ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

Katedra informačních technologií

Rozpoznávání generických objektů dané třídy v obraze

Diplomová práce

Vedoucí práce: Ing. Čestmír Halbich, CSc.

Diplomant: Michal Češpiva

2014

Abstrakt

V diplomové práci je navržen detektor chodců s využitím Houghova lesa a jemu předřazené kaskády binárních rozhodovacích stromů. Detekce je prováděna pomocí hledání maximální aposteriorní pravděpodobnosti sdružené pravděpodobnosti existence objektu na určité souřadnici a příslušnosti patchů k tomuto objektu. Teoretická část práce pak popisuje především techniky a postupy využité při samotném návrhu a implementaci detektoru: zobecněnou Houghovu transformaci pro hledání souřadnic detekovaných objektů, histogramy orientovaných gradientů (HOG) použité pro reprezentaci objektů nebo odvození sdružené pravděpodobnosti výskytu objektu a jemu příslušejících patchů obrazu.

Abstract

The aim of this thesis is to design and implement a pedestrian detector using Hough Forest with a serialized cascade of random decision trees working as weak classifiers for scene preprocessing. The detection task itself is handled using MAP inference on a joint probability distribution of hypotheses about object existence in a particular spatial configuration and image patches that vote for such objects in Hough voting space. The theory introducing this paper deals mainly with techniques that were used when designing and implementing the pedestrian detector. This includes Generalized Hough Transform, Histograms of Oriented Gradients or inference of the underlying joint probability function used for pedestrian detection in a given scene.

Klíčová slova

Rozpoznávání objektů, HOG, SVM, SIFT, Bayesův teorém, obecná Houghova transformace, MAP inference, kaskádová klasifikace, Houghův les, binární rozhodovací strom.

Keywords

Object detection, HOG, SVM, SIFT, Bayes' theorem, Generalized Hough Transform, MAP inference, cascade classification, Hough forest, binary decision tree.

Rozpoznávání generických objektů dané třídy v obraze

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně, pod vedením Ing. Čestmíra Halbicha, CSc. a uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Poděkování

Rád bych poděkoval za odborné vedení a cenné rady vedoucímu mé diplomové práce Ing. Čestmíru Halbichovi, CSc.

…….

Michal Češpiva

20. listopadu 2014

Obsah

Kapitola 1

1.1 Úvod

Přibližně polovina povrchu lidského mozku je nějakým způsobem zapojena do zpracování vjemů přicházejících do mozkové kůry skrze oči [1]. Nepřekvapí tedy, že přestože zdraví lidé právě díky tomuto obrovskému výpočetnímu výkonu dokážou instance mnoha různých tříd objektů rozpoznávat bez sebemenší námahy, počítačové vidění je naopak jedním z nejnáročnějších oborů umělé inteligence. Samotná oblast rozpoznávání objektů v obraze je pak mnohými považována za královnu oboru.

Přestože výsledky prudkého vývoje, minimálně v posledních 13 letech od implementace známého detektoru **Viola–Jones** v roce 2001[2], našly uplatnění v širokém spektru praktických aplikací, s nimiž přicházíme každodenně do styku, detektory se svými výstupy stále neblíží schopnostem lidského mozku a není jasné, zda této mety vůbec kdy dosáhnou – problém detekce je považován za AI kompletní [3].

1.2 Cíl práce a metodika

Obsah této práce se zabývá návrhem a implementací detektoru objektů dané třídy. Konkrétní implementace návrhu by měla být schopna rozpoznat několik instancí objektu „chodec“ z třídy objektů „chodci“ v testovaném obraze. Detektor je založen na takzvaném Houghově lese (Hough forest) kombinujícím prediktivní model lesa (random forest) binárních rozhodovacích stromů a obecnou Houghovu transformaci. Návrh vychází a kombinuje poznatky z několika předchozích prací na dané téma [4, 5, 6]. Inovací je zkombinování „klasického“ Houghova lesa s jeho paralelním řazením binárních rozhodovacích stromů a myšlenky několika vrstev jemu předřazených, do série zapojených binárních rozhodovacích stromů, tzv. slabých klasifikátorů, které mají za úkol snížit výpočetní náročnost a detekci objektů tak urychlit při pokud možno nezměněné přesnosti detekce. Zde vycházím z předpokladu, že snazší a tedy i rychlejší je nalézt místa, kde se objekt(y) nenachází a poslední stupeň detektoru, tj. paralelní Houghův les (silný klasifikátor) a jeho výpočetní

výkon/náročnost nechat pracovat až na hypotézách, které nebylo možné okamžitě zamítnout klasifikátory slabými.

Práce je rozdělena do několika částí. Nejprve obecně popisuje problematiku detekce objektů, výzvy, které před odborníky z oboru stojí, řešení, kterými se jim snaží čelit a několik nejdůležitějších znalostí, z nichž tato řešení vychází (kapitola 2). V další kapitole je podrobný výklad teoretických východisek, na kterých stojí navržený detektor (kapitola 3). Na tuto kapitolu navazuje popis samotného detektoru (kapitola 4), jeho testování a evaluace (kapitola 5). Závěrem je pak prostor pro diskusi výsledků a možných budoucích směrů, kterými by se práce mohla vydat (kapitola 6).

Kapitola 2

2.1 Detekce a rozpoznávání objektů

Na nejvyšší úrovni rozdělení je detekce nebo rozpoznávání objektů v obraze dvojího typu. Za prvé je to úloha nalezení konkrétního objektu, například jednoho konkrétního člověka, místa, jedné specifické budovy nebo rasy psa mezi všemi ostatními objekty z dané třídy a za druhé pak úloha identifikace všech objektů náležejících do jedné nebo více tříd s veškerou jejich variabilitou tvarů, rozměrů a barev, například nalezení všech židlí, lidí nebo aut v obraze.

Každá z těchto dvou základních úloh detekce objektů vyžaduje jiný přístup. V detekci konkrétních objektů jde o nalezení a spárování z hlediska specifičnosti pro daný objekt významných rysů a jejich následné geometrické ověření, zatímco při detekci generických objektů určité třídy jde především o nalezení co nejobecnějších rysů této kategorie objektů, na jejichž základě je vytvořen/naučen statistický model jejich vzhledu použitý pro predikci přítomnosti a případně lokalizaci dosud neviděného objektu dané kategorie v novém obraze.

Nadále se tato práci zaměřuje na teoretické základy postupů a technik využívaných především pro řešení úloh druhého typu – rozpoznávání generických objektů dané třídy – tedy i úlohy řešené v praktické části této práce.

2.2 Výzvy a překážky

V nestylizovaných scénách a záznamech těchto scén skrze objektivy fotoaparátů a kamer zachycujících běžný život, se objekty zájmu běžně nachází v zákrytu za objekty jinými. Objekty v rámci jedné kategorie existují v mnoha různých variantách barev, tvarů i velikostí. Lze je pozorovat za ztížených povětrnostních nebo snížených světelných podmínek. Stejně tak se mohou stavět do různých pozic nebo se různě deformovat. Jejich vzhled bude záviset i na pozici kamery a detekci může značně ztížit i něco tak těžko ovlivnitelného jako je pozadí, na kterém objekt pozorujeme – i pro lidské oko platí, že čím složitější pozadí co do tvarů a barev, tím obtížnější bude v nich nalézt záchytné body pro rychlou interpretaci scény.

Toto všechno naznačuje, že nelze provádět detekci pouze skrze prosté porovnání předloženého obrázku proti ať už jakkoli velké databázi vzorů objektů z kategorie, o kterou se zajímáme. Naopak je třeba hledat techniku, která nám umožní odhlédnout od takovéto vnitřní variability dané kategorie a spíše nám pomůže nalézt ty charakteristiky, které jsou uvnitř kategorie neměnné, nebo se jejich variabilita dá snadno modelovat.

Jedním z příkladů může být detekce chodců skrze kameru automobilu. Chodci čekající na přechodu mohou být částečně v zákrytu za jiným, u kraje silnice, za zaparkovaným autem, mohou být ve tmě osvětleni pouze světlomety přijíždějícího auta, kdy je po dlouhou dobu viditelná pouze část jejich těla. Může hustě sněžit, mohou sedět na kole nebo se pohybovat na pestré ulici rušného velkoměsta.

Obecně je vždy třeba odpovědět si na následující tři otázky:

- jakou zvolit pro objekty třídy reprezentaci a s tím spojený typ modelu?
- Jak naučit na základě této reprezentace klasifikátor objektů?
- A jak tento klasifikátor použít pro rozpoznávání nových objektů dané třídy.

Následující podkapitoly přiblíží některé v současné době hojně využívané přístupy k překonávání výše popsaných problémů a tím i odpovědi na tyto tři základní otázky. Nejedná se v žádném případě o vyčerpávající výčet všech možností, nýbrž o pouhý nástin možného a popis směrů, jimiž se v současné době rozpoznávání objektů ubírá a z části také úvod do technologií použitých v praktické části této práce.

2.3 Reprezentace objektů

Všechny možné techniky reprezentace vzhledu objektů, jež se v současné době v úlohách detekce objektů v obraze používají, lze rozdělit do dvou skupin:

- globální reprezentace a
- reprezentace založené na částech a geometrických vztazích mezi těmito částmi

2.3.1 Globální reprezentace

Pod pojmem globální je zde míněna reprezentace vzhledu objektu jako jeho celku. Nejjednodušší globální reprezentací je prostý vzorový obrázek objektu, který se při detekci porovnává s částmi na tento objekt testovaného obrázku.

Vzhled obrazu je zaznamenán pomocí vektoru pixelů tomuto objektu náležejících, tedy jejich intenzit ve stupních šedi, případně pomocí histogramu barev. Barva může být v některých případech pro detekci docela dobrým vodítkem, například pro rychlé nalezení oblastní obrazu s možným výskytem obličeje. V jiných případech však již příliš vypovídající není, zejména díky neuspokojivě vyřešené otázce změny barvy pod umělým osvětlením různých teplot. Detekce pak probíhá posuvem pixel za pixelem po testovaném obraze a výpočtem vzájemné korelace dvou vektorů stupňů šedi případně porovnáním histogramů barev. Tato reprezentace je tedy velmi naivní a v moderních detektorech se téměř nevyužívá pro svou neschopnost vyrovnat se s částečným zakrytím objetu, zhoršenými světelnými podmínkami, pootočením, deformací objektu nebo růzností měřítek, v nichž se objekt může vyskytovat.

Vyvstává tedy otázka, jak reprezentovat objekt tak, aby tato reprezentace byla vzhledem k výše popsanému invariantní a zároveň byla dostatečně dobrým popisem jeho typického vzhledu.

2.3.1.1 Globální gradienty a textura

Namísto záznamu prostých intenzit jednotlivých pixelů je v tomto přístupu využito filtrů pro detekci hran objektů. Vychází se z předpokladu, že objekty jsou snáze popsatelné skrze svůj tvar, obrysovou siluetu případně typickou texturu povrchu, mají-li nějakou. Zájem se tedy od doslovného popisu vzhledu přesouvá spíše k nalezení hran skrze změny intenzit a směrů, v nichž lze tyto změny v rámci objektu pozorovat.

Hlavní výhodou tohoto přístupu je necitlivost ke změnám světelných podmínek a částečně i k natočení objetu, měřítku a zákrytu. Dobrým příkladem v současnosti asi nejpoužívanějších deskriptorů vzhledu objektů založených na této technice jsou Histogramy orientovaných gradientů – HOG (*Histogram of Oriented Gradients*)[7], deskriptor SIFT (*Scale-invariant Feature Transform*)[8] nebo jeho, z hlediska rychlosti výpočtu a porovnávání, efektivnější verze SURF (*Speeded Up Robust Feature*) [9].

2.3.1.2 Histogramy orientovaných gradientů

Globální deskriptor poprvé použitý v [7] pro detekci chodců, leze s úspěchem použít i pro reprezentaci jiných objektů [10]. Deskriptor je založen na rozdělení zájmové oblasti na několik málo

stejně velkých buněk (pokrývajících stejný počet pixelů) a výpočtu velikostí a orientací změn intenzit pixelů (gradientů) uvnitř těchto buněk.

Například v úloze detekce chodců byl způsob vytvoření tohoto deskriptoru po mnoha testech na určení jednotlivých parametrů a rozměrů ustálen na následujícím postupu:

1. vypočtou se velikosti a směry gradientů na oblasti o rozměrech 64×128 pixelů (typický poměr stran pro vzpřímeného člověka)
2. výpočet histogramů uvnitř buněk o velikosti 8×8 pixelů – to je 8×16 buněk v celé oblasti
3. Normalizace histogramů na navzájem se překrývajících blocích buněk, jeden blok má typicky rozměr 2×2 buňky – to je 7×15 bloků v oblasti zájmu
4. Sloučení histogramů do jednoho vektoru

Výpočet velikosti gradientů probíhá nejčastěji pomocí konvoluce intenzit dvou sousedních pixelů pixelu $p(x, y)$, v němž je gradient počítán, a filtrů g:

$$p(x, y)$$

$$g_x = [-1\ 0\ 1]$$

$$g_y = [-1\ 0\ 1]^{\mathrm{T}}$$

Pak lze pomocí následujícího vzorce spočítat první parciální derivace intenzity obrazu I na $p(x, y)$ ve vertikálním a horizontálním směru.

$$\frac{\delta I}{\delta x} \approx I(x+1, y) - I(x-1, y)$$

$$\frac{\delta I}{\delta y} \approx I(x, y+1) - I(x, y-1)$$

Velikost gradientu je pak:

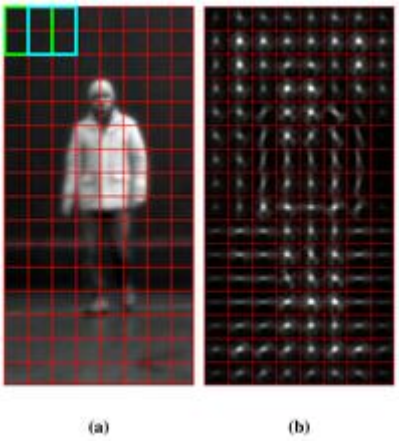
$$|\Delta I| = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$$

A směr gradientu lze vypočítat jako:

$$\theta = \tan^{-1}\left(\frac{\frac{\partial I}{\partial x}}{\frac{\partial I}{\partial y}}\right)$$

Nyní se pro každou buňku o velikosti 8 × 8 pixelů, na základě velikosti a orientace spočtených gradientů, vytvoří histogram s 9 biny o velikosti 20° (dohromady pokrývajících úhel 180°), pro které jednotlivé gradienty spočtené pro těchto 64 pixelů hlasují s ohledem na svou velikost a orientaci. Pokud je orientace některého z gradientů na pomezí dvou sousedících úhlů, reprezentovaných biny, je velikost takové gradientu mezi tyto dva sousední biny proporcionálně rozdělena podle vzdálenosti úhlu orientace gradientu od středů těchto binů.

Výsledkem je 16 × 8 = 128 histogramů pokrývajících celou oblast zájmu o velikost 128 × 64 pixelů. Pro představu je na následujícím obrázku znázorněn globální histogram orientovaných gradientů pro ukázkový obrázek člověka.



Obrázek. 1

Obrázek 1(a) ukazuje rozdělení zájmové oblasti na 8 × 16 buněk po 16 × 16 pixelech a z 50% překrývající se bloky o velikosti 2 × 2 buňky. Na obrázku 1(b) jsou pak pro každou z buněk zobrazeny příslušné histogramy orientace gradientů intenzity pixelů v těchto buňkách. Zdroj: [7]

Pro všechny navzájem se překrývající bloky (2 × 2 buňky, tj. 4 histogramy) je nyní provedena normalizace kontrastu v rámci jednoho bloku. Normalizaci lze provést například pomocí upravené L_2-*norm*.

$$f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}}$$

Kde f je výsledný normalizovaný vektor, v vektor původní, tedy sjednocení 4 vektorů ze čtyř buněk bloku, $\|v\|_2$ je euklidovská velikost (magnituda) vektoru a e nějaké velmi malé kladné číslo, které přičítáme, abychom se vyhnuli dělení nulou. Za povšimnutí stojí především to, že pro každou buňku je normalizace počítána čtyřikrát vzhledem k tomu, že bloky o velikosti 2 × 2 buňky se z 50% překrývají.

Normalizace tak pro každou buňku vyjde čtyřikrát jinak, všechny tyto varianty jsou ukládány do výsledného deskriptoru, což dále zvětšuje robustnost vůči změnám světelných podmínek a natočení objektu.

