

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

PROJEKT IZ RASPOZNAVANJA UZORAKA
kontrolna točka

**Detekcija pješaka u urbanim
okruženjima korištenjem značajki
temeljenih na teksturi i boji**

Iva Miholić, Gustav Matula, Tomislav Kiš

Zagreb, siječanj 2015.

SADRŽAJ

1. Detekcija pješaka u urbanim okruženjima	1
1.1. Opis projektnog zadatka	1
1.2. Pregled i opis postojećih rješenja	1
1.3. Konceptualno rješenje zadatka	3
1.3.1. Pregled značajki temeljenih na teksturi i boji	3
1.3.2. Redukcija dimenzije	4
1.3.3. Plan arhitekture sustava računalnog vida	5
2. Postupak rješavanja zadatka	6
2.1. Učitavanje primjera za trening	6
2.2. Ekstrakcija značajki	6
2.2.1. Značajke teksture	6
2.2.2. Značajke HOG-a i histograma boja	9
2.3. Treniranje	9
2.4. Treniranje druge faze klasifikatora	10
2.4.1. <i>Partial Least Squares</i> (PLS) kao metoda redukcije prostora značajki	10
2.4.2. Podešavanje parametara	11
2.5. Treniranje prve faze klasifikatora	12
3. Ispitivanje rješenja	13
3.1. Ispitna baza	13
3.2. Rezultati učenja i ispitivanja	14
3.3. Analiza rezultata	14
4. Opis programske implementacije rješenja	16
4.1. Treniranje klasifikatora	16
4.2. Program za evaluaciju istreniranog klasifikatora	17

4.3. Program za klasifikaciju pojedine slike	17
5. Zaključak	18
6. Literatura	19

1. Detekcija pješaka u urbanim okruženjima

1.1. Opis projektnog zadatka

Detekcija pješaka u urbanim okruženjima problem je povezan sa automobilskom industrijom, sigurnošću u prometu, sigurnosnim nadziranjem i robotikom. Rješava se kao detekcija objekta u okviru područja računalnog vida. Ovaj projektni zadatak obuhvaća izgradnju detektora pješaka na fotografijama iz urbanih okruženja korištenjem značajki temeljenih na bridovima, teksturi i boji.

1.2. Pregled i opis postojećih rješenja

Pregled najznačajnijih rješenja ovog problema dan je u [1]. Trenutačno postoje dva glavna pristupa tom problemu. Prva klasa metoda obuhvaća detektore za pojedine dijelove tijela koji se zatim kombiniraju u detektor čvojeke - pješaka dok druga klasa prihvaća problemu statističkom analizom kombinirajući mnogobrojne značajke unutar detekcijskog prozora u klasifikator.

Prvi značajniji napredak u detekciji pješaka bila je primjena *VJ* detektora objekata [8] na ovaj problem. Detektori temeljeni na histogramu usmjerenih gradijenata (engl. *Histogram of Oriented Gradients, HOG*) [2] uz linearni ili nelinearni skup potpornih vektora, postigli su značajne rezultate. No, primjena samo HOG značajki vodi do broja pogrešno pozitivno klasificiranih primjera čime se može stati na kraj uvodeći u sustav značajke temeljene na svojstvima boje, tekstura i oblika.

Više o HOG pristupu bit će riječi u sljedećim poglavljima jer ćemo ga koristiti kao kostur našeg prostora značajki.

Od ostalih rezultata, potrebno je izdvojiti postupke temeljene na modelu rastavljivih dijelova (engl. *Deformable Part Models, DPM*) u kojem se detekcija



Slika 1.1: Fotografija koja je klasificirana detektorom pješaka metodom kliznog prozora. Žuti okviri prikazuju okvire onih prozora koji su klasificirani kao prikaz pješaka.

dijelova tijela sažima u detekciju cijelog pješaka te nelinearne postupke učenja temeljenih na neuronskim mrežama i stabilima odluke [1]. Takvi složeniji postupci uspoređeni su sa linearnim SVM-om uz HOG i druge značajke nisu dali značajno bolji rezultat upućujući na korektnost naše odluke o arhitekturi detektora.

