príznakov v experimentoch získal (aj pre farebné výrezy aj pre výrezy v stupňoch šedi). Označenie BEST naopak reprezentuje použitie predpočítaného jadra najlepšieho výsledku pre daný typ príznakov. Ak bol však daný typ príznakov použitý pre popis farebných 3D výrezov a aj

Metóda extrakcie príznakov	Úspešnosť klasifikácie
SIFT	19.26 %
PCA vybielenie	22.9 %
Riedke Autoenkodére	22.88 %
ISA	**23.46 %**

(a) Výsledky pre HMDB

Metóda extrakcie príznakov	Úspešnosť klasifikácie
SIFT	73.11 %
PCA vybielenie	**74.92 %**
Riedke Autoenkodére	74.02 %
ISA	71.9 %

(b) Výsledky pre UCF11

Tabulka 7.8: Najlepšie dosiahnuté výsledky klasifikácie videa pre všetky použité príznaky.

Použité jadrá	Zvolená kombinácia	Úspešnosť klasifikácie
ALL	SIFT	22.00 %
ALL	PCA	20.78 %
ALL	AE	22.64 %
ALL	ISA	23.46 %
ALL	SIFT+PCA	23.51 %
ALL	**SIFT+AE**	**26.45 %**
ALL	SIFT+ISA	25.99 %
ALL	SIFT+PCA+AE+ISA	26.12 %
BEST	SIFT+PCA+AE+ISA	26.3 %
BEST	PCA+AE+ISA	23.83 %

(a) Výsledky pre HMDB

Použité jadrá	Zvolená kombinácia	Úspešnosť klasifikácie
ALL	SIFT	68.28 %
ALL	PCA	69.18 %
ALL	AE	72.92 %
ALL	ISA	72.80 %
ALL	SIFT+PCA	75.23 %
ALL	SIFT+AE	75.53 %
ALL	SIFT+ISA	74.01 %
ALL	SIFT+PCA+AE+ISA	74.92 %
BEST	**SIFT+PCA+AE+ISA**	**77.04 %**
BEST	PCA+AE+ISA	76.13 %
BEST	**SIFT+ISA**	**77.04 %**

(b) Výsledky pre UCF11

Tabulka 7.9: Výsledky dosiahnuté pomocou navrhnutej metódy Multiple Kernel Learning. ALL označuje použitie všetkých predpočítaných jadier z vykonaných experimentov pre príznaky uvedené v stĺpci Zvolená kombinácia. BEST označuje použitie jadra najlepšieho výsledku pre príznaky uvedené v stĺpci Zvolená kombinácia. PCA označuje príznaky získané PCA vybielením, ISA označuje príznaky získané Analýzou nezávislých podpriestorov, AE označuje príznaky získané Riedkymi Autoenkodérmi a SIFT označuje SIFT deskriptory.

pre popis 3D výrezov v stupňoch šedi, tak sa použili jadrá najlepších výsledkov pre obe možnosti (toto platí pre naučené príznaky).

Z tabuľky 7.9a vyplýva, že kombinácia jadier jedného typu priniesla zlepšenie úspešnosti klasifikácie iba pre referenčné SIFT deskriptory a to na 22 %. Pri ostatných typoch príznakov sa úspešnosť zhoršila alebo ostala približne rovnaká. Avšak zaujímavé výsledky boli dosiahnuté pri skombinovaní jadier referenčných SIFT deskriptorov s jadrami niektorého typu naučených príznakov. Všetky takéto kombinácie dosiahli značné zvýšenie úspešnosti klasifikácie a ako je možno vidieť v tabuľke 7.9a, tak práve kombinácia jadier SIFT deskriptorov a jadier naučených príznakov pomocou Riedkych Autoenkodérov dosiahla úplne najvyššiu úspešnosť klasifikácie.

Ako možno ďalej v tabuľke 7.9a vidieť, tak som tiež vyskúšal skombinovať všetky predpočítané jadrá pre všetky typy príznakov a úspešnosť klasifikácie bola tiež značne vysoká, avšak oproti kombinácií jadier SIFT deskriptorov a jadier naučených príznakov pomocou Riedkych Autoenkodérov sa nezvýšila. Lepšia úspešnosť klasifikácie pre kombináciu jadier všetkých typov príznakov bola dosiahnutá pri kombinácií jadier dosahujúcich najlepší vý-

sledok pre príslušný typ príznakov.

Z tabuľky 7.9a je vidieť ešte jeden zaujímavý výsledok, ktorým je výsledok získaný pre kombináciu jadier najlepších výsledkov pre všetky typy naučených príznakov. Pretože ako možno vidieť, tak výsledok tejto kombinácie je skoro o 3% menší než pri kombinácií obsahujúcej aj jadro najlepšieho výsledku SIFT deskriptorov.