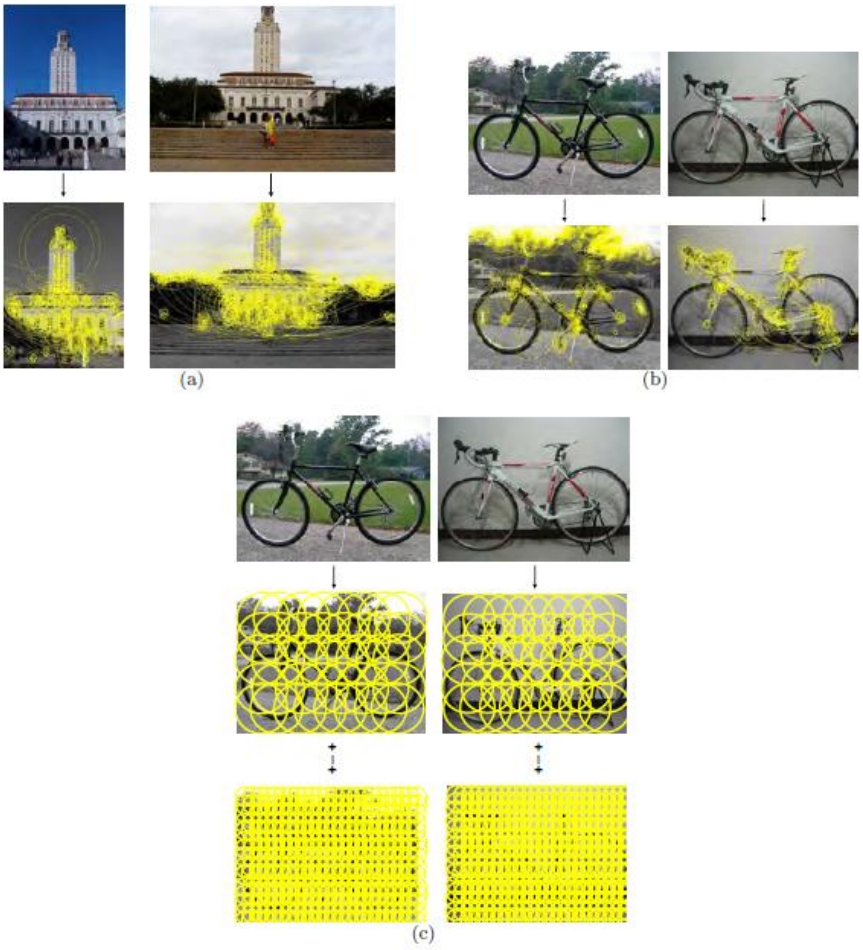
Konečným výsledkem je tedy vektor o velikost 128(histogramů) × 4(každá buňka se vyskytuje čtyřikrát kvůli normalizaci kontrastu přes překrývající se bloky) × 9(binů, rozsahů úhlů) = 4 608.

2.3.1.3 SIFT deskriptor s hustým vzorkováním

Podobně jako HOG deskriptor i SIFT deskriptory využívají velikostí gradientů a jejich orientací. Na rozdíl od klasického histogramu orientovaných gradientů se však SIFT používá typicky v kombinaci s jedním z detektorů význačných bodů [8], ve kterých se následně vypočítá. Není ho tedy třeba počítat pro celou zájmovou oblast. Tento přístup je však vhodnější při řešení úloh na detekci specifických objektů nebo při skládání 3D obrazu ze stero kamery spíše než v úlohách rozpoznávání generických objektů nějaké třídy. Pramení to z faktu, že u různých objektů stejné třídy nemusí detektory význačných bodů nalézt stejné body, což je naopak pravda pro jeden specifický objekt, přestože v různých obrázcích, pod různými úhly a na jiných měřítcích. Další nevýhodou je zjevný fakt, že při hustším vzorkování je oproti vzorkování pouze na význačných bodech získáno mnohem více obrazové informace o celkovém typickém vzhledu objektu. Jednou z hlavních nevýhod hustšího vzorkování je naopak mnohem větší výpočetní náročnost a do určité míry také ztráta necitlivosti k změnám měřítka a rotaci.

Přesto je SIFT deskriptor v úlohách detekce generických objektů a jejich klasifikace hojně využíván a to právě v kombinaci s hustým vzorkováním, např. v [11].

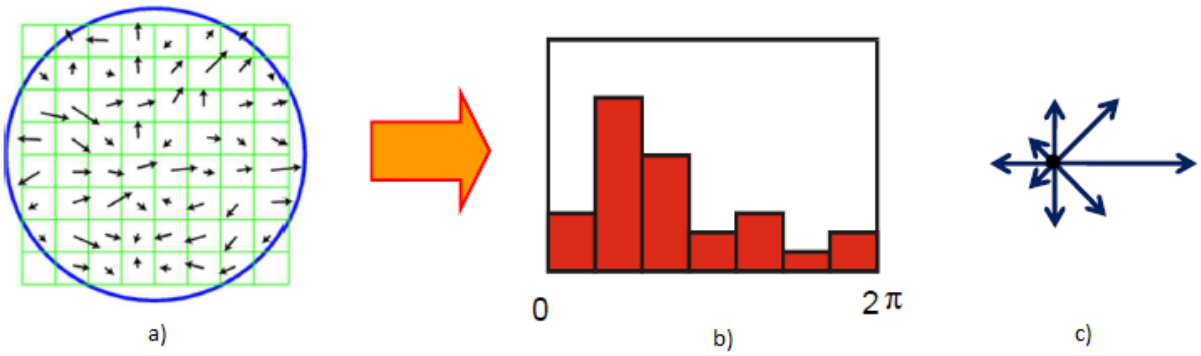
Hezkým příkladem výše popsaného je následující obrázek (Obrázek 2).



Obrázek 2

Na obrázku 2(a) jsou význačné body, z nichž bude počítán deskriptor SIFT, vzhledem k tomu, že se jedná o stejnou budovu, detekovány na přibližně stejných místech. Naopak obrázek 2(b) ukazuje, jak i dva velmi podobné objekty ze stejné kategorie mohou produkovat poměrně rozdílné význačné body. Obrázek 2(c) pak znázorňuje husté vzorkování pro výpočet SIFT nebo jiného deskriptoru. Zdroj: [32]

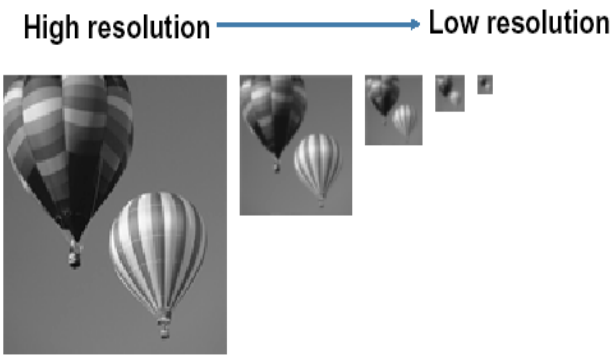
Podobně jako při výpočtu HOG deskriptoru se okolí detekovaného význačného bodu, nebo, to při hustém vzorkování, každého pixelu obrazu zahrnutého do vzorkování, rozdělí pomocí pravidelné čtvercové mřížky, v níž jsou na blocích o velikosti 4×4 každý o rozměru 4×4 pixely vypočítány histogramy gradientů s osmi úhlovými rozsahy (biny), do kterých každý pixel hlasuje na základě velikosti na něm spočteného gradientu a blízkosti velikosti úhlu orientace tohoto gradientu ke středům dvou nejbližších binů (úhlových rozsahů) histogramu (Obrázek 3). Postup normalizace vektorů je pak stejný jako u histogramů orientovaných gradientů

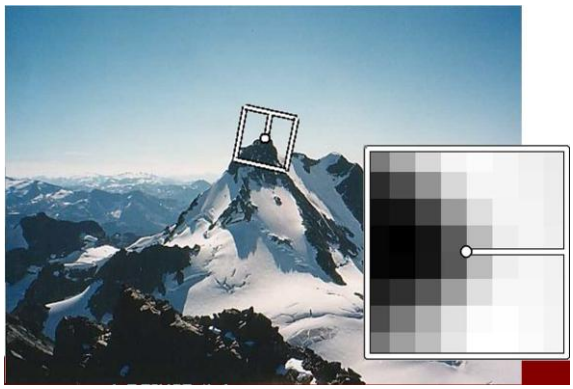


Obrázek 3

Obrázek 3 a) symbolicky znázorňuje směr a velikost vypočtených gradientů v místech jednotlivých pixelů na bloku velikostu 8×8 pixelů. Na obrázku 3 b) je znázorněn histogram, do kterého jednotlivé pixely volí a v obrázku 3 c) je histogram ve formě směrové hvězdice ukazující dominantní směr gradientů pixelů bloku, kterému tento histogram náleží. Zdroj: [8]

Hlavním rozdílem oproti HOG deskriptoru je zvýšená odolnost SIFT deskriptoru proti změně měřítka a rotaci objektu. Tato snížená citlivost je dána strategií výběru bodů, na kterých je deskriptor počítán. Při hustém vzorkování jsou deskriptory počítány pro jeden bod na hned několika měřítcích najednou, většinou pomocí obrazové pyramidy. Lze toho dosáhnout opakovanou konvolucí obrazu s Gaussovským filtrem, která rozmaže obraz a sníží tak jeho rozlišení, následným vynecháním každého druhého pixelu ve výsledném obraze získáme pyramidu původních obrazů vždy zmenšenou o $1/2$ (Obrázek 4). Necitlivost k rotaci je zajištěna skrze určení dominantní orientace gradientu v každém bodě, ve kterém se bude deskriptor počítat. Výběr okolí tohoto bodu je pak upraven tak, aby toto okolí bylo natočeno ve směru dominantní orientace gradientu (Obrázek 5).





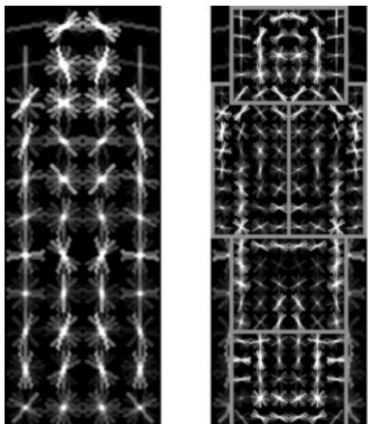
Obrázek 4

Obrázek 5

Obrázek 4 znázorňuje pyramidu různých měřítek obrázku. Obrázek 5 pak znázorňuje nalezení dominantní orientace gradientu na daném pixelu a pootočení oblasti pro výpočet SIFT deskriptoru v daném směru. Zdroj: [31]

2.3.2 Reprezentace založená na částech

Nejen vzhled, ale i rozložení sobě podobných částí, z nichž se skládá hledaný objekt, v rámci obrazu do jisté míry vykazuje podobnost nebo lze alespoň pozorovat několik málo konfigurací těchto částí, které se na objektu dané třídy opakují. K zaznamenání těchto konfigurací nebo vztahů mezi polohami dílčích částí objektu slouží reprezentace založená na částech. Vzhled částí pak lze stejně jako u globální reprezentace kódovat například SIFT nebo SURF deskriptory. V tomto případě však vzorkovanými jednotlivě právě pouze na těchto částech, kterým poté slouží jako vzhledové vzory, například agregované do takzvaných *bag-of-visual-words* [12], a ne plošně na celém objektu. Vizualizace rozdílu v těchto dvou přístupech při tvorbě deskriptoru viz obrázek 6.



Obrázek 6

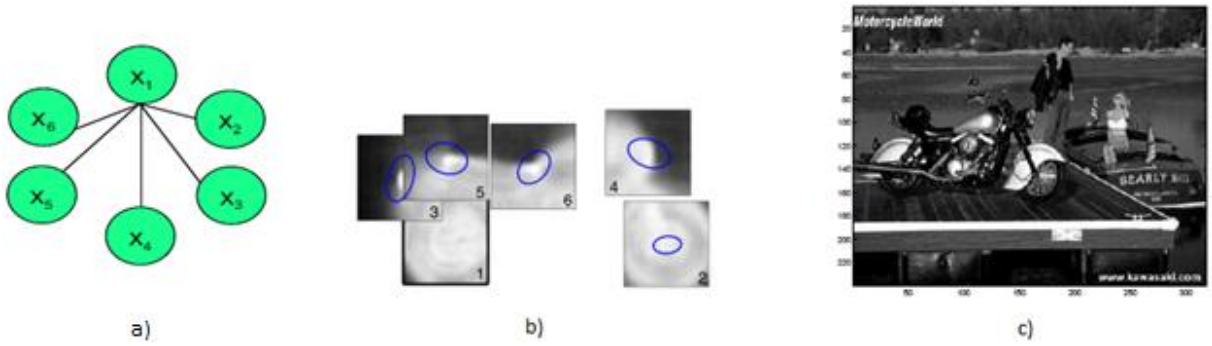
Obrázek 6: příklad vizualizace extrahovaných HOG deskriptorů pro celou postavu člověka (vlevo) a zvlášť pro jednotlivé její části (vpravo). Zdroj: [14]

Zřejmě poprvé se tato myšlenka v úloze rozpoznávání objektů objevila v práci [13], ve které je objekt složen z pevných částí spojených „pružinami“. Byla zde formulována funkce penalizující rozdíly ve vzhledu částí a napnutí pružin, skrze které byla modelována vzájemná konfigurace a vzdálenost mezi jednotlivými částmi.

Od té doby bylo navrženo mnoho dalších reprezentací objektů založených na částech. Dodnes nejpoužívanější a nejjednodušší z nich je hvězdicový model (*Star model*) používaný v mnoha variantách a zobecněních například do podoby stromu[14]. V tomto modelu je každá část objektu spojena, a tedy prostorově závislá, pouze s jeho „centrální“ částí.

2.3.2.1 Hvězdicový model

Tento model (Obrázek 7)reprezentuje objekty skrze vzhled jeho jednotlivých charakteristických částí a skrze příslušnosti těchto částí k centru objektu. Kromě vektoru deskriptoru kódujícího vzhled je tak ke každé části při konstrukci modelu přiřazen vektor směru a vzdálenosti ke středu objektu, ze kterého součást pochází a měřítko tohoto objektu. Při detekci pak v závislosti na zvolené strategii učení a detekce rozpoznané části volí pro společný střed a velikost pomocí vážených hlasů.



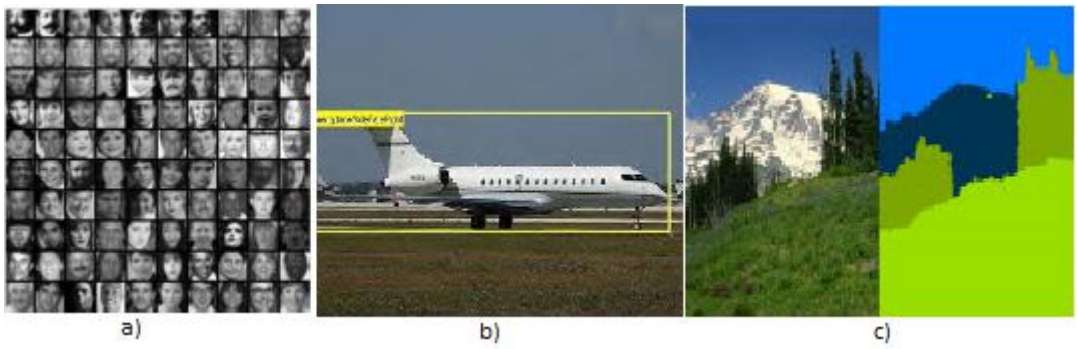
Obrázek 7

Obrázek 7 a) symbolicky znázorňuje hvězdicový model objektu, obrázek b) pak ukazuje příklad šesti na základě vzhledu rozpoznaných částí motocyklu, kde zadní kolo motocyklu je částí referenční, pro kterou zbylých pět hlasuje. Zdroj: [32]

Výhodou reprezentací založených na částech je jejich větší necitlivost k zákrytu hledaného objektu.

2.4 Klasifikátory a učení

Současné přístupy k učení a tvorbě modelů určených k rozpoznávání generických objektů daných tříd, většinou spoléhají na vizuální data popsaná štítky (*labels*) a alespoň částečně segmentovaná pomocí takzvaných obalových boxů (*bounding boxes*). Příklady označení a segmentace dat určených k natrénování modelů viz obrázek 8.



Obrázek 8

Obrázek 8 a) znázorňuje zarovnané a ostřihnuté tváře tak, aby vyplňovaly celou plochu trénovacích obrázků. Obrázek b) pak ve žlutém boxu obaluje objekt zájmu – letadlo – a částečně i jeho pozadí. Na obrázku c) pak lze vidět na segmentaci nejnáročnější anotaci zájmových objektů, kde stromy, hora, obloha i louka jsou do posledního pixelu od sebe odděleny pomocí různých barev. Zdroj: [32]

První zjevné dělení klasifikátorů založených na statistických modelech a z toho plynoucí výběr metody jejich učení je rozdělení na modely diskriminativní a generativní. Generativní modely se učí sdružené distribuce pravděpodobnosti ve formě $P(\boldsymbol{X}, k)$ kde $\boldsymbol{X}$ je vektor vstupních dat (deskriptorů) popisujících oblast obrazu testovanou na přítomnost hledaného objektu a $k = 1, \dots, K$ je vektor tříd, do nichž objekt může patřit včetně případu, že testované deskriptory patří pozadí. Diskriminativní modely naproti tomu modelují podmíněné distribuce pravděpodobnosti ve formě $P(k|\boldsymbol{X})$. Což z pohledu učení klasifikátorů pro rozpoznávání objektů nějaké kategorie interpretovat tak, že diskriminativní modely se učí přímo rozlišovat, zda testované deskriptory do dané třídy patří nebo ne na základě vzhledu deskriptorů – to znamená, snaží se najít co nejlepší dělicí čáru mezi pozitivními a negativními deskriptory, zatímco generativní klasifikátory po naučení modelují celou širokou variabilitu vzhledu kategorie hledaných objektů. Případně by se dalo říci, že generativní modely se snaží z trénovacích obrázků naučit všechny možné varianty trénovaného objektu a zároveň všechny možné variace pozadí („ne-objektu“), zatímco diskriminativní klasifikátor se bude snažit naučit nejcharakterističtější rozdíly mezi objektem a pozadím tak, aby se na jejich základě při testování

nového obrázku mohl přiklonit k jedné nebo druhé hypotéze (objekt-pozadí). Generativní klasifikátory tedy po naučení explicitně modelují skutečnou pravděpodobnostní distribuci naučené třídy, zatímco diskriminativní klasifikátory se učí modelovat hranici mezi třídami. Příkladem často používaného diskriminativního modelu nebo jeho variací je takzvaný *Support Vector Machine*[15]. Příkladem generativního modelu používaného v úlohách detekce objektů je kupříkladu Naivní Bayesovský model (*Naive Bayes*)[16].