Pristup na koji ćemo se fokusirati koristi metodu skalabilnog kliznog prozora. Unutar prozora na fotografiji određuju se značajke te se skup piksela unutar prozora binarno klasificira kao fotografija pješaka. Prozor zatim "putuje" po fotografiji testirajući druge skupove piksela. Slika se onda može dodatno skalirati (efektivno skalirajući prozor) nakon čega se ponavlja isti postupak. Primjer klasificirane fotografije ovakvim postupkom vidljiv je na slici 1.1. Problem takvog pristupa je ignoriranje konteksta oko okvira koji se promatra što se uspješno rješava uvođenjem novih značajki, odnosno novih informacija o prozoru u sustav.

Do sada su najviše korištene značajke temeljene na informaciji o bridovima, boji, teksturi lokalnim oblicima te svojstvima gradijenta i kovarijance. Dodavanje novih značajki pokazao se kao vrlo uspješan način poboljšanja rada detektora pješaka, no proširenje prostora značajki može biti problematično za klasične algoritme učenja što se rješava redukcijom prostora značajki metodom Fisherove diskriminantne analize, analizom glavnih komponenti ((engl. *Principal Component Analysis*, *PCA*), ili postupkom parcijalnih najmanjih kvadrata (engl. *Partial Least Squares*, *PLS*) [6].

1.3. Konceptualno rješenje zadatka

1.3.1. Pregled značajki temeljenih na teksturi i boji

Promatrajući pješake na fotografijama, možemo uočiti karakteristike koje ih razlikuju od, na primjer, pozadine: vertikalni bridovi uz rub siluete, često uniformna boja odjeće, razlika u teksturi iste naspram teksturi pozadine, boja kože u području glave, ruku i nogu... Motivacija za komplementiranjem značajki boje i texture značajkama vezanim uz bridove (HOG deskriptori) tako se prirodno nameće kao bolji opis sustava.

Značajke temeljene na teksturi i boji obično se koriste kao nadopuna značajkama fokusiranim na bridove, za koju ćemo koristiti popularnu HOG metodu Dalal i Triggsa [2]. Njihova se metoda pokazala dobrom na više baza podataka, ali promatranjem isključivo gradijenata potencijalno odbacujemo korisne izvore informacija, što vodi lažnoj pozitivnoj klasifikaciji. Primjerice tekstura nam dosta pomaže kod prepoznavanja odjeće, ali i pozadine, a boja kod prepoznavanja boje kože.

Značajke temeljene na teksturi

Vjerojatno najpoznatija metoda ekstrakcije značajki koje opisuju teksturu potječe iz Haralickovog rada [5] još iz 1979. Za opis texture koristi takozvanu *co-occurrence* matricu, iz koje se zatim računaju same značajke, kao što su npr. korelacija, srednje vrijednosti i razne mjere entropije.

Osnovna ideja iza matrice jest određivanje vjerojatnosti susjedstva svih parova intenziteta boje. Tako primjerice horizontalna matrica H kao element $h_{i,j}$ sadrži vjerojatnost da je nasumični par horizontalno susjednih piksela ima intenzitete redom i i j . Matrica se tipično računa za horizontalni, vertikalni, te oba dijagonalna smjera (smjerovi 0° , 45° , 90° , 135°). Još jedan parametar koji se može koristiti jest udaljenost d , pa tako matrica $V^{(d)}$ uzima u obzir parove koji su vertikalno susjedni na udaljenosti d .

Svaka tako dobivena matrica promatra se kao matrica zajedničke distribucije vjerojatnosti $p(i, j)$, te iz nje računamo značajke. Tipični primjeri značajki koje Haralick navodi u svome radu su primjerice:

– varijanca:

$$\sum_i \sum_j (i - \mu)^2 p(i, j)$$

- korelacija:

$$\frac{\sum_i \sum_j (ij)p(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y}$$

- entropija:

$$-\sum_i \sum_j p(i, j) \log(p(i, j))$$

Značajke temeljene na boji

U [6] se za iskorištavanje informacije sadržane u boji koristi jednostavno proširenje HOG histograma. Tijekom izgrade HOG histograma, promatramo koji boji pripada gradijent s najvećom normom te se formira trostupčani histogram frekvencija svake od tri boje za trenutni prozor detekcije. Tako uz histogram gradijenata promatramo i histogram boja.

1.3.2. Redukcija dimenzije

Dovođenje mnogo značajki u sustav, odnosno visoka dimenzionalnost vektora značajki, može predstavljati problem za klasične tehnike učenje poput SVM-a, posebice uz često mali skup primjera za učenje.