Výsledky pre dátovú sadu UCF11 sa nachádzajú v tabuľke 7.9b. Ako je vidieť v tejto tabuľke, tak podobne ako pri výsledkoch vyššie sa pri skombinovaní jadier jedného typu príznakov nedosiahli lepšie výsledky, ale úspešnosť klasifikácie naopak klesla. Avšak pri kombinácií jadier referenčných SIFT deskriptorov s jadrami niektorého typu naučených príznakov sa úspešnosť opäť zvýšila. Pri kombinácií všetkých jadier všetkých typov príznakov bola úspešnosť klasifikácie rovnaká ako najlepší výsledok príznakov získaných PCA vybielením. Zlepšenie úspešnosti (o 2.12 %) tejto kombinácie nastalo podobne ako pre HMDB, keď som použil pre každý typ príznakov jadro najlepšieho dosiahnutého výsledku. Podobne ako pri HMDB som vyskúšal experiment s kombináciou všetkých typov naučených príznakov bez jadra najlepšieho výsledku SIFT deskriptorov a úspešnosť klasifikácie sa znížila o 0.91 %.

Ako je ďalej vidieť v tabuľke 7.9b, tak zaujímavý výsledok som dosiahol pri kombinácií jadier najlepších výsledkov SIFT deskriptorov a príznakov naučených pomocou ISA. Zaujímavý výsledok preto, lebo pri tejto kombinácií som dosiahol najlepšiu úspešnosť klasifikácie, avšak naučené príznaky pomocou ISA samostatne dosiahli najhoršie výsledky a úspešnosť SIFT deskriptorov tiež nebola príliš vysoká.

Z týchto výsledkov pre obe dátové sady je teda vidno, že kombinácia jadier rovnakého typu príznakov neprináša lepšie výsledky. Čiastočne to platí aj pre kombináciu jadier všetkých typov naučených príznakov, kedy sa úspešnosť síce zvýšila, ale nedosiahla najlepší výsledok. Avšak pri použití kombinácie jadier SIFT deskriptorov a niektorého typu naučených príznakov sa úspešnosť klasifikácie značne zvyšuje a môže to byť aj vtedy ak použité príznaky samostatne nedosahujú príliš veľkú úspešnosť klasifikácie (viď. kombinácia SIFT + ISA pre dátovú sadu UCF11). Ukazuje sa teda, že kombinácia ručne-navrhnutých príznakov s naučenými príznakmi môže priniesť značné zvýšenie úspešnosti klasifikácie a to aj napriek tomu ak samostatne tieto príznaky nedosahujú príliš dobré výsledky.

Kapitola 8

Záver

Pre potreby riešenej úlohy som preštudoval základné príznaky a metódy ich extrakcie, ktoré sa používajú pri klasifikácií videa. Na základe tejto štúdie a štúdie o metódach klasifikácie pomocou týchto príznakov som vytvoril kapitolu 2.

Ďalší prehľad som si spravil o metódach učenia príznakov. Tento prehľad pozostával z pochopenia činnosti neurónových sietí, ich trénovania a pochopenia činností jednotlivých metód učenia príznakov. Na základe tejto štúdie bola vytvorená kapitola 3.

Ako príznaky pre klasifikáciu videa som použil príznaky naučené pomocou metód učenia príznakov, ktorými boli PCA vybielenie, Analýza nezávislých podpriestorov a Riedke Autoenkodére. Tieto metódy učenia príznakov som vybral s cieľom čo najlepšie popísať video nezávisle na klasifikačnej úlohe. Pre ich zapojenie do procesu extrakcie príznakov bolo nutné pochopiť akú súvislosť majú tieto metódy s obrazom a ako používajú natrénované bázy pri jeho transformácií na novú reprezentáciu. Pre ich zapojenie do procesu extrakcie príznakov pre klasifikáciu videa pomocou Bag of Words som vytvoril kapitolu 4. Okrem toho som v tejto kapitole popísal spôsob extrakcie použitých referenčných príznakov videa, ktorými boli SIFT deskriptory a navrhnutú metódu inšpirovanú princípom Multiple Kernel Learning s cieľom zlepšiť výslednú úspešnosť klasifikácie prostredníctvom kombinácie použitých príznakov.

Popísané metódy som buď implementoval alebo som použil existujúce nástroje, čo som popísal v kapitole 5.