Obecně se pro jakýkoliv typ modelu při jeho trénování používá pozitivních i negativních deskriptorů (dat). To jest takových, které pocházejí z objektů třídy, kterou se snažíme model naučit rozpoznávat (pozitivní) i z objektů mimo tuto třídu (negativní) – negativní deskriptory lze získat například vně obalového boxu, tak jak je zobrazen na obrázku 8 b), nebo na náhodném místě z obrázku, který nikde neobsahuje objekt zájmu. Oběma typům deskriptoru pak samozřejmě přiřadíme příslušnou anotaci – tedy zda náleží či nenáleží objektu trénované třídy.

2.4.1 Naivní Bayesovský klasifikátor

Jak vyplývá už z názvu, vychází tento klasifikátor ze známého Bayesova teorému. Během učení se snaží vystavět pravděpodobnostní model na popisu třídy skrze všechny k tomu dodané hodnoty deskriptorů popisující objekty na trénovacích obrázcích.

Mějme množinu uspořádaných dvojic $X = \{(\boldsymbol{X}_1, y_1), \dots, (\boldsymbol{X}_N y_N)\}$, kde $\boldsymbol{X} \in \mathbb{R}^D$ jsou deskriptory nebo skupiny různých deskriptorů v D-dimenzionálním prostoru a $y \in \{1, \dots, K\}$ jsou jim odpovídající třídy, které deskriptory popisují. Za předpokladu nezávislosti (proto naivní) komponent vektoru $\boldsymbol{X}$ lze z trénovacích dat následujícím způsobem vypočítat podmíněné pravděpodobnostní distribuce:

$$P(x_1, x_2, \dots, x_D | y = k) = \prod_{d=1}^{D} P(x_d | y = k)$$

Pomocí Bayesova teorému pak můžeme spočítat podmíněné posteriorní pravděpodobnostní distribuce:

$$P(y = k | x_1, x_2, \dots, x_D) = \frac{P(x_1, x_2, \dots, x_D | y = k) \times P(y = k)}{P(x_1, x_2, \dots, x_D)} = \frac{P(y = k) \times \prod_{d=1}^{D} P(x_d | y = k)}{P(x_1, x_2, \dots, x_D)}$$

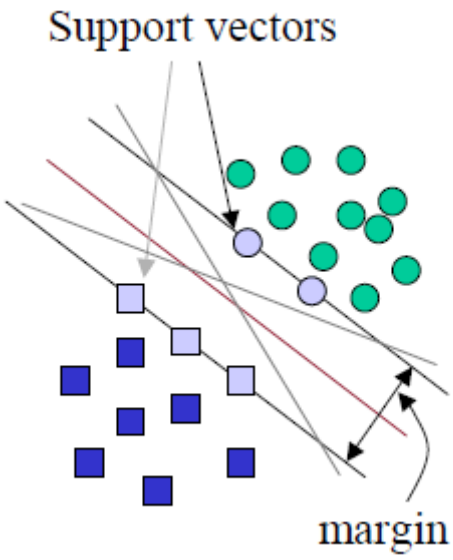
Pro nový vektor $\boldsymbol{X} = (x_1, x_2, \dots, x_D)$ je poté příslušnost k některé ze tříd, tedy hodnota Y, spočtena podle:

$$Y = \arg\max_c P(y = k|x_1, x_2, \dots, x_D) = \arg\max_c P(y = k) \times \prod_{d=1}^{D} P(x_d|y = k)$$

Přičemž pravděpodobnosti $P(y = k)$ a $P(x_d|y = k)$ jsou odhadnuty z trénovacích dat. Tato metoda je přes své hrubé předpoklady o vzájemné nezávislosti výskytu/ne-výskytu mezi jednotlivými hodnotami vektoru $\boldsymbol{X}$ často používaná v klasifikaci mnohodimensionálních dat pro svou rychlost a jednoduchost. Přes své zjednodušující předpoklady však podává i v komplexních úlohách detekce objektů velmi dobré výsledky [16].

2.4.2 Support Vector Machine

V případě skupiny algoritmů označených jako *Support Vector Machine* (dále jen SVM)se snažíme pravděpodobnostní model naučit strukturu dat tak, abychom je pomocí nějakého lineárního nebo nelineárního pravidla dokázali oddělit do vzájemně disjunktních množin (Obrázek 9).



Obrázek 9

Na obrázku 9 jsou symbolicky znázorněny tři nadroviny dělící tato data na dvě vzájemně disjunktní množiny - podprostory. Všechny dělí prostor korektně, pouze červená nadrovina však optimálně – maximalizuje vzdálenoti (margin) od bodů v obou podprostorech. Zdroj: [33]

Chceme tedy znát nějaké mapování z $X \rightarrow Y$, kde $x \in X$ je nějaký objekt a $y \in Y$ nějaká třída.

V jednoduchém příkladě se dvěma třídami (například hledaný objekt a pozadí) máme $x \in \mathbb{R}^D$ a $y \in \{\pm 1\}$ a množina trénovacích dat pak má podobu $X = \{(\boldsymbol{X}_1^D, y_1), \dots, (\boldsymbol{X}_N^D, y_N)\}$. Na této množině se snažíme najít takovou nadrovinu (*hyperplane*), abychom maximalizovali vzdálenost k nejbližším

bodům v obou poloprostorech. Funkce popisující tyto hadroviny jsou plně určeny takzvanými *Support Vectors* (podpůrné vektory), což jsou body z obou podprostorů, které jsou podél nadroviny nadrovině nejblíže (Obrázek 9), jsou to tedy takové body (data), které by posunuly nadrovinu, pokud bychom je z prostoru odstranily.

Rovnice nadroviny v dvourozměrném prostoru je vyjádřena jako $ax + by = c$, hledáme tedy taková a, b, c, aby pro všechny body v obrázku 9 platilo $ax + by \geq c$ pro všechny body označené kuličkami a $ax + by \leq c$ pro všechny body označené čtverečky. Takových možností je samozřejmě mnoho, dále tedy budeme brát v potaz pouze body vybarvené světle modře (kuličky i čtverečky) za současné podmínky, že máme na obě strany maximalizovat vzdálenost rovnice přímky od těchto bodů. Pomocí normálového vektoru w této nadroviny a skalárního součinu ▪ ji lze vyjádřit takto:

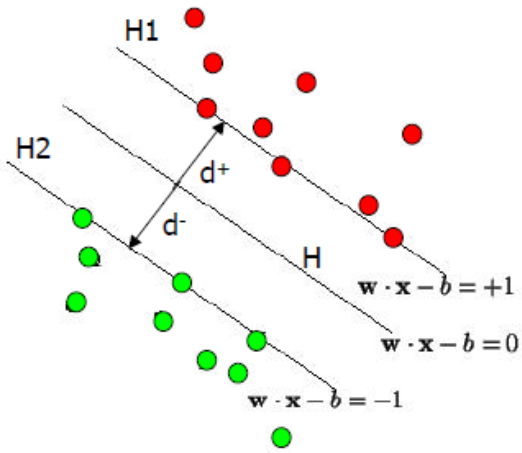
$$\text{H: } w \cdot x - b = 0$$

A sobě navzájem nejvzdálenější nadroviny (H1, H2) na nichž leží bod(y) z obou polorovin touto nadrovinou(H) rozdělených a zároveň k ní nejbližších (bodů k nadrovině H), tedy tzv. Support Vectors, vyjádříme v našem případě takto (Obrázek 10):

$$\text{H1: } w \cdot x - b = 1$$

a

$$\text{H2: } w \cdot x - b = -1$$



Obrázek 10

Na obrázku 10 jsou znázorněny Suppor Vectors pro pozitivní (červené body, $y_i = 1$) a negativní (zelené body, $y_i = -1$) data, jimi procházející roviny H1, H2 a minimální vzdálenosti od dělící nadroviny H k nejbližšímu negativnímu (vzdálenost d-) a pozitivnímu (vzdálenost d+) bodu. Zdroj: [33]

Vzdálenost H1 a H2 od počátku soustavy souřadnic je rovna $\frac{|-1-b|}{\|w\|}$ respektive $\frac{|1-b|}{\|w\|}$, kde $\|w\|$ je Euklidovská norma jejich normálových vektorů. Vzdálenost mezi H1 a H nebo H2 a H je tedy $\frac{1}{\|w\|}$ a vzdálenost mezi H1 a H2 jednoduše bude $\frac{2}{\|w\|}$.

Za předpokladu, že lze data takto lineárně separovat, je k jejich separaci nutné nalézt takové dvě rovnoběžné nadroviny H1 a H2, aby mezi nimi neležel žádný z bodů, které se snažíme oddělit a zároveň od sebe byly co nejdále. To lze zajistit minimalizací $\|w\|$ a splněním následujících dvou podmínek:

$$w \cdot x_i + b \leq -1$$

Pro $y_i = -1$ a

$$w \cdot x_i + b \geq +1$$

Pro $y_i = +1$, po úpravě lze zkombinovat na:

$$y_i(w \cdot x_i + b) - 1 \geq 0, \qquad \forall i$$

Kvůli snazšímu výpočtu minimalizace $\|w\|$ (obsahuje odmocninu) je výraz nahrazen $\frac{\|w\|^2}{2}$, čímž získáme k výpočtu vhodnější formu, která se zároveň numericky rovná původní.

Zde se dá šikovně využít Lagrangeových multiplikátorů, sloužících k nalezení extrému funkce za nějakého fixního daného omezení. Multiplikátory nám zde poslouží k nalezení minima $\frac{\|w\|^2}{2}$ bez nutnosti explicitně řešit $y_i(w \cdot x_i + b) - 1 \geq 0$. Kombinací těchto funkcí a převedením problému na hledání řešení pro:

$$\nabla \frac{\|w\|^2}{2} = \nabla \lambda_i (y_i(w \cdot x_i + b) - 1)$$

Respektive skrze pomocnou funkci L:

$$L = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \lambda_i (y_i(w \cdot x_i + b) - 1)$$

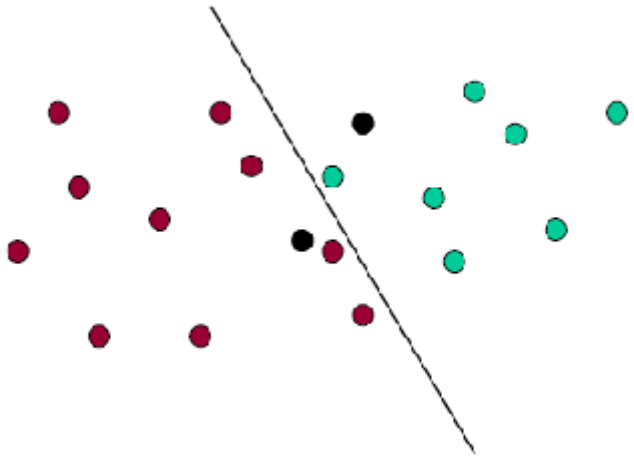
Jejíž derivací položenou rovnu nule získáme výsledný w:

$$w = \sum_{i=1}^{N} \lambda_i y_i x_i$$

Body x_i, jejichž korespondující multiplikátory λ_i nabývají hodnoty větší než nula, odpovídají bodům určujícím nadroviny H1 a H2 a leží tedy na $y_i(w \cdot x_i + b) = 1$.

Pro lineárně neseparovatelná data je situace poněkud složitější (Obrázek 11). Buď je třeba zavést penalizaci za špatně separovaná data a tu minimalizovat – tedy najít kompromis mezi maximalizací vzdálenost H1 a H2 a minimalizací této penalizace. Podmínka je pak upravena do podoby.

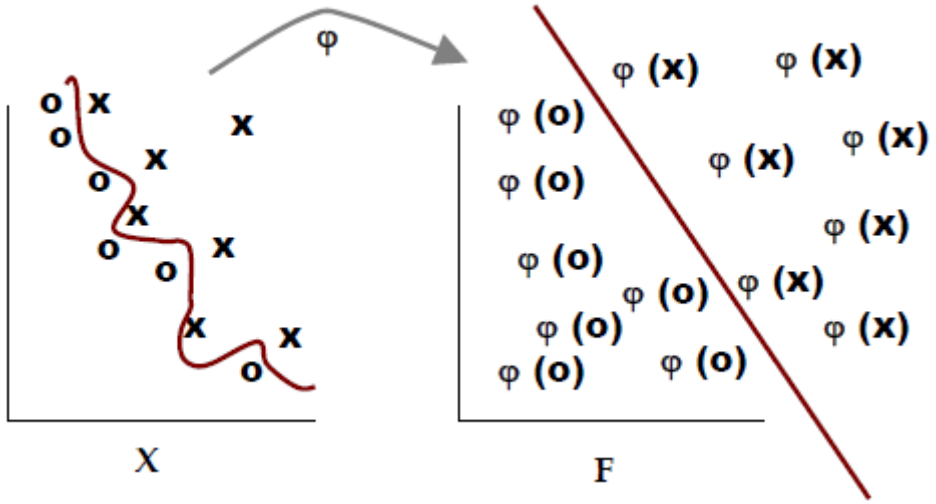
$$y_i(w \cdot x_i + b) - 1 + \mathcal{E}_i \geq 0$$



Obrázek 11

Černé body budou k podmínce $y_i(w \cdot x_i + b) - 1 + \mathcal{E}_i \geq 0$ *přičítat zápornou penalizaci, protože jsou na „špatné straně" nadroviny. Zdroj: [33]*

Nebo se pokusit dvourozměrná data namapovat do vícerozměrného prostoru pomocí nějaké transformace φ (Obrázek 12).



Obrázek 12

Symbolická ukázka transformace φ z prostoru X do F. Zdroj: [33]

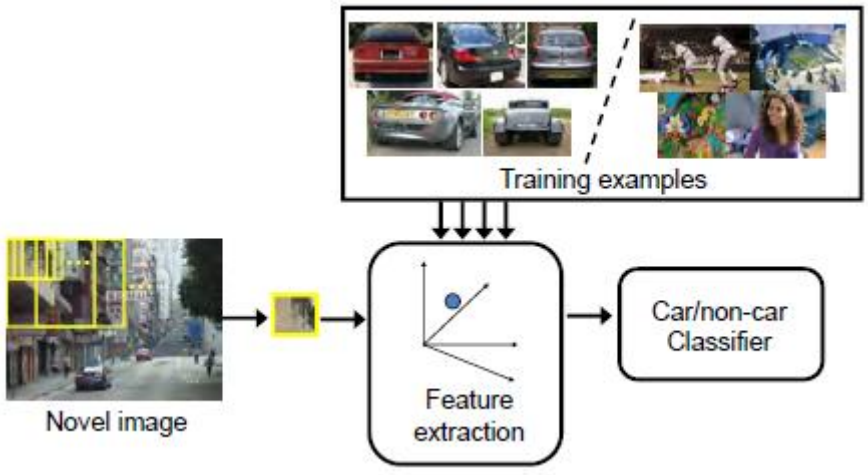
Problematika SVM je nepoměrně komplexnější, její podrobnější a rozsáhlejší vysvětlení však přesahuje rámec této práce. Více k ní například v [17].

2.5 Detekce generických objektů

Poté co extrahujeme z obrazu příslušné vlastnosti našeho objektu zájmu a natrénujeme na ně námi zvolený pravděpodobnostní model, je třeba zvolit strategii detekce nových objektů z námi naučené třídy v nových, modelem dosud neviděných, obrazech. Existují opět v zásadě dva způsoby jak k detekci přistupovat. V prvním případě jde o detekci pomocí posuvných oken, kde je objekt reprezentován skrze svůj celostní vzhled pomocí globálních deskriptorů a je tedy třeba mít takové deskriptory a klasifikátory, které dokážou zachytit vzhled a tvar objektů dané třídy v jejich široké variabilitě a dostatečně je skrze ni zobecnit. Vyžaduje to také často na natrénování i aplikaci náročné nelineární klasifikátory. Naproti tomu druhý přístup k detekci pomocí modelů založených na specifických částech objektů přenáší váhu detekce od reprezentace objektů a učení směrem ke komplexnějším algoritmům vyhledávání objektů v prostoru možných konfigurací objektů reprezentujících jejich variabilitu.

2.5.1 Detekce pomocí posuvných oken

Při detekci pomocí posuvných oken jde o problém klasifikace obsahu obrazu, který se nachází ve výseči (podmnožině) tohoto obrazu určené právě rozměry a tvarem jakéhosi okna, které po obraze posouváme a na základě stejných vlastností (stejných typů deskriptoru), které jsme vypočetli pro natrénování našeho rozhodovacího modelu, pod tímto oknem pomocí tohoto modelu testujeme obraz na přítomnost objektu námi naučené třídy (Obrázek 13).



Obrázek 12

Hlavní komponenty detektoru využívajícího posuvu oken po testovaném obraze. Zdroj: [32]

Abychom byli pomocí toho přístupu schopni detekovat objekty různých měřítek, je třeba vytvořit si z testovaného obrázku obrazovou pyramidu měřítek, ve kterých nás případný objekt zajímá, např. pomocí gaussovského filtru – viz podkapitola o deskriptorech. Okna jsou tedy posouvána po celé ploše obrázků na všech úrovních pyramidy a výstupní hodnoty klasifikátoru jsou na každém místě uloženy.