Pretpostavimo (slično kao u [6]) da je detekcijski prozor podijeljen na preklapajuće blokove u iz kojih se zatim ekstrahiraju opisane značajke. Vektori značajki za sve blokove istog prozora konkatenuiraju se u vektor značajki prozora. Velika dimenzija takvog prostora značajki koju dobivamo promatrajući, uz gradijente, teksturu i boju, predstavlja problem za klasične metode strojnog učenja. No, zbog podijele prozora detekcije u blokove, značajke se ekstrahiraju iz susjednih blokova, što neizbježno vodi do sličnih značajki u bliskim blokovima te samim time i do kolinearnosti. Tako ima smisla iskoristiti nekakvu tehniku redukcije dimenzije, kao što su PCA (engl. *Principal Component Analysis*) ili FDA (engl. *Fisher Discriminant Analysis*). [6] koristi PLS (*Partial Least Squares*), inače regresijsku metodu, praktičnu i za redukciju dimenzije skupa značajki zbog njene brzine i primjenjivosti na ovaj specifičan model.

Nakon redukcije dimenzije za treniranje klasifikatora možemo koristiti metode učenja kao što je SVM (*Support Vector Machine*) koje bi na prostoru s previše dimenzija bile neupotrebljive.

1.3.3. Plan arhitekture sustava računalnog vida

Sustav će se kao i obično sastojati od dvije osnovne komponente: treniranja i testiranja te primjene.

Treniranje klasifikatora



Slika 1.2: Dijagram treniranja SVM klasifikatora

Treniranje se sastoji od nekoliko koraka koji su okvirno prikazani na slici 1.2. Za svaku sliku iz baze računaju se gradijenti (horizontalni i vertikalni smjer). Zatim po slici pomičemo klizni prozor, dijelimo ga na preklapajuće blokove i za svaki blok računamo histogram gradijenata i histogram boja. Prozor potom ponovno dijelimo na blokove (ne nužno istih dimenzija kao u prethodnom koraku), te računamo *co-occurrence* matrice za sva četiri smjera i udaljenost $d = 1$ matricu, iz kojih dobivamo značajke koje opisuju teksturu. U sljedećem koraku reduciramo dimenziju prostora značajki. Konačno, na tako pojednostavljenim primjerima treniramo SVM. U tijeku postupka parametri metode redukcije i SVM-a dobivaju se metodom 10-koračne križne validacije (*10-fold-cross-validation*).

Testiranje klasifikatora

Novotreniranom klasifikatoru ćemo zatim testirati sposobnost generalizacije i pre-treniranost. Pregled i analiza testiranja dana je u zasebnom poglavlju.

Primjena klasifikatora

Slično kao kod treniranja promatramo klizni prozor i za njega računamo značajke, kojima zatim reduciramo dimenziju te ih dajemo kao ulazne podatke klasifikatoru. Ukoliko klasifikator procijeni da se radi o pješaku, dojavljujemo poziciju trenutnog kliznog prozora.

2. Postupak rješavanja zadatka

2.1. Učitavanje primjera za trening

Na početku treniranja klasifikatora, učitava se baza slika u boji INRIA train_64x128_H96 sastavljena od 2416 pozitivnih primjera slika veličine 64x128 piksela te negativnih slika različitih dimenzija. Kako bi se postigla veličina 64x128 piksela, sa svake je slike eliminiran postojeći crni okvir širine 16 bita. Iz negativnih primjera učitanih slika slučajno se izabere 10000 prozora veličine 64x128 piksela koje ćemo koristiti kao negativne primjere u treningu. Zatim se iz slika ekstrahiraju značajke. Nakon ekstrakcije značajki, učitane slike oslobađaju se iz memorije.

2.2. Ekstrakcija značajki

Za dani prozor, odnosno sliku veličine 64x128 piksela, ekstrahiraju se značajke tako da najprije podijelimo prozor na preklapajuće blokove, ekstrahiramo značajke nad njima i konkatenujemo ih u vektor značajki za ulazni prozor pritom pamteći koje su značajke došle iz kojeg bloka.