Pre experimenty som vybral dve dátové sady HMDB (pre klasifikáciu ľudskej činnosti) a UCF11 (pre klasifikáciu akcie vo videu), ktoré sú voľne dostupné a pre ktoré existujú referenčné výsledky. Tieto sady som popísal v kapitole 6.

Výsledky experimentov, ktoré som vykonal a porovnal sa nachádzajú v kapitole 7. Z výsledkov som zistil, že naučené príznaky dosahujú dobré výsledky pri klasifikácií videa a v porovnaní s publikovanými výsledkami ručne-navrhnutých príznakov som pomocou naučených príznakov dosiahol lepšie výsledky.

Z výsledkov experimentov s kombináciami použitých príznakov pomocou navrhnutej metódy Multiple Kernel Learning som zistil, že pri kombinácií príznakov jedného typu sa výsledky nezlepšili, avšak pri kombinácií naučených príznakov a ručne-navrhnutých príznakov som dosiahol značného zlepšenia úspešnosti klasifikácie videa. Toto značné zvýšenie úspešnosti klasifikácie som dosiahol aj v prípade, keď ručne-navrhnuté príznaky a naučené príznaky samostatne nedosiahli príliš vysoké úspešnosti klasifikácie.

Ako rozšírenie tejto práce by bolo vhodné dokončiť experimenty pre dátovú sadu UCF11 a potvrdiť tak dobré výsledky, ktoré sú na nej momentálne dosiahnuté pomocou naučených príznakov. Okrem toho by bolo zaujímavé porovnať naučené príznaky získané pomocou

Riedkych Autoenkodérov a Riedkymi Boltzmanovými strojmi a otestovanie kombinácie príznakov pomocou inej metódy ako bola použitá v tejto práci (navrhnutá metóda inšpirovaná Multiple Kernel Learning). V budúcnosti by som tiež chcel vyskúšať použiť pre vytvorenie novej reprezentácie videa viac vrstiev použitých metód učenia príznakov, ktoré by boli schopné vo videu zachytiť komplexnejšie štruktúry.

Literatura

[1] SGE [online]. 2013-05-13 [cit. 2013-05-14].
URL http://merlin.fit.vutbr.cz/wiki/index.php?title=SGE

[2] Exercise:Sparse Autoencoder [online]. 2013-07-10 [cit. 2013-05-14].
URL
http://ufldl.stanford.edu/wiki/index.php/Exercise:Sparse_Autoencoder

[3] RapidMiner [online]. [cit. 2011-05-15].
URL http://rapid-i.com/content/view/181/190/lang,en/

[4] Bell, A. J.; Sejnowski, T. J.: The "independent components"of natural scenes are edge filters. *Vision research*, ročník 37, č. 23, Prosinec 1997: s. 3327–3338, ISSN 0042-6989.
URL http://view.ncbi.nlm.nih.gov/pubmed/9425547

[5] Beran, V.; Hradiš, M.; Otrusina, L.; aj.: Brno University of Technology at TRECVid 2011. In *TRECVID 2011: Participant Notebook Papers and Slides*, National Institute of Standards and Technology, 2011, str. 10.
URL http://www.fit.vutbr.cz/research/view_pub.php?id=9841

[6] Brezeale, D.; Cook, D.: Automatic Video Classification: A Survey of the Literature. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, ročník 38, č. 3, may 2008: s. 416 –430, ISSN 1094-6977.

[7] Calonder, M.; Lepetit, V.; Ozuysal, M.; aj.: BRIEF: Computing a Local Binary Descriptor Very Fast. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ročník 34, č. 7, 2012: s. 1281–1298, ISSN 0162-8828.

[8] Chang, C.-C.; Lin, C.-J.: LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, ročník 2, 2011: s. 27:1–27:27, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[9] Coates, A.; Ng, A. Y.; Lee, H.: An Analysis of Single-Layer Networks in Unsupervised Feature Learning. *Journal of Machine Learning Research - Proceedings Track*, ročník 15, 2011: s. 215–223.
URL http://dblp.uni-trier.de/db/journals/jmlr/jmlrp15.html#CoatesNL11

[10] Dollar, P.; Rabaud, V.; Cottrell, G.; aj.: Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, oct. 2005, s. 65 – 72.

[11] Eronen, A.; Peltonen, V.; Tuomi, J.; aj.: Audio-based context recognition. *Audio, Speech, and Language Processing, IEEE Transactions on*, ročník 14, č. 1, jan. 2006: s. 321 – 329, ISSN 1558-7916.

[12] van Gemert, J.; Veenman, C.; Smeulders, A.; aj.: Visual Word Ambiguity. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, ročník 32, č. 7, july 2010: s. 1271 –1283, ISSN 0162-8828.