Výstupem klasifikátoru na takovém okně je typicky míra shody obrazu pod oknem s naučeným vzhledem hledaného objektu pixel po pixelu normovaná na plochu jejich sjednocení. Obecně se pak takový rozsah pohybuje od 0 pro „pozadí“ až po 1 pro objekt přesně odpovídající naučenému vzhledu. V závislosti na druhu aplikace je možné posouvat hranici pro pozitivní detekci mezi 0 a 1 tak, abychom měli více špatně pozitivně (*false positives* – hranice je nastavena blíže 0)nebo více špatně negativně (*false negatives* – hranice je nastavena blíže 1) určených objektů.

Během let se pro výpočetní náročnost takového přístupu přišlo s několika vylepšeními, která mají za úkol snížit potřebu počítat deskriptory v každém bodě obrazu. Pokud například hledáme chodce v obraze, má smysl začít s prohledáváním, tedy posouváním okna, uprostřed obrazu anebo pomocí řady klasifikátorů postupovat od jednodušších k složitějším modelům, kdy ty složitější a tedy náročnější na výpočetní výkon jsou použity pouze v případě, že řada slabších klasifikátorů před nimi danou oblast nedokázaly vyloučit [2]. Další strategie využívají naopak kontextu, ve kterém se hledané objekty běžně vyskytují [18]. Zde se pomocí takzvaných globálních GIST deskriptorů [19] určí charakter scény, například zda se jedná o pohled ulicí, les, uzavřenou místnost atp. a na základě této charakteristiky se přistupuje k hledání objektu zájmu. Například pokud se díváme dolů ulicí, je nepravděpodobné, že se chodec bude nacházet v levém horním rohu obrazu, případně je nesmyslné hledat krávu v kanceláři (Obrázek 13).

Obrázek 12

Zapojením GIST deskriptorů bychom zjistili, že se jedná o pohled, který je vlevo nahoře uzavřen a naopak se otevírá z levého dolního rohu směrem do pravého horního. GIST deskriptory by tak umožnily vynechat prohledávání uzavřených oblastí a soustředit se pouze na ty části scény, kde je výskyt chodce možný a to dokonce pouze v příslušných měřítcích v závislosti na vzdálenosti od objektivu kamery. Zdroj: [32]

S šikovně zvolenými deskriptory je přístup k detekci za pomoci posuvu okna obrazem pro určitý typ aplikací poměrně dobrou volbou. Především je díky tom, že hledá po celém obraze, v různých natočeních a měřítcích, velmi spolehlivý pokud jde o hledání objektů, jež mají prostorově přibližně neměnnou strukturu (chodec, obličej) a je poměrně snadný na implementaci, ať už jde o učení nebo detekci, vzhledem k tomu, že pracuje s globálními deskriptory objektů. Na druhou stranu má však také spousty nevýhod. Především velké nároky na výpočetní výkon, dále ne každý objekt má alespoň přibližně hranatý tvar, což znamená, že pro nehranaté objety je do deskriptorů okna nutné zanést velké množství plochy pozadí. Další nevýhodou je, že jde vlastně o binární (hledaný objekt × pozadí) rozhodování a tak je pro detektory, které by chtěly rozpoznávat objekty z více tříd najednou, nutné zavést pro každou třídu vlastní klasifikátor nebo jejich skupinu.

2.5.2 Detekce pomocí částí

Strategie detekce objektů pomocí jejich částí přináší výhodu především ve snížené citlivosti k zákrytu objektu a v lepší práci s variabilitou co se týče jeho prostorové konfigurace. Jedním z dobrých příkladů této strategie je Obecná Houghova transformace [20] použitá i v praktické části této práce.

2.5.2.1 Houghova transformace

Teoretickým východiskem pro obecnou Houghovu transformaci je standardní Houghova transformace [21, 22] definující vztah mezi 2D obrazem nějakého geometrického tvaru (přímka, kružnice, elipsy… atp.) a parametry tento tvar definujícími. Každý takový tvar totiž lze vyjádřit pomocí obecné parametrické rovnice:

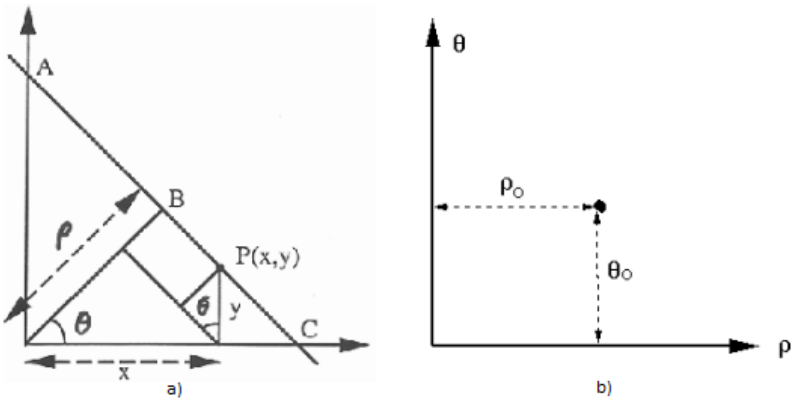
$$f(x, y, \alpha_1, \ldots, \alpha_n) = 0$$

Jde tedy o transformaci reprezentace tvaru z dvourozměrného obrazu (jednotlivé body x, y) do n-D dimensionálního prostoru parametrů, příp. naopak.

Mějme například přímku v dvourozměrném prostoru vyjádřenou parametrickou rovnicí:

$$f(x, y, \rho_0, \theta_0) = x \cos\theta_0 + y \sin\theta_0 - \rho_0 = 0$$

Kde ρ_0 je délka normály této přímky od počátku soustavy souřadnic a $\theta_0$úhel normály jež svírá s osou x. Takovou přímku lze pak v Houghově prostoru zobrazit bodem odpovídajícím souřadnicím na (ρ_0, θ_0) (Obrázek 13).



Obrázek 13

Obrázek 13 a) zobrazuje přímku v dvourozměrném prostoru. Její odpovídající reprezentace v Houghově prostoru je pak vyobrazena bodem se souřadnicemi ρ_0, θ_0 na obrázku 13 b). Zdroj: [20]

Obdobně lze transformovat jakýkoli jednotlivý bod (x_0, y_0). Po vyřešení rovnice shora pro ρ (vzdálenost bodu x_0, y_0 od počátku soustavy souřadnic):

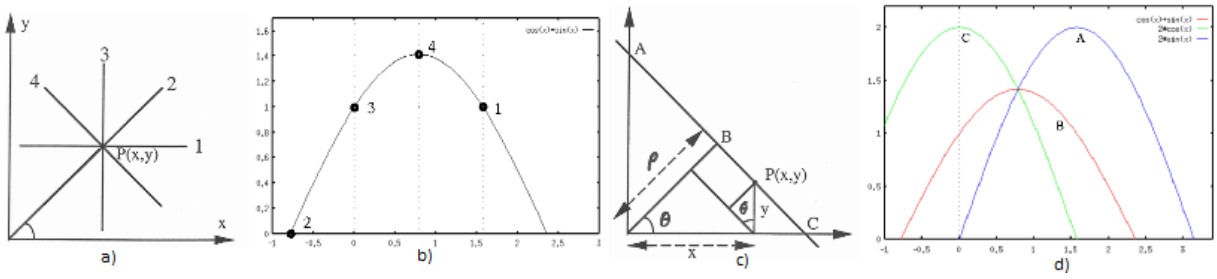
$$\rho: \quad x_0 \cos\theta + y_0 \sin\theta = \sqrt{x_0^2 + y_0^2}\left(\frac{x_0}{\sqrt{x_0^2 + y_0^2}}\cos\theta + \frac{y_0}{\sqrt{x_0^2 + y_0^2}}\sin\theta\right)$$
$$= r_0(\cos\alpha_0 \cos\theta + \sin\alpha_0 \sin\theta) = r_0 \cos(\alpha_0 - \theta)$$

Kde

$$r_0 = \sqrt{x_0^2 + y_0^2}$$

$$\alpha_0 = \tan^{-1}\left(\frac{y_0}{x_0}\right)$$

Dostaneme pro bod v dvourozměrném prostoru (x, y) sinusoidu v Houghově prostoru (Obrázek 14) – tedy pro parametry (ρ, θ)každé přímky, která tímto bodem prochází. A naopak každý bod na této sinusoidě pak odpovídá přímce v původním dvourozměrném obraze, na které bude ležet bod tvořící v Houghově prostoru právě tuto sinusoidu. Pokud na tyto sinusoidy pohlížíme jako na hlasy, pak lineární kombinace bodů v prostoru (x, y) vytvoří lokální maxima těchto hlasů v bodech, kde se sinusoidy jednotlivých lineáních bodů protínají. Každé takové lokální maximum určuje jednu přímku protínající všechny body, které pro toto maximum v (ρ, θ) prostoru hlasovaly (Obrázek 14).



Obrázek 14

Obrázek 14 a) zobrazuje přímky odpovídající bodům na sinusoidě z obrázku 14 b), všechny přímky procházejí stejným bodem $P(x, y)$, jemuž naopak odpovídá sinusoida, na které leží všechny body odpovídající v tomto bodě se protínajícím přímkám. v dvourozměrném prostoru. Obrázek 14 c) znázorňuje přímku s třemi na ní ležícími body A, B a C, jejichž sinusoidy a protnutí v prostoru (ρ, θ) je znázorněno na obrázku 14 d). Hodnota (ρ_0, θ_0), kde se sinusoidy A, B a C protínají pak určuje směr a velikost normály přímky na které body A, B a C leží v prostoru (x, y). Zdroj: [20]

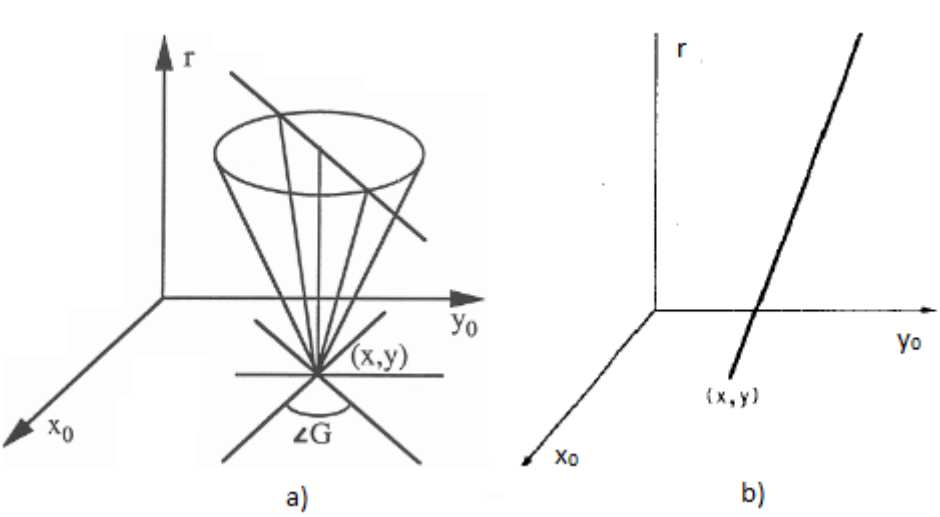
Podobné principy lze aplikovat i na jiné geometrické tvary, které lze popsat pomocí jejich parametrů. Jako je například kružnice, kde každý bod kružnice nabývá kónického tvaru v 3D Houghově prostoru (parametry x_0, y_0, r pro střed a průměr) nebo elipsa, kde se body promítají do objektů v 4D Houghově prostoru (parametry x_0, y_0, a, b pro střed a obě poloosy). Detekce takových tvarů v obraze pak probíhá podle následujícího algoritmu:

Pro kružnici:

$$f(x, y, x_0, y_0, r) = (x - x_0)^2 + (y - y_0)^2 - r^2 = 0$$

$$\frac{dy}{dx} = \frac{f'_x(x,y)}{f'_y(x,y)} = -\frac{x - x_0}{y - y_0} = -\tan\phi = -\tan\left(\Delta G \pm \frac{\pi}{2}\right) = \cot\Delta G = \frac{\cos\Delta G}{\sin\Delta G}$$

Geometricky lze tyto dvě rovnice pro nějaký na kružnici pevně zvolený bod (x, y) zobrazit jako kužel (první rovnice) jenž je ve své ose protínán rovinou specifikovanou druhou rovnicí (Obrázek 15 (a)), přičemž hodnota ΔG udává směr gradientu v bodě (x, y). Tj. směr kolmý na hranu, která je tímto bodem tvořena. Pro každý takovýto bod obrazu lze sestavit v Houghově prostoru (x_0, y_0, r) jeden kužel s tím, že kužely bodů, které leží na stejné kružnici, se v tomto prostru budou protínat v jednom bodě (x_k, y_k, r_k), jež bude určovat parametry kružnice, na nichž leží. Navíc tím, že známe ΔG lze redukovat stupně volnosti 3D prostoru (x_0, y_0, r) a ušetřit tím paměť i výpočetní výkon. Z kuželu se tak pro každý bod stane přímka v původním Houghově 3D prostoru (Obrázek 15 (b)).



Obrázek 15

Na obrázku a) je kónická reprezentace bodu kružnice v Houghově (x_0, y_0, r)prostoru.Obrázek b) pak demonstruje reprezentaci stejného bodu po redukci stupňů volnosti ze 3 na 1 pomocí určení směru gradientu v tomto bodě. Zdroj: [20]

Řešením těchto dvou rovnic pro x_0 a y_0 získáme:

$$x_0 = x \pm r\cos\ \Delta G$$

$$y_0 = x \pm r\sin\ \Delta G$$

Nyní pro všechny body (pixely) v obraze kde $|G(x,y)| > T_G$;

Pro všechny r

{

$$x_0 = x \pm r \cos \Delta G$$

$$y_0 = y \pm r \sin \Delta G$$

$$H(x_0, y_0, r) = H(x_0, y_0, r) + 1$$

}

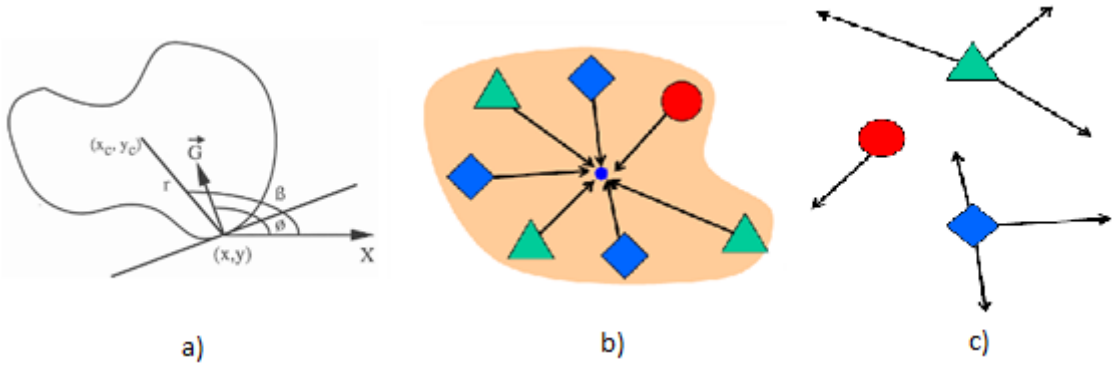
Pro všechny body Houghova prostoru parametrů, které splňují $H(x_0, y_0, r) > T_H$, pak existuje v obraze kružnice s průměrem r a středem (x_0, y_0).

Zobecněná Houghova transformace naproti tomu nevyžaduje, aby transformovaná křivka byla popsatelná parametrickou rovnicí, a umožňuje tak detekci libovolných tvarů nebo i celých objektů na základě nalezení částí odpovídajících příslušným vzorům.

2.5.2.2 Obecná Houghova transformace

Za použití stejné myšlenky, jaká se uplatňuje v detekci parametrických křivek pomocí Houghovi transformace, lze detekovat i takové tvary, které nedokážeme popsat parametrickou rovnicí nebo dokonce i celé objekty, jež rozdělíme na jejich charakteristické části, z nichž je lze poté zpětně složit [20].

V libovolné křivce nebo objektu, které budeme hledat, si určíme referenční bod x(Obrázek 16). Spočteme směr gradientů v bodech, jež tuto křivku tvoří. Tyto hodnoty a vektory určující směr od bodů ležících na křivce k vybranému bodu referenčnímu zaznamenáme do tzv. R-tabulky.



Obrázek 16

Na obrázku a) je parametrickou rovnicí nepopsatelná křivka s vyznačeným referenčním bodem (x_c, y_c), bodem ležícím na křivce(x, y), směrovým vektorem r od tohoto bodu k bodu referenčnímu kónická reprezentace, úhlem β, který tento vektor svírá s osou x a úhlem $\emptyset$ jenž svírá gradient $\vec{G}$ s vodorovnou osou. Na obrázku b) je schematicky zobrazen objekt složený ze třech vzájemně různých částí a jejich referenční bod v objektu. Na obrázku c) je pak zobrazen model objektu z obrázku b) reprezentovaný těmito třemi různými částmi a všemi přípustnými vektory určujícími jejich možnou polohu vzhledem k referenčnímu bodu. Zdroj: [20]

Tato R-tabulka (Tabulka 1) se sestává právě z vektorů určujících směr k referenčnímu bodu křivky a směrů gradientů jednotlivých bodů, z nichž se křivka skládá. Tabulka je indexována právě podle úhlů gradientů tak, že pro každý index lze zaznamenat více možných vektorů k referenčnímu bodu. Na základě těchto údajů jsou pak modelovány všechny možné výskyty křivky v testovaném obraze v určité konfiguraci (měřítku a natočení) a místě.