2.2.1. Značajke teksture

Iz svakog bloka veličine 16, odnosno 32 piksela dobivenih pomakom po prozoru za 8 odnosno 16 piksela, računaju se 4 matrice susjedstva (engl. *co-occurrence matrices*): matrice zadužene za horizontalne i vertikalne susjede te matrice zadužene za susjede na obje dijagonale. Svaka matrica uzima u obzir učestalost parova najbližih susjeda u pojedinom kanalu boja za svoj smjer čime ćemo zapravo dobiti 12 matrica susjedstva. Za značajke teksture koristimo *HSV* (engl. *hue*, *saturation*, *value*) model boja.

HSV model boja filtrira tri kanala: ton boje (engl. *hue*) određen stupnjem (0° - 360°), čistoću boje (engl. *saturation*) određenu vrijednošću od 0 do 1 koja

određuje koliko je bijele dodano u boju te svjetlina boje (engl. *brightness*) također sa vrijednosti od 0 do 1 gdje 0 označuje crnu boju. Sa tim se opisom *HSV* čini kao najbolji model za temelj ekstrakcije značajki teksture. Prije računanja matrica susjedstva, vrijednosti sva 3 kanala diskretiziraju se na vrijednosti od 1 do $N_g = 16$.

Iz 12 matrica susjedstva ekstrahira se 13 Haralickovih značajki teksture, dakle sveukupno 156 značajki po bloku. Neka je P jedna od matrica susjedstva. Ona je dimenzija $N_g \times N_g$. Vrijednost $P(i, j)$ daje frekvenciju pojavljivanja parova vrijednost i i j kao (prvih) susjeda u diskretiziranom kanalu boje. Formiramo novu matricu p dimenzija $N_g \times N_g$ sa aproksimacijama parova diskretiziranih vrijednosti:

$$R = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} P(i, j)$$

$$p(i, j) = \frac{P(i, j)}{R}$$

Slijedi definicija 13 Haralickovih značajki teksture. Vrijednosti funkcija f_1, f_2, \dots, f_{13} koriste se kao značajke našeg modela.

Suma kvadrata:

$$f_1 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j)^2$$

Varijanca:

$$f_2 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i - \mu)^2 p(i, j)$$

gdje je μ srednja vrijednost $p(x, y)$.

Inverzni diferencijski moment:

$$f_3 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{1}{1 + (i - j)^2} p(i, j) \quad (2.1)$$

Entropija:

$$f_4 = - \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \log(p(i, j))$$

Neka su

$$p_x(i) = \sum_{j=1}^{N_g} p(i, j)$$

$$p_y(j) = \sum_{i=1}^{N_g} p(i, j).$$

te $\mu_x, \mu_y, \sigma_x, \sigma_y$ srednje vrijednosti i standardne devijacije pripadnih vjerojatnosti.

Korelacija:

$$f_6 = \frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (ij)p(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y}$$

Neka je

$$p_{x-y}(k) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \text{ za } |i - j| = k, k \in \{0, 1, \dots, N_g - 1\}$$

Kontrast:

$$f_7 = \sum_{k=0}^{N_g-1} k^2 p_{x-y}(k)$$

Varijanca p_{x-y} :

$$f_8 = \text{Var}(p_{x-y})$$

Entropija p_{x-y} :

$$f_9 = - \sum_{i=0}^{N_g-1} p_{x-y}(i) \log(p_{x-y}(i))$$

Neka je

$$p_{x+y}(k) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \text{ za } i + j = k, k \in \{2, 3, \dots, 2N_g\}$$

Očekivanje p_{x+y} :

$$f_{10} = \sum_{i=2}^{2N_g} i p_{x+y}(i)$$

Varijanca p_{x+y} :

$$f_{11} = \sum_{i=2}^{2N_g} (i - f_{10})^2 p_{x+y}(i)$$

Entropija p_{x+y} :

$$f_{12} = - \sum_{i=2}^{2N_g} p_{x+y}(i) \log(p_{x+y}(i))$$

Neka su definirane entropije

$$HXY1 = - \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \log(p_x(i)p_y(j))$$

$$HXY2 = - \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p_x(i)p_y(j) \log(p_x(i)p_y(j))$$

te

$$f_{12} = \frac{f_9 - HXY1}{\max(HX, HY)}$$

$$f_{13} = \sqrt{1 - \exp(-2(HXY2 - f_9))}$$

mjere korelacije.