[13] Gönen, M.; Alpaydın, E.: Multiple Kernel Learning Algorithms. *J. Mach. Learn. Res.*, ročník 12, Červenec 2011: s. 2211–2268, ISSN 1532-4435.
URL http://dl.acm.org/citation.cfm?id=1953048.2021071

[14] Guan, G.; Wang, Z.; Yu, K.; aj.: Video Summarization with Global and Local Features. In *Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on*, july 2012, s. 570 –575.

[15] Hradiš, M.; Řezníček, I.; Behúň, K.: Semantic Class Detectors in Video Genre Recognition. In *Proceedings of VISAPP 2012*, SciTePress - Science and Technology Publications, 2012, ISBN 978-989-8565-03-7, s. 640–646.
URL http://www.fit.vutbr.cz/research/view_pub.php?id=9853

[16] Jiang, Y.-G.; Bhattacharya, S.; Chang, S.-F.; aj.: High-level event recognition in unconstrained videos. In *International Journal of Multimedia Information Retrieval*, New York, NY, USA: Springer-Verlag, 2012, ISSN 2192-662X.
URL http://link.springer.com/article/10.1007%2Fs13735-012-0024-2

[17] Jiang, Y.-G.; Ngo, C.-W.; Yang, J.: Towards optimal bag-of-features for object categorization and semantic video retrieval. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, CIVR '07, New York, NY, USA: ACM, 2007, ISBN 978-1-59593-733-9, s. 494–501.
URL http://doi.acm.org/10.1145/1282280.1282352

[18] Kogler, M.; Del Fabro, M.; Lux, M.; aj.: Global vs. Local Feature in Video Summarization: Experimental Results. In *Proceedings of the 10th International Workshop of the Multimedia Metadata Community on Semantic Multimedia Database Technologies (SeMuDaTe'09) in conjunction with the 4th International Conference on Semantic and Digital Media Technologies (SAMT 2009)*, editace R. Klamma; H. Kosch; M. Lux; F. Stegmaier, Aachen, Germany: http://ceur-ws.org, Prosinec 2009, str. 6.
URL
http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-539/

[19] Kuehne, H.; Jhuang, H.; Garrote, E.; aj.: HMDB: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[20] Le, Q.; Zou, W.; Yeung, S.; aj.: Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, june 2011, ISSN 1063-6919, s. 3361 –3368.

[21] Le, Q. V.; Ngiam, J.; Coates, A.; aj.: On Optimization Methods for Deep Learning. 2011.

[22] Lisin, D.; Mattar, M.; Blaschko, M.; aj.: Combining Local and Global Image Features for Object Class Recognition. In *Computer Vision and Pattern Recognition - Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, june 2005, ISSN 1063-6919, str. 47.

[23] Liu, D.; Shyu, M.-L.; Chen, C.; aj.: Integration of global and local information in videos for key frame extraction. In *Information Reuse and Integration (IRI), 2010 IEEE International Conference on*, aug. 2010, s. 171 –176.

[24] Liu, J.; Luo, J.; Shah, M.: Recognizing realistic actions from videos "in the wild". In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, june 2009, ISSN 1063-6919, s. 1996 –2003.

[25] Lowe, D. G.: Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, ročník 60, č. 2, Listopad 2004: s. 91–110, ISSN 0920-5691.
URL http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94

[26] Lucas, B. D.; Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision (DARPA). In *Proceedings of the 1981 DARPA Image Understanding Workshop*, April 1981, s. 121–130.

[27] Mandel, M.; Ellis, D.: Song-level features and support vector machines for music classification. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR 2005)*, 2005, s. 594–599.
URL http://www.bibsonomy.org/bibtex/20067d1077b462e5bfbbb204aab2aa4c2/andre%40ismll

[28] Messing, R.; Pal, C. J.; Kautz, H. A.: Activity recognition using the velocity histories of tracked keypoints. In *ICCV*, 2009, s. 104–111.
URL http://dx.doi.org/10.1109/ICCV.2009.5459154

[29] Ng, A.: Sparse Autoencoder. *CS294A Lecture notes*, 2011.
URL http://www.stanford.edu/class/cs294a/sparseAutoencoder.pdf

[30] Ng, A.; Ngiam, J.; Foo, C. Y.; aj.: Linear Decoders [online]. 2011-05-26 [cit. 2012-12-29].
URL http://ufldl.stanford.edu/wiki/index.php/Linear_Decoders

[31] Niebles, J. C.; Chen, C.-W.; Fei-Fei, L.: Modeling temporal structure of decomposable motion segments for activity classification. In *Proceedings of the 11th European conference on Computer vision: Part II*, ECCV'10, Berlin, Heidelberg: Springer-Verlag, 2010, ISBN 3-642-15551-0, 978-3-642-15551-2, s. 392–405.
URL http://dl.acm.org/citation.cfm?id=1888028.1888059

[32] Oliva, A.; Torralba, A.: Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *International Journal of Computer Vision*, ročník 42, 2001: s. 145–175.