$\phi_1 = 0$	$(r,\beta)_{1_1}$	$(r,\beta)_{1_2}$	$\cdots$	$(r,\beta)_{1_{n_1}}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
ϕ_j	$(r,\beta)_{j_1}$	$(r,\beta)_{j_2}$	$\cdots$	$(r,\beta)_{j_{n_1}}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$\phi_k = \pi$	$(r,\beta)_{k_1}$	$(r,\beta)_{k_2}$	$\cdots$	$(r,\beta)_{k_{n_1}}$

Tabulka 1

Tabulka úhlů $\emptyset$, jež svírají gradienty bodů na křivce s vodorovnou osou a jim odpovídající vektory (velikost r a úhel β) směřující od těchto bodů k referenčnímu bodu. Zdroj: [20]

Detekce křivky pak probíhá následujícím způsobem: Pro každý bod (x, y) testovaného obrazu s gradientem větším než nějaká dolní hranice pro hranu $|G(x_c, y_c)| > T_G$ na základě úhlu $\emptyset$, který jeho gradient svírá s vodorovnou osou, urči nejbližší řádek podle hodnot prvního sloupce a pro všechny uspořádané dvojice (r, β) v tomto řádku spočti možné výskyty referenčního bodu.

$$x_c = x \pm r \cos \beta$$

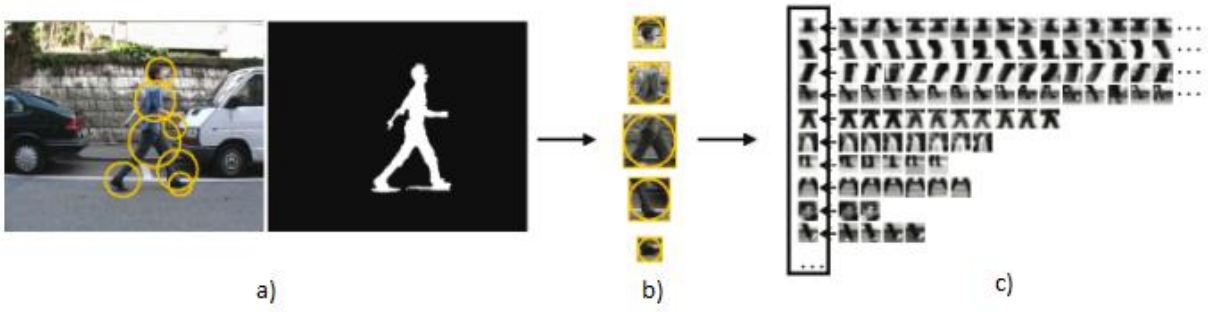
$$y_c = y \pm r \sin \beta$$

Pro každý bod $(\mathrm{x_c}, \mathrm{y}_c)$ pak zvětši hodnotu akumulátoru Houghova prostoru $H(x_c, y_c)$ o jedna.

$$H(x_c, y_c) = H(x_c, y_c) + 1$$

Body $(\mathrm{x_c}, \mathrm{y}_c)$ pro které platí, že $H(x_c, y_c) > T_h$ jsou pak referenční body všech nalezených křivek.

Pokud jde o detekci objektů, jejichž přesnou podobu neznáme, je třeba obecnou Houghovu transformaci dále upravit. Tato varianta transformace se poprvé objevila v [23]. Prvně je třeba vytvořit si jakýsi slovník typických částí objektu (Obrázek 17).



Obrázek 17

Obrázek a) zobrazuje identifikaci typických částí objektu, z nichž se po extrakci příznaků (obrázek (b)) vytvoří slovník obrazových slov – bag-of-words (obrázek (c)). Zdroj: [32]

Tyto části jsou z trénovacích obrázků extrahovány společně s vektory určujícími jejich polohu vzhledem k referenčnímu bodu objektu – typicky střed objektu. Jsou tedy určeny svým vzhledem a tzv. offsetem (vektorem určujícím polohu vzhledem ke středu objektu). Po vytvoření clusterů sobě podobných částí, jež tvoří samotný slovník, tak dostáváme obdobu původní R-tabulky. Shluky (clustery) vizuálních slov lze získat například pomocí K-Means algoritmus [24].

2.5.2.2.1 K-means

Na počátku zvolíme k libovolných shluků na libovolných pozicích, neboli centrech c těchto shluků, v prostoru tvořeném shlukovanými vektory x – částmi objektů. Poté pomocí minimalizace následující funkce:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2$$

Přiřadíme každou část x k jejímu nejbližšímu centroidu c, čímž získáme první rozdělení. Nyní přepočítáme původních k hodnot středů c, tak aby odpovídaly těžištím nově vytvořených shluků, a postup opakujeme, dokud se tato těžiště nepřestanou v prostoru hýbat.

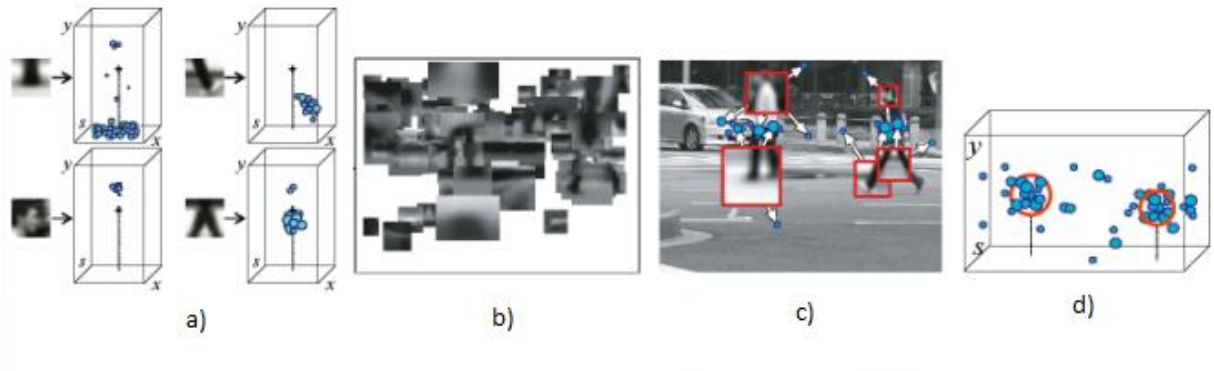
Výsledkem jsou tedy shluky sobě podobných částí a za každý shluk množina vektrů relativních pozic vzhledem ke středu objektu (Obrázek 18 (a)). V nové obrázku, testovaném na přítomnost hledaného objektu, pak porovnáváme podle zvolené strategie a míry všechny, nebo podle nějakého klíče zvolené, patche s těmito shluky a na základě vektorů shlukům přiřazených nejpodobnějšímu shluku tento patch hlasuje do Houghova prostoru na vektory (offsety) určené pozice (obrázek 18 (b), (c)).

Váha hlasu patche se vzhledem v a lokací l pro objekt o na pozici $\boldsymbol{x}$ v Houghově prostoru je určena následující distribucí pravděpodobnosti modelující nejistotu, zda patch objektu skutečně patří:

$$P(o, \boldsymbol{x}|v, l) = \sum_i p(o, \boldsymbol{x}|C_i, l)p(C_i|v)$$

Tato pravděpodobností interpretace váhy hlasu je asi největším rozdílem oproti původní zobecněné Houghově transformaci, kde jsme hledali křivky libovolných ale přesně určených tvarů, zatímco zde jde o hledání objektů na základě vzhledu pouze podobných bez znalosti jejich přesného tvaru nebo podoby.

Lokální maxima Houghova prostoru pak odpovídají středům objektů na daných pozicích (Obrázek 18 (d)).



Obrázek 18

Obrázek a) zobrazuje shluky a pozice, na nichž se části objektu tyto shluky tvořící mohou vyskytovat. Na obrázku b) a c) jsou pak vyobrazeny patche, jež by mohly patřit objektu na základě jejich porovnání se shluky, a vektory + velikost vážených pravděpodobnostních hlasů pro středy objektů. Obrázek d) symbolicky znázorňuje identifikovaná lokální maxima v Houghově prostoru. Zdroj: [32]

Kapitola 3

3.1 Houghův les

Houghův les, poprvé zmíněný v [6], spojuje myšlenku zobecněné Houghovy transformace, tedy její varianty použité v ISM[23], s náhodnými rozhodovacími stromy (*Random tree/forest*)[25].

Objekt je zde reprezentován pomocí takzvaných patchů. Patch $\mathcal{P}_i$ se skládá z uspořádané trojice $\mathcal{P}_i = (I_i, \boldsymbol{d}_i, y_i)$, pokud se jedná o patch popisující část objektu, nebo dvojice $\mathcal{P}_i = (I_i, y_i)$, pokud reprezentuje pozadí. I_i je vzhled patche (několik různých kanálů spočtených z barev RGB a odstínů šedi), $\boldsymbol{d}_i$ je vektor směřující od středu patche ke středu objektu a y_i pak určuje, zda patch patří objektu (např. $y_i = 1$) nebo pozadí (např. $y_i = 0$).

Les $\mathcal{F} = \{\mathcal{T}_t\}_{t=1}^{T}$ se obecně (Houghův i náhodný) skládá z množiny T stromů. Každý takový strom $\mathcal{T}_t$ se skládá z uzlů a listů, kde každý uzel se dělí na dvě větve zakončené opět buď dělící uzlem, nebo koncovým listem. Na rozdíl od již zmíněného ISM, který vytváří jakýsi slovník vizuálních slov využívajíce k tomu shlukování (viz. předchozí kapitola a [11]), vytváří Houghův les pro každý strom množinu hierarchicky uspořádaných testů, jež na základě vzhledu dělí patche do z hlediska pozice směrem k centru objektu a (ne)příslušnosti patche k objektu více či méně čistých množin patchů v listech těchto stromů.

Cílem je klasifikátor, který dokáže podle vzhledu I_i s co největší jistotou určit, zda patch $\mathcal{P}_i$ patří k objektu (tj. zda $y_i = 1$) a zároveň určit jeho polohu, respektive polohu středu objektu (tj. přiřadit mu vektor $\boldsymbol{d}_i$), ke kterému patří. Jde tedy o to dosáhnout při trénování stromů v každém z listů množiny patchů s co největší čistotou co do směru a velikosti vektorů ke středům objektů a co do příslušnosti k objektu/pozadí. Toho se dosahuje optimalizací testů v uzlech stromů při jejich učení. Tj. optimalizací rozdělení množiny patchů na vstupu uzlu na dvě co nejodlišnější podmnožiny. Hledáme takové parametry ø, které skrze ziskovou funkci $g(\emptyset, A_{node})$ optimalizují funkci $f_\emptyset$ dělící vstupní množinu trénovacích patchů uzlu $A_{node} = \{\mathcal{P}_i\}$ na podmnožiny A_L, A_R pro levou a pravou větev.

$$f_\emptyset(\mathcal{P}) \in \{0,1\}$$

$$A_L(\emptyset) = \{\mathcal{P} \in A_{node} | f_\emptyset(\mathcal{P}) = 0\}$$

$$A_R(\emptyset) = \{\mathcal{P} \in A_{node} | f_\emptyset(\mathcal{P}) = 1\}$$

$$g(\emptyset, A_{node}) = \mathcal{H}(A_{node}) - \sum_{S \in \{L,R\}} \frac{|A_S(\emptyset)|}{|A_{node}|} \mathcal{H}\big(A_S(\emptyset)\big)$$

Kde tvar funkce $\mathcal{H}(A)$ záleží na tom, zda optimalizujeme informační zisk různých rozdělení A_{node} podle směru a velikosti středových vektorů nebo podle příslušnosti k objektu. Hledáme tedy $\emptyset^{opt}$

$$\emptyset^{opt} = \arg\max_{\emptyset} g(\emptyset, A_{node})$$

Možností, jak vybrat $f_\emptyset(\mathcal{P})$ a $g(\emptyset, A_{node})$, resp. $\mathcal{H}(A_{node})$, je několik. Pro $f_\emptyset(\mathcal{P})$ jde však obecně o to vybrat náhodně na jednom patchi dvě místa p, q (dva pixely nebo čtvercové oblasti) a odečíst jejich hodnoty-intenzity $I^a(p), I^a(q)$ na náhodně vybraném kanálu a (RGB, stupně šedi, HOG deskriptor… atp.), přičemž pokud tento rozdíl nepřesáhne nějakou mez τ, je patch poslán do levé větve a naopak.

$$f_\emptyset(\mathcal{P}) = \begin{cases} 0, & if\ I^a(p) < I^a(q) + \tau \\ 1, & if\ I^a(p) \geq I^a(q) + \tau \end{cases}$$

Pro $\mathcal{H}(A_{node})$, měřící optimálnost rozdělení, záleží na tom, zda budeme $g(\emptyset, A_{node})$ potažmo $f_\emptyset(\mathcal{P})$ optimalizovat pro klasifikaci – rozdělení A_{node} na A_L a A_R podle příslušnosti k objektu/pozadí – nebo pro regresi – rozdělení A_{node} na A_L a A_R tak, aby vektory $\boldsymbol{d}_i$ pozitivních patchů $\mathcal{P}_i = (I_i, \boldsymbol{d}_i, y_i)$ měly pro levou i pravou větev co nejpodobnější velikost i směr.

Pro klasifikaci:

$$\mathcal{H}_c(A) = -\sum_c p(c|A) \log(p(c|A))$$

Což odpovídá takzvané informační nejistotě (entropii) [26], kde $c \in \{objekt(1),\ pozadí(0)\}$ a $p(c|A_{node}) = \frac{|A_c|}{|A_{node}|}$, pomocí níž měříme a snažíme se maximalizovat informační zisky $g(\emptyset, A_{node})$, kterých dosáhneme při různých rozděleních A_{node} na A_L a A_R pro různé $\emptyset = \{p, q, a, \tau\}$ funkce $f_\emptyset(\mathcal{P})_{p,q,a,\tau}$. Zde stojí za poznámku, že pomocí minimalizace $\mathcal{H}_c(A)$ maximalizujeme $g(\emptyset, A_{node})$.

Pro regresi – podobně jako pro klasifikaci, s tím rozdílem, že minimalizujeme rozptyly:

$$\mathcal{H}_r(A) = \sum_{i:D_{c_i=1}^{A_L}} \left\| \boldsymbol{d}_i - \frac{1}{|D_{c=1}^{A_L}|} \sum_{j:D_{c_j=1}^{A_L}} d_j \right\|^2 + \sum_{i:D_{c_i=1}^{A_R}} \left\| \boldsymbol{d}_i - \frac{1}{|D_{c=1}^{A_R}|} \sum_{j:D_{c_j=1}^{A_R}} d_j \right\|^2$$

V každém z listů každého stromu je pak uložena množina patchů $\mathcal{P}$ optimalizovaná přes obě tyto míry, protože míra podle, které se optimalizuje rozdělení množiny A_{node} je vybírána v každém z uzlů vedoucích k listu náhodně.

Trénování stromu končí po dosažení úplně čistých množin v listech případně po dosažení nějakého omezení na hloubku stromu – maximální počet uzlů, skrz které je možné dostat se ke každému listu – aby nedošlo k přetrénování.

Každý list tedy ukládá počet patchů (negativních i positivních), které do něj dorazily, a v případě, že jde o patch z objektu, tak i vektor směru ke středu tohoto objektu.

Odtud pak pravděpodobnostní interpretace Houghova hlasu, který daný list L generuje, při detekci:

Mějme nějakou náhodnou proměnnou $H(x)$ z intervalu $(0,1)$ určující, zda se na pozici x v obraze vyskytuje střed hledaného objektu nebo ne. Potom pro každý patch $\mathcal{P}(y) = (I(y), \boldsymbol{d}(y), c(y))$, kde y, je poloha patche, I jeho vzhled, $\boldsymbol{d}$ vektor ke středu objetu a $c \in \{0,1\}$ je příslušnost k pozadí/objektu, máme:

$$P\big(H(\boldsymbol{x})\big|I(\boldsymbol{y})\big) = P\big(H(\boldsymbol{x}), c(\boldsymbol{y}) = 1\big|I(\boldsymbol{y})\big) =$$

$$P\big(H(\boldsymbol{x})\big|c(\boldsymbol{y}) = 1, I(\boldsymbol{y})\big) \cdot P\big(c(\boldsymbol{y}) = 1\big|I(\boldsymbol{y})\big) =$$

$$P(\boldsymbol{d}(\boldsymbol{y}) = \boldsymbol{y} - \boldsymbol{x}|c(\boldsymbol{y}) = 1, I(\boldsymbol{y})) \cdot P(c(\boldsymbol{y}) = 1|I(\boldsymbol{y}))$$

Výrazy $P(\boldsymbol{d}(\boldsymbol{y}) = \boldsymbol{y} - \boldsymbol{x}|c(\boldsymbol{y}) = 1, I(\boldsymbol{y}))$ a $P(c(\boldsymbol{y}) = 1|I(\boldsymbol{y}))$, lze získat průchodem patche $\mathcal{P}(y)$ skrze uzly stromu T a výpočtem následující rovnice z hodnot uložených v listu L, do kterého patch dorazí:

$$P(H(\boldsymbol{x})|I(\boldsymbol{y}); T) = \left[\frac{1}{|D_L|} \sum_{d \in D_L} \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\|(\boldsymbol{y} - \boldsymbol{x}) - d\|^2}{2\sigma^2}\right)\right] \cdot \frac{|D_L|}{|\mathcal{P}_L|}$$

Kde D_L je množina všech směrových vektoru ke středu objektu uložených v listu L stromu T, $\boldsymbol{y}$ poloha patche $\mathcal{P}$, $\boldsymbol{x}$ poloha hypotézy o přítomnosti objektu a σ^2 je kovarianční matice $I_{2\times2}$ pro $|D_L|$ Gaussových funkcí pro odhad hustoty pravděpodobnosti $P(\boldsymbol{d}(\boldsymbol{y}) = \boldsymbol{y} - \boldsymbol{x}|c(\boldsymbol{y}) = 1, I(\boldsymbol{y}))$ pomocí

Parzen windows function [27] z vektorů $\boldsymbol{d} \in D_L$. Výraz $P(c(\boldsymbol{y}) = 1|I(\boldsymbol{y}))$ je jednoduše roven $\frac{|D_L|}{|\mathcal{P}_L|}$, kde $|\mathcal{P}_L|$ je počet pozitivních i negativních patchů, které do listu L při učení dorazily.