Za svaki blok u prozoru dobivamo tako $13 * 3 * 4 = 156$ značajki što je ukupno 24960 značajki tekstone u 160 blokova jednog prozora.

2.2.2. Značajke HOG-a i histograma boja

Za ekstrakciju značajki HOG-a i boje koristimo podijelu na kvadratne blokove širine 12, 16, 20, 26, 32, 40, 48 i 64 piksela koji također 'klize' po prozoru za pola duljine svoje širine. Za svaki kanal boje (u ovom slučaju to su kanali modela 'RGB') i svaki blok računaju se značajke koje se zatim konkatenuiraju u vektor značajki prozora pamteći pritom koja značajka pripada kojem bloku.

Nagli prijelazi odnosno gradijenti bloka računaju se konvoluiranjem jednog kanala bloka sa jezgrenom kvadratnom matricom neparnih dimenzija. Jezgrena matrica klizi po bloku i za svaki piksel izračunava pripadajući prijelaz. Gradijenti se računaju koristeći prijelazne matrice za smjerove 0° i 90° . Konvolucijske matrice za navedene smjerove su:

0 stupnjeva:

$$K_0 = \begin{pmatrix} -1 & 0 & 1 \end{pmatrix}$$

90 stupnjeva:

$$K_{90} = \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} \quad (2.2)$$

. Vrijednosti gradijenata se diskretiziraju u interval $[1, 12]$ te se za značajke uzima frekvencija svake diskretizirane vrijednosti (12 značajki po svakom kanalu boje u bloku). Dodatno, tijekom postupka se promatra u kojem je kanalu gradijent trenutačno najviši te se shodno tome ažurira histogram boja sa 3 koša koji mjere frekvenciju odabira svaka od 3 kanala boje kao onog sa najvišom vrijednošću gradijenta tijekom postupka.

Za jedan blok tako imamo 39 značajki, a za cijeli prozor, odnosno 558 blokova, imamo 21762 značajki.

2.3. Treniranje

Ukupan broj ekstrahiranih značajki za jedan prozor, 46722 značajki, prevelik je za direktno treniranje SVM klasifikatora te je prostoru značajki potrebno smanjiti dimenziju. No, i samo računanje velikog broja značajki i preslikavanje u prostor manje dimenzije pokazao se kao vremenski zahtjevan postupak (računanje svih značajki za jedan prozor u našoj implementaciji traje otprilike 0.1 sekundu) te ćemo iz tog razloga trenirati dva klasifikatora: prvi, jednostavniji model koji je osjetljiv na pogrešno pozitivno detektiranje negativnih primjera te drugi, složeniji

model. Za vrijeme klasifikacije, prvi će model služiti kao filter za prozore koje želimo propustiti u drugi, složeniji i računalno zahtjevniji model. Najprije ćemo opisati postupak treniranja složenijeg klasifikatora jer treniranje svake od faza međusobno je nezavisno, a metoda treniranja složenijeg klasifikatora je jednostavnija.

2.4. Treniranje druge faze klasifikatora

Na ulazu imamo 2416 vektora značajki pozitivno klasificirana prozora te 10000 vektora značajki negativno klasificiranih prozora koje smo slučajno izabrali iz skupa negativnih primjera u bazi podataka. Skup negativno klasificiranih prozora slučajno podijelimo po pola. Početni skup za treniranje sastojat će se od 2416 pozitivno klasificirana te 5000 negativno klasificirana primjera.

Treniramo SVM sljedećim iterativnim postupkom:

- 0. podesi parametar dimenzije reduciranog prostora značajki
- 1. nauči ostale parametre metode PLS i smanji dimenziju prostora značajki
- 2. treniraj SVM uz podešavanje parametara u reduciranom dimenzijskom prostoru
- 3. klasificiraj skup za trening i dodatne negativne primjere dane na ulazu
- 4. negativne primjere koji su pogrešno klasificirani, a nisu u skupu za trening dodaj u skup za trening
- 5. vrati se na 1. do zadovoljavajućeg smanjenja progreske klasifikacije

Iz postupka saznajemo veličinu dimenzije reduciranog prostora značajki, parametre PLS-a koji će se spremiti u datoteku te parametre SVM-a koji će se također spremiti u datoteku.