[33] Sapienza, M.; Cuzzolin, F.; Torr, P.: Learning discriminative space-time actions from weakly labelled videos. In *Proceedings of the British Machine Vision Conference*, BMVA Press, 2012, ISBN 1-901725-46-4, s. 123.1–123.12.

[34] nguyen Ta, D.; chao Chen, W.; Gelfand, N.; aj.: SURFTrac: Efficient Tracking and Continuous Object Recognition using Local Feature Descriptors. In *In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR09*, 2009.

[35] Wang, F.; Jiang, Y.-G.; Ngo, C.-W.: Video event detection using motion relativity and visual relatedness. In *Proceedings of the 16th ACM international conference on Multimedia*, MM '08, New York, NY, USA: ACM, 2008, ISBN 978-1-60558-303-7, s. 239–248, doi:10.1145/1459359.1459392.
URL http://doi.acm.org/10.1145/1459359.1459392

[36] Wang, H.; Klaser, A.; Schmid, C.; aj.: Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, june 2011, ISSN 1063-6919, s. 3169 –3176.

[37] Wang, H.; Ullah, M. M.; Kläser, A.; aj.: Evaluation of local spatio-temporal features for action recognition. In *British Machine Vision Conference*, sep 2009, str. 127.
URL http://lear.inrialpes.fr/pubs/2009/WUKLS09

[38] Zhang, J.; Huang, K.; Yu, Y.; aj.: Boosted local structured HOG-LBP for object localization. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, june 2011, ISSN 1063-6919, s. 1393 –1400.

Příloha A

Obsah CD

Obsah priloženého CD:

- Adresár `featLearMethods` so zdrojovými kódmi v jazyku MATLAB pre trénovanie a vytváranie novej reprezentácie vstupných 3D výrezov pomocou metód učenia príznakov, ktorými sú Analýza nezávislých podpriestorov, Riedke Autoenkodére a vybielenie Analýzov hlavných komponentov. Tento adresár tiež obsahuje súbor README_FLM.txt, ktorý popisuje inštaláciu a použitie týchto kódov pre učenie príznakov videa.
- Adresár `svm` so zdorojovými kódmi v jazyku MATLAB pre prácu s klasifikátorom Support Vector Machine z knižnice LibSVM pre klasifikáciu a navrhnutú metódu Multiple Kernel Learning. Tento adresár tiež obsahuje súbor README_SVM.txt, ktorý popisuje inštalaciu a použitie vytvorených funkcií.
- Zdrojové kódy tejto práce pre LaTeXsú umiestnené v adresári `report`.
- Plagát v PDF.

Příloha B

Plakat

Príznaky pre klasifikáciu videa

Použitý proces klasifikácie videa

Priestorovo-časové výrezy videa

PCA vybielenie

Riedky Autoenkodér

Analýza Nezávislých podpriestorov

Slovník kódových slov

Slovníkový preklad

Bag of Words reprezentácia videa

Support Vector Machine

$$K(X,X') = \exp\left(-\gamma \sum_{n=1}^{V} \frac{(x_n - x_n')^2}{x_n + x_n'}\right)$$

Multiple kernel learning

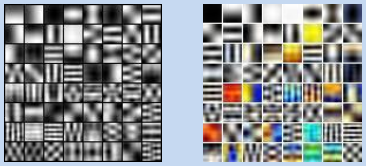
$$\sum_{i=1}^{n} K_i(X,X')/n$$

Výhoda:
- príznaky sa dokážu prispôsobiť pre konkrétnu sadu videí a konkrétnu úlohu

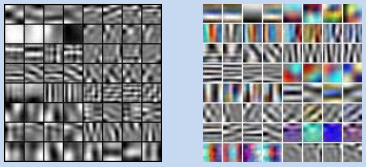
Nevýhoda:
- nutnosť natrénovania použitej metódy učenia príznakov

Ukážky naučených filtrov pre prostredné snímky priesorovo-časových výrezov

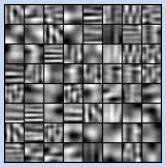
PCA vybielenie



Analýza nezávislých podpriestorov



Riedke autoenkodére



Autor: Bc. Kamil Behúň
Vedúci: Ing. Michal Hradiš