Pro všechny stromy T, náležející do lesa $F = \{T_t\}_{t=1}^{|F|}$, pak velikost hlasu patche $\mathcal{P}(y)$ pro objekt se středem $\boldsymbol{x}$, odpovídá průměru jejich hlasů:

$$P(H(\boldsymbol{x})\big|I(\boldsymbol{y}); \{T_t\}_{t=1}^{T}) = \frac{1}{|F|}\sum_{t=1}^{|F|} P(H(\boldsymbol{x})|I(\boldsymbol{y}); T_t)$$

Při standardním hlasování do Houhgova prostoru je pak tato hodnota pro každý patch $\mathcal{P}(y)$ celého obrazu přičtena na pozici $\boldsymbol{x}$ v Houghově prostoru $V(x)$:

$$V(x) = \sum_{y \in B(x)} P(H(\boldsymbol{x})\big|I(\boldsymbol{y}); \{T_t\}_{t=1}^{T})$$

Lokální maxima v prostoru $V(x)$ pak určují středy $\boldsymbol{x}$ objektů v obraze.

Tento přístup, tj. sčítání hlasů za jednotlivé patche a hledání lokálních maxim v $V(x)$, je sice poměrně efektivní, avšak z teoretického hlediska nesprávný vzhledem k pravděpodobnostnímu charakteru těchto hlasů – pravděpodobnosti nezávislých náhodných proměnných by se měly násobit, nikoli sčítat. V [4] proto navrhli autoři jiný přístup k sdružování hlasů od jednotlivých patchů a také jiný přístup k hledání maxim hlasů, než jsou jinak používané techniky na potlačení nemaximálních hodnot (*non-maxima suppression)* v nějakém okolí lokálního maxima prostoru $V(x)$, jež vyžadují citlivé nastavování množství dalších parametrů jako je rádius hledání maxima nebo hranice, kdy je hodnota přijata jako lokání maximum [28].

V [4] autoři zvolili namísto sčítání hlasů v Houghově prostoru $V(x)$ následující postup jejich agregace.

3.1.1 Pravděpodobnostní model pro detekci pomocí Houghova lesa

Pro všechny patche $P_i(I_i, \boldsymbol{x}_i, c_i)$ a hypotézy $H_h(\boldsymbol{x}_i + \boldsymbol{d})$, kde $H_h = 1$, pokud se na pozici $\boldsymbol{x}_i + \boldsymbol{d}$ vyskytuje střed objektu h, a $H_h = 0$, pokud je zde pozadí. Stejně tak $P_i = h$, v případě, že tento patch ze své pozice $\boldsymbol{x}_i$ generuje objekt na pozici $\boldsymbol{x}_i + \boldsymbol{d}$, a $P_i = 0$, pokud je na pozici $\boldsymbol{x}_i$ pozadí. Hledáme pak maximum jejich sdružené podmíněné pravděpodobnosti $p(\boldsymbol{H}, \boldsymbol{\mathcal{P}}|\boldsymbol{I})$. Což je po aplikaci Bayesova teorému:

$$p(\boldsymbol{H}, \boldsymbol{\mathcal{P}}|\boldsymbol{I}) \propto p(\boldsymbol{I}|\boldsymbol{H}, \boldsymbol{\mathcal{P}}) \cdot p(\boldsymbol{H}, \boldsymbol{\mathcal{P}})$$

Kde pro $p(\boldsymbol{I}|\boldsymbol{H}, \boldsymbol{\mathcal{P}})$, za předpokladu:

1. že sdružené distribuce $\boldsymbol{I} = \{I_1, \dots, I_i, \dots, I_N\}$ vzhledu patchů jsou navzájem nezávislé za podmínky sdružené pravděpodobnosti $\boldsymbol{H}, \boldsymbol{\mathcal{P}}$ existujících objektů $\boldsymbol{H} = \{H_0 = 1, H_1 \in \{0,1\}, \dots, H_h \in \{0,1\}, \dots, H_K \in \{0,1\}\}$ a jejich přiřazení patchům $\boldsymbol{\mathcal{P}} = \{P_1 \in \{0, \dots, h, \dots, K\}, \dots, P_i \in \{0, \dots, h, \dots, K\}, \dots, P_N \in \{0, \dots, h, \dots, K\}\}$ – myšleno tak, že jakmile existuje nějaké distribuci sdružených pravděpodobností $\boldsymbol{H}, \boldsymbol{\mathcal{P}}$
2. a za současného předpokladu, že vzhled I_i patche P_i je závislý pouze na jemu přiřazené hypotéze h a na všech ostatních je nezávislý, máme:

$$p(\boldsymbol{I}|\boldsymbol{H}, \boldsymbol{\mathcal{P}}) = \prod_{i=1}^{N} p(I_i|\mathcal{P}_i)$$

Z čehož po další aplikaci Bayesova teorému:

$$p(\boldsymbol{I}|\boldsymbol{H}, \boldsymbol{\mathcal{P}}) = \prod_{i=1}^{N} p(I_i|\mathcal{P}_i) \propto \prod_{i=1}^{N} \frac{p(\mathcal{P}_i|I_i)}{p(\mathcal{P}_i)}$$

Pro $p(\boldsymbol{H}, \boldsymbol{\mathcal{P}})$ pak v [4] zavádí následující omezení na povolené konfigurace $p\big(\boldsymbol{H} = \{H_0 = 1, H_1 \in \{0,1\}, \dots, H_h \in \{0,1\}, \dots, H_K \in \{0,1\}\}, \boldsymbol{\mathcal{P}} = \{P_1 \in \{0, \dots, h, \dots, K\}, \dots, P_i \in \{0, \dots, h, \dots, K\}, \dots, P_N \in \{0, \dots, h, \dots, K\}\}\big)$ tak, aby se nestalo, že existuje patch P_i přiřazený nějakému objektu h, který je však na své pozici $\boldsymbol{y}_h = \boldsymbol{x}_i + \boldsymbol{d}$ roven nule, tj. $H_h(\boldsymbol{y}_h) = 0$:

$$\prod_{i=1}^{N} H_{P_i=h} = 1$$

A penalizace za každé $H_h = 1$, tedy za každou aktivovanou hypotézu o přítomnosti objektu tak, aby obraz byl vysvětlen co nejmenším možným počtem objektů:

$$p(\boldsymbol{H}) = \prod_{h=0}^{K} e^{-\lambda H_h}$$

Z této penalizace a omezení konfigurace pak pro $p(\boldsymbol{H}, \boldsymbol{\mathcal{P}})$ dostaneme:

$$p(\boldsymbol{H}, \boldsymbol{\mathcal{P}}) = \prod_{i=1}^{N} H_{P_i=h} \cdot \prod_{h=0}^{K} e^{-\lambda H_h} \cdot \prod_{i=1}^{N} p(\mathcal{P}_i)$$

Nyní dosazením odvozených $p(\boldsymbol{H}, \boldsymbol{\mathcal{P}})$ a $p(\boldsymbol{I}|\boldsymbol{H}, \boldsymbol{\mathcal{P}})$ do $p(\boldsymbol{H}, \boldsymbol{\mathcal{P}}|\boldsymbol{I}) \propto p(\boldsymbol{I}|\boldsymbol{H}, \boldsymbol{\mathcal{P}}) \cdot p(\boldsymbol{H}, \boldsymbol{\mathcal{P}})$ a zkrácením, získáme:

$$p(\boldsymbol{H}, \boldsymbol{\mathcal{P}}|\boldsymbol{I}) \propto \prod_{i=1}^{N} p(\mathcal{P}_i|I_i) \cdot \prod_{i=1}^{N} H_{P_i=h} \cdot \prod_{h=0}^{K} e^{-\lambda H_h}$$

Tento pravděpodobnostní model použijeme k nalezení konfigurací $\boldsymbol{H} = \{H_0 = 1, H_1 \in \{0,1\}, \dots, H_h \in \{0,1\}, \dots, H_K \in \{0,1\}\}$, $\boldsymbol{\mathcal{P}} = \{P_1 \in \{0, \dots, h, \dots, K\}, \dots, P_i \in \{0, \dots, h, \dots, K\}, \dots, P_N \in \{0, \dots, h, \dots, K\}\}$ s maximální aposteriorní pravděpodobností $p(\boldsymbol{H}, \boldsymbol{\mathcal{P}}|\boldsymbol{I})$ – MAP inference.

Kapitola 4

Aplikace, jež je součástí této práce, byla navržena a implementována v jazyce C++ s využitím volně dostupné knihovny OpenCV a pro potřeby této práce upravených částí kódu z [4]. Pravděpodobností model (dvě vrstvy Houghova lesu), jenž je základem této aplikace, byl natrénován na rozpoznávání objektů typu „chodec“, čímž je míněn člověk v alespoň přibližně vzpřímené poloze bez ohledu na stranu, kterou je chodec natočen ke kameře. Aplikace není limitována jednou třídou objektů, lze ji s příslušnými úpravami (například typ extrahovaných příznaků z obrazu) naučit na libovolný typ objektů

Aplikace je rozdělena do dvou hlavních částí. První je určena k naučení obou použitých vrstev Houghova lesa a byla pro účel této práce implementována od úplného začátku na základě poznatků popsaných v kapitole 3. V druhé části je pak přepracovaný kód z [4], určený k výpočtu pravděpodobnosti hypotézy o výskytu objektu na určité pozici v obraze – detekci objektů – pomocí MAP inference.

4.1 Učení

Učení jednotlivých stromů Houghova lesa začíná výpočtem příznaků v pozitivních a negativních obrazech, pro potřeby detekce chodců v této práci bylo vybráno 16 + 16 typů příznaků. Ve všech trénovacích obrazech (pozitivních i negativních) je pak po celé jejich ploše spočítáno všech 32 vybraných příznaků (kanálů obrazu) – 3 pro kanály barevného prostoru **L*a*b**, 4 kanály pro **první a druhé derivace intenzity obrazu** ve směrech× a **y**, 9 kanálů pro **9 binů HoG příznaků**, 16 kanálů pro **max/min filtrované obrazy** předchozích 16 kanálů kvůli odfiltrování šumu.

Po načtení trénovacího obrazu a spočítání příznaků, se započne s extrakcí čtvercových patchů o velikost 16x16 pixelů. Pozitivní obraz obsahuje na pozici určené 6 souřadnicemi (levý horní roh, pravý dolní roh a střed) takzvaný obalový box, uvnitř kterého se nachází objekt, na který se mají stromy naučit. Uvnitř toho obalového boxu se na náhodně zvolených pozicích extrahuje 50 patchů označených jako pozitivní – tedy jako náležející k hledanému objektu – s připojenou informací o relativní poloze středu objektu vzhledem k poloze, ze které byl patch odebrán. Stejně tak z každého negativního obrázku je náhodně vybráno 50 pozic, ze kterých se na všech kanálech extrahuje tomu odpovídající počet patchů. U negativních patchů je zaznamenán pouze jejich vzhled.

Ze všech pozitivních a negativních trénovacích patchů je pro učení každého jednotlivého stromu vybrána jejich náhodná podmnožina. Tato podmnožina je na každém uzlu trénovaného stromu, jenž není listem stromu, podrobena třem tisícům náhodných binárních testů. Z těchto testů se vybere ten, který dělí vstupní množinu podle nějakého klíče na dvě podmnožiny, s co nejmenší vnitřní variabilitou sledované klíčové statistiky. Testy tak mají v každém uzlu za úkol rozdělit vstupní množinu patchů do dvou navazujících větví, kde na ně na základě ohodnocení z vybraného nejlepšího testu čeká buď další uzel a tedy i další výběr dělícího binárního testu nebo list stromu, ve kterém se dané větvení končí a rekurzivně pokračuje o úroveň výš, dokud všechna větvení nekončí listem na základě nemožnosti nalezení lepšího dělení nebo kvůli tomu, že se narazilo na předem danou maximální hloubku stromu.

4.1.1 Nalezení testu

Hledání nejlepšího binárního testu pro dělení vstupní množiny patchů na dvě podmnožiny s minimálními rozptyly ve vzhledu patchů (klasifikace) nebo jejich relativních pozicích vzhledem ke středu objektu (regrese) probíhá pomocí takzvaného 2-pixelového testu.

Pro každý ze třech tisíc binárních testů se náhodně vybere jeden kanál obrazu, v něm jeden pozitivní a jeden negativní patch a v každém z nich po dvou pixelech na stejných náhodných pozicích v rámci tohoto negativního a pozitivního patche. Hodnoty obou pixelů z jednoho patche ve vybraném kanálu se odečtou a tyto rozdíly intenzit (pixelů uvnitř pozitivního a negativního patche) se zprůměrují. Tento průměr pak slouží jako hranice tolerance, po kterou muže zajít rozdílnost ve vzhledu dvou patchů, aby mohly podle daného testu skončit ve stejné podmnožině.

Kvalitativní hodnocení schopnosti testu rozdělit vstupní množinu na dvě navzájem co možná nejvíce rozdílné (a tedy naopak vnitřně homogenní) podmnožiny se provádí pomocí jedné ze dvou měr. Tato míra je pro každý uzel stromu, kde se optimalizuje test, vybrána náhodně. Program volí mezi minimalizací rozptylu relativních pozic patchů ke středu objektu a minimalizací rozdílů ve vzhledu jednotlivých patchů skrze maximalizaci informačního zisku – minimalizaci informačního entropie v daném testovacím uzlu. Test s nejlepším skóre je zvolen za test nejlépe rozdělující vstupní množinu pozitivních a negativních patchů na dvě podmnožiny.

4.1.2 Konstrukce listů stromu

V případě, že skrze optimalizaci testů již nelze nalézt dělení množiny patchů na podmnožiny tak, aby jedna z podmnožin nezůstala prázdná, případně v situaci, kdy jedna z výsledných podmnožin neobsahuje žádné pozitivní patche, přichází čas na tvorbu listu stromu.

V každém listu binárního rozhodovacího stromu Houghova lesu, kterým větvení stromu končí, je uložen celkový počet patchů, které do uzlu (teď již listu) dorazily a počet + hodnoty souřadnic relativních směrů dorazivších pozitivních patchů vzhledem k středům objektů, z nichž byly patche extrahovány.

Tyto hodnoty pak slouží k výpočtu pravděpodobnosti příslušnosti patche k objektu a k určení středu objektu při detekci.

Konstrukcí všech stromů v lese učení prediktivního modelu Houghova lesu končí. V této práci byla po provedení experimentů s různými počty do kaskády řazených stromů zvolena varianta lesa s 15 paralelně natrénovanými stromy o maximální hloubce větvení uzlů 15 – tj. od kořenového uzlu stromu je to k jakémukoli listu stromu maximálně skrze 13 dalších větvení. A jedním tomuto lesu předřazeným stromem, jenž má za úkol rychle odfiltrovat oblasti obrazu, které s největší pravděpodobností žádný hledaný objekt neobsahují, případně odfiltrovat ty části objektu, které pro svou podobnost s pozadím nejsou o výskytu objektu dostatečně vypovídající a snížit tak výpočetní náročnost detekce samotných objektů a jejich pozic, kterou zajišťuje les 15 stromů pod tímto předřazeným slabým klasifikátorem.

K naučení první vrstvy kaskády byla v této prácí zvolena následující metoda. Ze 400 pozitivních a 400 negativních obrazů pro tyto účely standardně používaného datasetu *TUD campus pedestrians* bylo v každém obrázku extrahováno padesát patchů. Strom v první kaskádě tedy na prvním uzlu pracoval se 40000 patchů, které se v průběhu učení výše popsaným způsobem snažil na základě podobnosti relativních poloh patchů vzhledem ke středům objektů, ze kterých pocházejí , rozdělit do co možná nejhomogennějších listů. Tento strom pak byl použit pro otestování všech 400 pozivních obrazů, původně určených jeho natrénovaní. Pozitivní obrazy, u kterých bylo pro tento strom obtížné v místě výskytu objektu tento objekt rozpoznat – v tom smyslu, že strom označil více než 70% pixelů tomuto objektu náležejících za pixely pozadí – byly zrcadlově otočeny a vloženy zpět mezi původní pozitivní obrazy. Jednalo se o 168 původních pozitivních obrazů, čímž se množina pozitivních trénovacích obrázků rozrostla na 568 obrazů. Tato vrstva kaskády byla pro každý obraz schopna správně odfiltrovat 30-40% pixelů náležejících pozadí.

Druhá vrstva kaskády byla natrénována s použitím nové množiny 600 negativních obrazů, kde přibližně třetinu tvoří náhodně vybrané obrazy z původních negativních obrazů a zbytek nové obraz „pozadí“ – obrazy na nichž se objekt nevyskytuje. Množina pozitivních obrazů se skládá z obrazů původních a těch zrcadlově otočených, které první stupeň kaskády nebyl schopen rozpoznat.

Třetí vrstva, paralelní les 15 Houghových stromů, pak byla při trénování rozdělena do třech skupin po pěti stromech. První skupina byla natrénována na všech 568 pozitivních obrazech a stejném počtu obrazů negativních. S touto skupinou pěti stromů byla provedena detekce na pozitivních trénovacích obrázcích, z nichž obrázky s objekty, které těchto pět stromů nedokázalo správně detekovat, byly použity pro natrénování další skupiny. Analogicky se postupovalo také při učení poslední třetiny Houghova lesa – objekty, které prvních deset stromů neurčilo správně, byly použity pro učení posledních pěti.

4.2 Detekce

Nejčastěší přístup k detekci při použití detektorů založených na hlasování pomocí Houghova lesa spočívá ve sčítání pravděpodobností, které jednotlivé patche z okolí hypotetického středu hledaného objektu přisuzují skutečnosti, že by se zde objekt s tímto středem skutečně mohl vyskytovat. Tyto hodnoty pravděpodobností jsou výstupem listů jednotlivých stromů, do kterých patche dorazí poté, co projdou skrze rozhodovací uzly těchto předem natrénovaných stromů. Po sečtení všech hlasů (velikostí pravděpodobností) v Houghově hlasovacím prostoru se za pomoci různých technik (např. pomocí Gaussova filtru, nebo mean-shiftu) hledají pozice s maximem hlasů, které pak určují středy hledaných objektů.

V této práci byl naproti tomu použit pravděpodobnostní přístup k slučování hlasů odvozený v [4], viz též předchozí kapitola. Tato metoda slučování hlasů více odpovídá pravděpodobnostnímu charakteru těchto hlasů, jež na výstupu jednotlivých listů L stromů T Houghova lesa mají následující formu:

$$P(\mathcal{P}_i = h|I_i;T) = \left[\frac{1}{|D_L|}\sum_{d_l \in D_L, l=1}^{|D_L|}\frac{1}{2\pi\sigma^2}\exp\left(-\frac{\|(x_i - y_h) - d_l\|^2}{2\sigma^2}\right)\right].\frac{|D_L|}{|\mathcal{P}_L|}$$

Kde

$P(\mathcal{P}_i = h|I_i;T)$… je pravděpodobnost, že patch i se vzhledem I_i generuje objekt h.

Hledání maximální sdružené a posteriorní pravděpodobnosti (MAP inference) $p(\boldsymbol{H}, \boldsymbol{\mathcal{P}}|\boldsymbol{I})$ z předchozí kapitoly je zde nahlíženo z pohledu jednoduchého dopravního problému, jak rozložit *m* cílových míst tak, aby bylo při minimalizaci nákladů s tím spojených obslouženo všech *n* zákazníků. Převedeno do řeči hlasů generovaných patchi $\mathcal{P}_i(I, x)$ v listech L s $|D_L| > 0$ a hypotéz $H_h(\boldsymbol{y}_h) = 1$ o nalezených objektech, používané v problematice detekce objektů využívající obecné Houghovy transformace: jde o to, jak všechny patche $\boldsymbol{\mathcal{P}}$ přiřadit aktivovaným hypotézám $\boldsymbol{H}$ (včetně řiřazení patche k pozadí, kde $h = 0$) a minimalizovat přitom náklady, tedy maximalizovat pravděpodobnost, této příslušnosti. Náklad, tedy hlas, je zde vyjádřen jako absolutní hodnota logaritmu pravděpodobnosti, že daný patch $\mathcal{P}_i$ volí pro určitou hypotézu $|\log P(\mathcal{P}_i = h|I_i)|$.

Po zlogaritmování výrazu $p(\boldsymbol{H}, \boldsymbol{\mathcal{P}}|\boldsymbol{I})$ a načtení všech hlasů $|\log P(\mathcal{P}_i = h|I_i)|\ \forall i, h$ do Houghova prostoru hlasů $V(\boldsymbol{H}(\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{d}) \in \{0,1\}^K)$ se tedy hledá takové přiřazení patchů $\boldsymbol{\mathcal{P}}$ k aktivovaným hypotézám $\boldsymbol{H}$, tj. hypotézám pro které platí, že hodnota hlasů v Houghově prostoru $V(\boldsymbol{H})$ je pro ně v absolutní hodnotě větší než penalizace λ za jejich aktivaci, tj. $V(\boldsymbol{H}) > \lambda$, které minimalizuje pro tyto hypotézy následující nákladovou funkci:

$$Cost(\boldsymbol{H}) = \min_{\boldsymbol{\mathcal{P}}} Cost(\boldsymbol{H}, \boldsymbol{P})$$

$$= \sum_{h=1}^{K} |-\lambda H_h| + \sum_{i=1}^{N} \min\left(\min_{h:H_h=1, h\neq 0} (|\log P(\mathcal{P}_i = h|I_i)|), |\log P(\mathcal{P}_i = 0|I_i)| \right)$$

Kde $Cost(\boldsymbol{H}, \boldsymbol{P})$ je jednoduše absolutní hodnota logaritmu $p(\boldsymbol{H}, \boldsymbol{\mathcal{P}}|\boldsymbol{I})$:

$$Cost(\boldsymbol{H}, \boldsymbol{P}) = |\log p(\boldsymbol{H}, \boldsymbol{\mathcal{P}}|\boldsymbol{I})| = \sum_{i=1}^{N} |\log P(\mathcal{P}_i = h|I_i)| + \sum_{h=0}^{K} |-\lambda H_h| + \sum_{i=1}^{N} \left| \lim_{\mathrm{H_{P_i=h}} \to 0} \log(\mathrm{H_{P_i=h}}) \right|$$

Což je přesně forma jednoduchého dopravního problému (*Uncapacitated Facility Location Problem*)[29], jehož přibližné řešení lze nalézt pomocí hladového algoritmu:

V této práci byl použit následující algoritmus minimalizace výše zmíněného výrazu:

Nejprve je využito prvních dvou předřazených vrstev kaskády, každá o jednom stromu, pro „odfiltrování“ patchů $\mathcal{P}_i$, které s pravděpodobností $P(\mathcal{P}_i = 0|I_i; s) = 1 - \frac{|D_L|}{|\mathcal{P}_L|} \geq 0{,}7$ patří pozadí, kde $s \in S = \{1,2,3\}$ je úroveň kaskády. V těchto úrovních kaskády tedy nejde o stromy Houghova lesa hlasující pro hypotetické středy objektů, ale o náhodné rozhodovací stromy použité pro klasifikaci patchů. Pozice rovnající se pozicím pixelů ve středu takových patchů jsou v pomocné masce, svými rozměry odpovídající testovanému obrazu, nastaveny na 0 – nulová pravděpodobnost náležení patche k objektu z hledané třídy. Přičemž každá úroveň už počítá pravděpodobnosti pouze

pro patche, jež nebyly nastaveny předchozí úrovní na nulu. Patche, u kterých se vrstvy shodnou na pravděpodobnosti příslušnosti k pozadí $P(\mathcal{P}_i = 0|I_i; s) < 0{,}7$, mají zprůměrovanou pravděpodobnost $1 - P\left(\mathcal{P}_i = 0\middle|I_i; \{s\}_1^{|S|-1}\right) = \frac{\sum_{s=1}^{|S|-1} 1-P(\mathcal{P}_i=0|I_i;s)}{|S|-1}$ za $|S| - 1$ úrovní kaskády zanesenu na příslušnou pozici pomocné masky a pokud je tato větší nebo rovna 70%, je takový patch postoupen do poslední úrovně kaskády detektoru pro minimalizaci výše odvozené nákladové funkce:

1. Na počátku běhu samotného hladového algoritmu je pomocí poslední úrovně kaskády spočten náklad příslušnosti všech patchů $\boldsymbol{P}$, které nebyly v předchozích kaskádách detektoru označeny jako pozadí, k pozadí tj. $|\log P(\mathcal{P}_i = 0|I_i)|$. Tyto hodnoty lze v kontextu dopravního problému brát jako náklad na obsloužení zákazníka i jediným zatím otevřeným dodavatelem y_0, který v naší interpretaci odpovídá hypotéze, že patch je vysvětlen pozadím a nepatří k žádnému objektu (k uspokojení zákazníka není třeba otevírat novou lokaci dodavatele). Tato hypotéza (dodavatel) je vždy aktivní, tj. vždy platí $H_0 = 1$.
2. Každý patch $\mathcal{P}_i$ z $\boldsymbol{P}$ hlasuje pro všechny jemu dostupné hypotézy h z $\boldsymbol{H}$ na pozicích $y_h = x_i + \boldsymbol{d}$ do akumulátoru hlasů $M(\boldsymbol{y})$. Velikost takového hlasu V patche $\mathcal{P}_i$ pro h na y_h je rovna: $V_i(h) = \min(|\log P(\mathcal{P}_i = h|I_i)| - |\log P(\mathcal{P}_i = 0|I_i)|, 0)$ – tedy hodnota, o kterou by se snížily náklady na vysvětlení (obsloužení) patche i, jeho přeřazením z $h = 0$ do lokace h mezi detekované objekty oproti nákladům na jeho vysvětlení tvrzením o příslušnosti k pozadí. Dostupnost hypotéz je pro patch i dána jeho souřadnicí, maximálním obalovým boxem kolem této souřadnice na daném měřítku a relativními pozicemi středů možných objektů (vůči x_i), jež jsou určeny hlasy generovanými patchem i v listech jednotlivých stromů.
3. V Houghově prostoru hlasů se poté nalezne hodnota globálního minima, tedy taková lokace, která nejvíce snižuje celkové náklady (největší záporné číslo) a pozice, na které byla nalezena, se označí za střed nově detekovaného objektu v případě, že tato hodnota je větší než náklad na novou detekci λ. Tj. $|\min M(\boldsymbol{y})| > \lambda$. V opačném případě algoritmus končí.
4. Z akumulátoru hlasů $M(\boldsymbol{y})$ vymaž hlasy patchů $\mathcal{P}_i$, které hlasovaly pro poslední detekci, a vrať se do bodu 3.

Tento algoritmus se od algoritmu použitého v [4] liší v tom, že je „hladový“ jak ve výběru, kterou z hypotézy označit za objekt, tak i v případě hlasujících patchů, které jsou poté co je „jejich“ hypotéza označena za detekci iterativně mazány z množiny všech volících hlasů a považovány za vysvětlené. Tento přístup byl v této práci zvolen na základě předpokladu, že jeden patch nemůže náležet ke dvěma objektům, zároveň je zde předpoklad, že tento způsob hlasování bude jednodušší a zároveň tedy i rychlejší.

Detekce objektů na různých měřítcích je stejně jako v [4] zajištěna vytvořením obrazové pyramidy a pyramidy akumulátorů simulujících Houghův prostor na těchto měřítcích.

Kapitola 5

5.1 Testy a evaluace

Pro ověření, ohodnocení a porovnání schopností navrženého detektoru chodců byly v této prácí použity dvě, standardně za tímto účelem používané, obrazové sady: *TUD Crossing* a *TUD Campus*.

První sada pochází z kamery zabírající přechod pro chodce, video je rozloženo do sekvence 200 snímků zachycujících přibližně 1000 pozitivních instancí objektu „chodec“. Chodci se na snímcích vyskytují v přibližně stejném měřítku, ze dvou bočních stran a často ve velkém vzájemném zákrytu bez možnosti určení přesného počtu chodců v některých extrémních případech. Standardně se jako přítomnost chodce bere situace, kdy je takový chodec viditelný alespoň z 50%. V práci [4], ze které myšlenkově vychází i tato práce, se však autoři zaměřili na zpřesnění detekce ve smyslu nalezení i takových chodců, jimž je vidět jen hlava a alespoň jedna noha.

Druhá sada obsahuje přes 70 snímků z kamery zabírající kampus university s chodci v minimálně třech různých měřítcích s celkovým počtem pozitivních výskytů rovnajícím se více než třem stovkám chodců. Chodci jsou opět viditelní z obou bočních stran a často z více než poloviny v zákrytu za sebou.

Stejně jako v ostatních pracích byl i v této práci použit standardní protokol pro určení pozitivních detekcí. Každá taková detekce je detektorem vyznačena obalovým boxem okolo detekovaného objektu. Aby mohla být detekce označena za pozitivní je třeba aby se detektorem určený obalový box minimálně z 50% shodoval s obalovým boxem určeným anotací, jež je součástí každé testovací sady, obsahující 4 souřadnice určující polohu a velikost obdelníku okolo každého anotovaného objektu.

Pro porovnání detektoru navrženého a implementovaného v této práci s detektorem z [4] bylo, co se přesnosti detekcí týče, použito takzvaných *Precision – Recall* křivek. Po jedné pro každý detektor a testovací sadu. Hodnoty pro křivky jsou získány následujícím způsobem:

Na ose x vynášíme takzvaný $Recall$, což je poměr pozitivních detekcí detektoru ku skutečnému počtu objektů danému anotací sady, tedy:

$$Recall = \frac{TP}{TP + FN}$$

Kde TP je počet pozitivních detekcí (*True Positive*) a FN (*False Negative*)je počet objektů, které detektor „přehlédl“ a označil jako pozadí.

Různé hodnoty pro $Recall$ jsou obecně u detektorů objektů získávány nastavením parametrů detektoru určujících jeho citlivost – míru jistoty potřebnou pro označení hypotézy h o výskytu objektu za pravdu. V případě práce z [4] a této práce se jedná o nastavování různých hodnot λ (viz předchozí kapitola). Čím nižší tato hodnota λ určující náklad na vysvětlení scény přidáním další detekce, tím je detektor citlivější a pro zakreslení další detekce mu stačí nižší míra jistoty, zde podmíněné pravděpodobnosti $P(\mathcal{P}_i = h|I_i; T)$, přes všechny stromy T.

Pro takto dané hodnoty osy x pak pro osu y spočteme přesnost ($Precision$) detektoru jako:

$$Precision = \frac{TP}{TP + FP}$$

Kde FP je počet falešných detekcí, tedy detekcí na místech, kde se žádné skutečné objekty nevyskytují.

Z těchto dvou hodnot pak lze v místě rovnosti $Recall = Precision$ získat takzvaný *Equal Error Rate* (*EER*) charakterizující detektor, resp. jeho přesnost, jako procento detekovaných objektů při rovnajících se pravděpodobnosti, že dojde k falešné detekci a pravděpodobnosti nezaznamenání objektu. Z těchto hodnot lze odvodit „optimální“ míru citlivosti detektoru pro jeho nastavení v reálném provozu, ačkoliv takové optimum vždy závisí také na konkrétní aplikaci detektoru, kdy v některých případech je žádoucí minimalizovat falešné detekce a tím přijít i o některé detekce skutečných objektů a v jiných naopak mít co největší skóre detekovaných i se zvýšeným rizikem, že dojde k falešným detekcím.

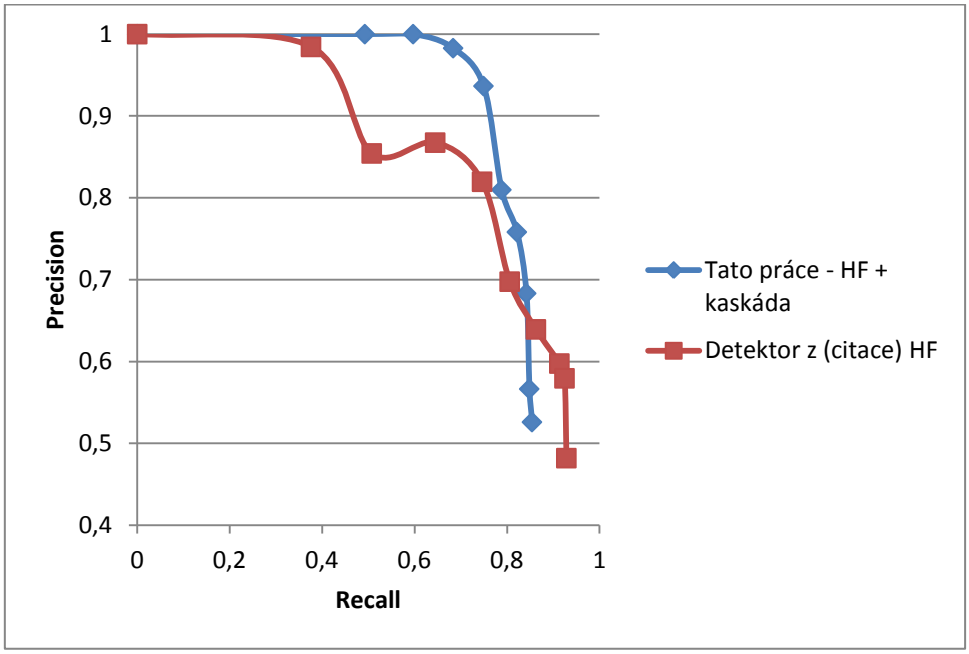
Rychlost detekcí byla měřena pomocí funkcí *getTickCount* a *getTickFrequency* z knihovny OpenCV, jejichž výstupní hodnoty poté byly převedeny na vteřiny. Konfigurace testovacího PC:

- CPU Intel Core i5-3320M 2.60 GHz
- RAM 4.0 GB
- Integrovaný grafický adaptér Intel Graphics HD 4000
- Windows 7 Enterprise SP1 64 bit

Během testů vždy běžel pouze jeden z detektorů a nejnutnější systémové úlohy. Detektory byly testovány samostatně. Houghův les byl pro oba detektory použit stejný – ten natrénovaný v této práci.

5.1.1 Test nad datovou sadou TUD crossing

Výsledky testů nad touto datovou sadou byly poměrně překvapivé. Z *Precision* – *Recall* křivek v grafu 1 lze vyčíst, že detektor navržený v této práci má pod křivkou o něco větší plochu než detektor z [4]. To ukazuje na výrazné snížení výskytu falešných detekcí při stejném počtu detekcí pozitivních.



Graf 1

Precision – Recall křivky pro detektor z [4] (červená křivka) a detektor navržený a implementovaný v této práci (modrá křivka). Obrázková sada TUD crossing.

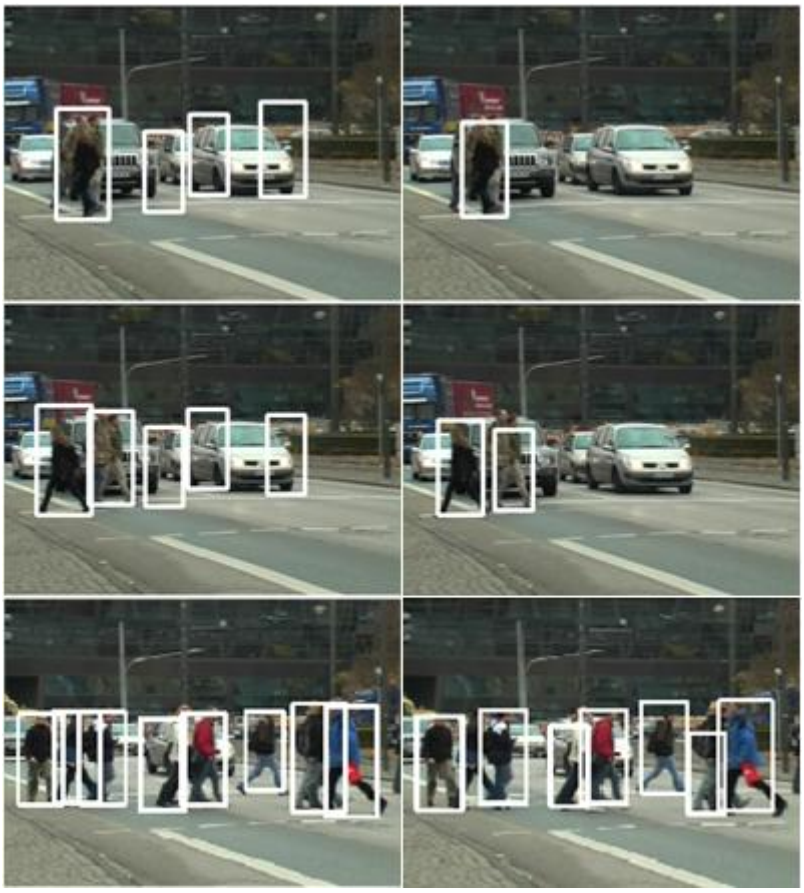
Na druhou stranu se však nepodařilo dosáhnout vyšších hodnot *Recall* než nějakých 0,84, zatímco hodnoty pro *Recall* se v případě detektoru z [4] významně blíží k jedné a křivka pro detektor z této práce tak padá mnohem strměji. Tato situace je dána tím, že detektor z [4] je schopen detekovat chodce i s již velmi výrazným zakrytím nebo velmi blízko sebe. Detektor z této práce toho bohužel kvůli nulování hlasů není schopen – často vyhodnotí chodce, kteří jsou těsně u sebe jako jeden objekt. Pro ilustraci viz příklady detekcí na této datové sadě na obrázku 19.

V tabulce 2 jsou vyznačeny *EER* a průměrné rychlosti detekce na snímek při rozlišení 414 x 310 pixelů.

Detektor	EER	Ø Rychlost (s/snímek)
Detektor z [4] HF	~76%	268
Detektor HF+kaskáda(tato práce)	**~80%**	**79**

Tabulka 2

Tabulka Equal Error Rate a rychlostí zpracování snímků z testovací sady TUD crossing.

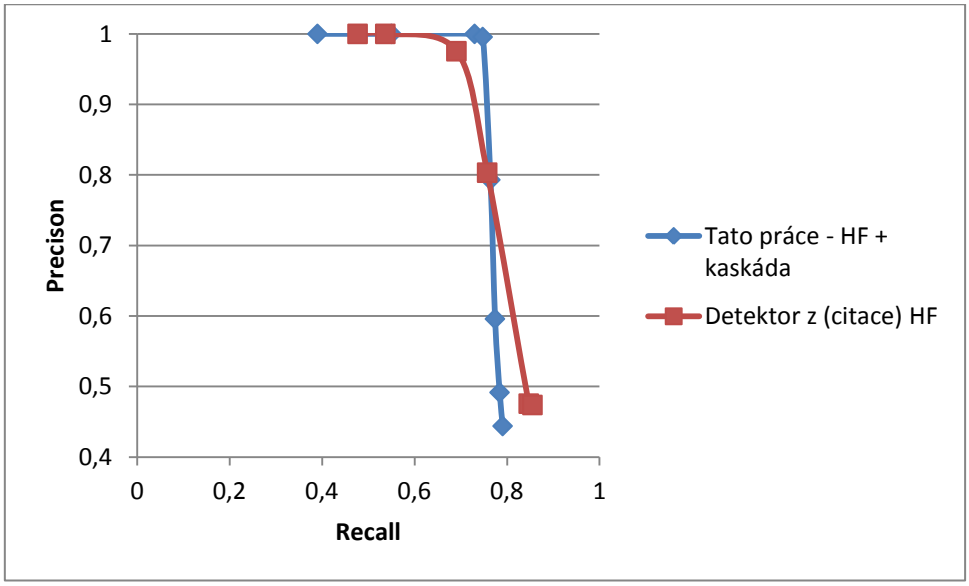


Obrázek 19

Na obrázku je na prvních dvou párech obrázků pro ilustraci znázorněn pokles falešných detekcí oproti detektoru z [4] (obrázky vlevo) při zachování citlivosti na pozitivní detekce. Poslední dvojice obrázků naopak ilustruje lepší schopnost detektoru z [4] detekovat i objekty, které se nachází velmi blízko sebe, případně ve velkém zákrytu.

5.1.2 Test nad datovou sadou TUD campus

Podobně jako v případě obrázkové sady *TUD crossing* i zde lze z *Precision – Recall* křivek vyčíst, že v rozmezí hodnot *Recall* přibližně od 0,6 do 0,8 je v této práci navržený detektor schopen při stejné citlivosti na pozitivní detekce částečně snížit počet detekcí negativních. Opět se však negativně projevil vliv nulování hlasů, v tomto případě dokonce v ještě větší míře, což je způsobeno především velkým zákrytem objektů na velmi rozdílných měřítcích. Pro objekt na velkém měřítku je totiž třeba pro detekci více hlasů a vznikne tedy i větší plocha, která se po odhlasování nuluje, čímž však snadno zaniknou i hlasy pro objekty menších měřítek, které se nachází v těsné blízkosti, respektive ve velkém zákrytu za tímto větším objektem (Obrázek 20). To, že je tento vliv na testovací obrázkové sadě *TUD crossing* mnohem větší než v předchozím případě, je zřejmé i z mnohem prudšího poklesu křivky pro detektor z této práce kolem hodnoty *Recall* 0,78. Detektor navržený v této práci nalezne pro velmi rozdílná nastavení citlivosti velmi podobné množství pozitivních detekcí, jediné co se při testech na TUD campus sadě významně mění při změně citlivosti je tedy počet falešných detekcí – změny v těch pozitivních jsou však téměř nulové.



Graf 2

Precision – Recall křivky pro detektor z [4] (červená křivka) a detektor navržený a implementovaný v této práci (modrá křivka). Obrázková sada TUD campus.

Detektor	EER	Ø Rychlost (s/snímek)
Detektor z [4] HF	~77%	186
Detektor HF+kaskáda(tato práce)	**~77%**	**67**

Tabulka 3

Tabulka Equal Error Rate a rychlostí zpracování snímků z testovací sady TUD campus.

Obrázek 20

Horní pár obrázků ilustruje, jak špatně si detektor navržený v této práci poradí s objekty, které jsou těsně u sebe (obr. vlevo nahoře) – detektor z [4] dokázal detekovat tři k sobě těsně přiléhající objekty (obr. vpravo nahoře), zatímco detektor z této práce kvůli nulování hlasů objekt mezi první a druhou detekcí zleva přehlédl. Spodní dvojice obrázků na druhou stranu ukazuje, že nulováním hlasů lze předejít některým zdvojeným detekcím v okolí již jednou detekovaného objektu.

Kapitola 6

6.1 Závěr, diskuse výsledků testů, návrhy

Výsledky testů ukazují, že předpoklady vyřčené v úvodu a cílech práce se bohužel potvrdily jen z části. Hodnoty průměrné rychlosti detekce v sekundách na jeden snímek ukazují, že nebylo od věci předpokládat zrychlení pomocí nalezení míst, na kterých se objekt „určitě“ nevyskytuje. Z testů vyplývá, že nedošlo ke snížení počtu detekcí, pokud nešlo o objekty, který si byly vzájemně velmi blízko - z pohledu pozorovatele se dotýkaly – nebo ve velkém zákrytu. To však lze přisuzovat spíše použitému algoritmu pro samotnou detekci, tedy mazání hlasů tak, aby žádný hlas, který už hlasoval pro jeden detekovaný objekt, nemohl hlasovat pro dva a více objektů. Předpoklad, že jeden hlas, potažmo patch, nemůže patřit dvě objektům zároveň se tedy v testech ukázal jako nepříliš domyšlený. Ostatně i lidský mozek v případě, že vidí člověka se čtyřma nohama a dvěma hlavami, předpokládá spíše existenci dvou lidí v zákrytu než nějakou hříčku přírody. Pokud by hlasy nebyly nulovány, ale bylo jim naopak dovoleno změnit příslušnost k hypotéze, mohl by jeden patch hlasovat pro středy vícero objektů, které se nacházejí těsně u sebe nebo i v zákrytu.

Použití kaskády přineslo, v závislosti na množství objektů v obraze, přibližně 3 – 4 násobné zrychlení. První dva stupně kaskády totiž dokážou správně odfiltrovat mezi 60 až 75% patchů náležejícím k pozadí, přičemž průchod skrze náhodné binární rozhodovací stromy těchto dvou kaskád je mnohem rychlejší než hlasování do Houghova prostoru pomocí lesa v poslední fázi detekce, kde hledáme už konkrétní pozice objektů. Ostatně hlasování do Houghova prostoru obecně bývá u detektorů využívajících tuto techniku výpočetně a časově nejnáročnější částí detekce. V této práci by se dala náročnost hlasování dále snížit paralelizací výpočtů a především zmenšením hlasovacího prostoru ze třech dimenzí (x, y, s - měřítko) na dvě (x, y) pomocí natrénovaní lesa, který by zároveň se souřadnicemi x a y explicitně hlasoval i pro měřítko s. Případně použít techniku z [30], využívající namísto příznaků vypočtených v celé obrazové pyramidě, složené ze stejných obrázků různých měřítek, pouze aproximace příznaků vypočtené v jednom základním měřítku na další požadovaná měřítka.

Další možností pro zrychlení, ale i zpřesnění, detektoru by mohlo být použití nějakého více „diskriminativního“ přístupu při odfiltrování pozadí. Například použít namísto binárního rozhodovacího stromu klasifikátor SVM a princip detekce pomocí posuvného okna, čímž by bylo

možné rychle vyloučit celé oblasti obrazu uvnitř posouvaného detekčního okna a ne jen jednotlivé pixely.

Literatura

1. Felleman, D. J.; van Essen, D. C.: Distributed Hierarchical Processing in the Primate Cerebral Cortex, *Cerebral Cortex* 1, 1991, s. 1-47.
2. Viola, P.; Jones, M.: Robust real-time face detection. International journal of computer vision, ročník 57, č. 2, 2004, s. 137-154.
3. Kisačanin, B.; Pavlović, V.; Huang, T. S.: Real-Time Vision for Human-Computer Interaction. Springer US, 2005.
4. Barinova, O.; Lempitsky, V.; Kohli, P.: On detection of multiple object instances using hough transforms. In IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2010, s. 2233-2240.
5. Gall, J.; Yao, A.; Razavi, N.; van Gool, L.; Lempitsky V.: Hough Forests for Object Detection, Tracking, and Action Recognition. In IEEE Transactions on Pattern Analysis and Machine Intelligence, ročník 33, č. 11, 2011, s. 2188-2202.
6. Gall, J.; Lempitsky, V.: Class-specific hough forests for object detection. In IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, s. 1022-1029.
7. Dalal, N.; Triggs, B.: Histograms of Oriented Gradients for Human Detection. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, San Diego, CA, USA.
8. Lowe, D. G. : Distinctive image features from scale-invariant keypoints. In International Journal of Computer Vision, ročník 60, č. 2, 2004, s. 91–110.
9. Bay, H.; Ess, A.; Tuytelaars, T.; van Gool, L.: SURF: Speeded Up Robust Features, Computer Vision and Image Understanding (CVIU), 2008, č. 3, s. 346–359.
10. Li, S.; Wang, D.; Zheng, Z.; Wang, H.: Multi-view vehicle detection in traffic surveillance combining HOG-HCT and deformable part models, In Proceedings of the 2012 International Conference on Wavelet Analysis and Pattern Recognition, Xian, 15-17 July, 2012.
11. Afzal, A. G.; Wagan, A.: Exploring Local Features and the Bag-of-Visual-Words Approach for BioImage Classification. In ACM Conference on Bioinformatics, Computational Biology and Biomedical Informatics (ACM BCB), 2013, Washington, DC, s. 694 – 695.
12. Yang, J.; Jiang, Y.G.; Yu-Gang, H.; Ngo, C. W.: Evaluating bag-of-visual-words representations in scene classification. In Proceedings of the international workshop on Workshop on multimedia information retrieval, 2007, Augsburg, Bavaria, Germany.

13. Biederman, I.: Recognition by Components: A Theory of Human Image Understanding. In Psychology Review, Vol. 94, č. 2, 1987, s. 115–147.

14. Felzenszwalb, P.; Girshick, R.; McAllester, D.; Ramanan, D.: Object Detection with Discriminatively Trained Part Based Models. In IEEE Transactions on Pattern Analysis and Machine Intelligence, ročník 32, č. 9, 2010.

15. Malisiewicz, T.; Abhinav, G.; Efros, A. A.: Ensemble of exemplar-SVMs for object detection and beyond. In Computer Vision (ICCV), 2011 IEEE International Conference, IEEE, 2011, s. 89-96.

16. McCann, S.; Lowe, D. G.: Local naive bayes nearest neighbor for image classification. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference, IEEE, 2012, s. 3650-3656.

17. Burges, J.C.C.: A Tutorial on Support Vector Machines for Pattern Recognition, In Data Mining and Knowledge Discovery 2, 1998, s. 121-167.

18. Torralba, A.; Murphy, K. P.; Freeman, W. T.; Rubin, M. A.: Context-based vision system for place and object recognition. In Computer Vision, 2003. Proceedings. Ninth IEEE International Conference, IEEE, 2003, s. 273-280.

19. Oliva, A.; Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. International journal of computer vision, ročník 42, č. 3, 2001, s. 145-175.

20. Ballard, D. H.: Generalizing the Hough transform to detect arbitrary shapes. Pattern recognition, ročník 13, č. 2, 1981, s. 111-122.

21. Duda, R. O.; Hart, P. E.: Use of the Hough Transformation to Detect Lines and Curves in Pictures, Comm. ACM, č. 15, 1972, s. 11‑15.

22. Hough, P.: Method and Means for Recognizing Complex Patterns, 1962, U. S. Patent 3069654.

23. Leibe, B.; Leonardis, A.; Schiele, B.: Combined object categorization and segmentation with an implicit shape model. In Workshop on statistical learning in computer vision, ECCV, ročník 2, č. 5, 2004, s. 7.

24. Hartigan, J. A.; Wong, A. M.: Algorithm AS 136: A k-means clustering algorithm. Applied statistics, 1979, s. 100-108.

25. Breiman, L.: Random forests. In Machine learning, ročník 45, č. 1, 2001, s. 5-32.

26. Cover, T. M.; Thomas, J. A.: Entropy, relative entropy and mutual information. In Elements of Information Theory, 1991, s. 12-49.

27. Tresp, V.; Neuneier, R.; Ahmad, S.: Efficient methods for dealing with missing data in supervised learning. In Advances in neural information processing systems, Morgan Kaufmann Publishers, 1995, s. 689-696.

28. Neubeck, A.; van Gool, L.: Efficient non-maximum suppression. 18th International Conference on Pattern Recognition, ICPR 2006, s. 850-855.

29. Vygen, J.: Approximation Algorithms for Facility Location Problems. Lecture notes from Research Institute for Discrete Mathematics, University of Bonn, Germany, 2005.

30. Benenson, R.; Mathias, M.; Timofte, R.; van Gool, L.: Pedestrian detection at 100 frames per second. In Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, s. 2903-2910.

31. Brown, M.; Szeliski, R.; Winder, S.: Multi-image matching using multi-scale oriented patches, International Conference on Computer Vision and Pattern Recognition, 2005, s. 510-517.

32. Grauman, K.; Leibe, B.: Synthesis Lectures on Computer Vision # 1, University of Texas at Austin, 2011.

33. Berwick, R.: Lectures on Artifical Intelligence, „An Idiot's Guide to Support Vector Machines (SVMs)", Massachusetts Institute of Technology, 2003.

Přílohy

DVD s textem této práce, zdrojovým kódem implementovaného detektoru, trénovací datovou sadou, použitými testovacími sadami a ukázkami detekce.