2.4.1. *Partial Least Squares* (PLS) kao metoda redukcije prostora značajki

Kod metode parcijalnih najmanjih kvadrata (*Partial Least Squares*, PLS) konstruiramo set prediktorskih (ili latentnih) značajki uzimajući u obzir klasifikaciju ulaznih primjera. Detaljni opis postupka može se naći u [7], a ovdje slijedi kraće objašnjenje.

Neka je $\mathcal{X} \subset \mathbf{R}^m$ prostor vektora značajki dimenzije m , a $\mathcal{Y} \subset \mathbf{R}$ prostor jednodimenzionalnih vektora - klasifikacija primjera. Za n primjera, PLS postupkom nalazimo sljedeću dekompoziciju matrica $\mathbf{X}(n \times m)$ i $\mathbf{y}(n \times 1)$ čiji su retci iz \mathcal{X} odnosno \mathcal{Y} :

$$\begin{aligned}\mathbf{X} &= \mathbf{T}\mathbf{P}^T + \mathbf{E} \\ \mathbf{y} &= \mathbf{U}\mathbf{q}^T + \mathbf{f}\end{aligned}$$

Matrice \mathbf{T} i \mathbf{U} su dimenzija $(n \times p)$ i sadrže vektore značajki nove dimenzije p . Matrica \mathbf{P} dimenzije $(m \times p)$ i vektor \mathbf{q} dimenzije p označavaju koje su se značajke očuvale i kojim se poretком nalaze u \mathbf{T} odnosno \mathbf{U} , a matrica \mathbf{E} i vektor \mathbf{f} su rezidue.

Primjernom algoritma NIPALS (engl. *non linear iterative partial least squares*) želimo konstruirati skup težinskih vektora $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_p\}$ takav da

$$[\text{cov}(\mathbf{t}_i, \mathbf{u}_i)]^2 = \max_{|\mathbf{w}_i|=1} [\text{cov}(\mathbf{X}\mathbf{w}_i, \mathbf{y})]^2$$

gdje je \mathbf{t}_i i -ti stupac matrice \mathbf{T} , a \mathbf{u}_i i -ti stupac matrice \mathbf{U} . $\text{cov}(\mathbf{t}_i, \mathbf{u}_i)$ je procjena kovarijance između ta dva vektora. Postupak se ponavlja dok se ne postigne željena dimenzija p .

Tako se dobiva ortogonalni skup vektora \mathbf{w}_i maksimizirajući pritom kovarijancu između elemenata u \mathbf{X} i \mathbf{y} . U usporedbi s PCA, PLS uzima u obzir varijance ulaznih značajki kao i predloženu klasifikaciju. Fischerova diskriminantna analiza, s druge strane, također uzima u obzir klasifikaciju, ali redukcija značajki dovodi do dimenzije $c - 1$ gdje je c broj klasa.

Redukcija dimenzije na kraju se odvija projiciranjem vektora značajki dimenzija m na skup težinskih vektora \mathbf{W} rezultirajući novim vektorom značajki dimenzije p .

2.4.2. Podešavanje parametara

Svi parametri posešavaju se u postupku 10-koračne unakrsne validacije. U postupku najprije podijelimo skup za trening na 10 jednakih dijelova. Postupak se ponavlja 10 puta tako da uzmemo jednu 10 primjera kao skup za validaciju, treniramo klasifikator na ostalim primjerima te na skupu za validaciju odredimo uspješnost klasifikacije za dani parametar. Onu vrijednost parametra za koji je pogreška generalizacije u prosjeku najmanja uzmemo za konačnu vrijednost.

2.5. Treniranje prve faze klasifikatora

Vremenski složenu ekstrakciju svih značajki za jedan prozor kao i redukciju istih na prostor manjih dimenzija, želimo optimizirati uvođenjem predfaze koja će birati prozore za ulaz u složeniji detektor. To ćemo učiniti odabirom podskupa blokova prozora najvišeg ratinga uz čije ćemo značajke trenirati jednostavniji klasifikator. Birajući manji skup blokova za klasifikaciju, ne trebamo računati značajke za svaki blok prozora, već samo za one odabrane. Trening i validacija odvija se na skupu od 2416 pozitivnih i 10000 negativnih prozora.

Značajnost značajke na PLS projekciji (skraćeno VIP - *Variable importance on projection*) [9] može se definirati za j -tu značajku kao

$$VIP_j = \sqrt{m \sum_{k=1}^p b_k^2 w_{jk}^2 / \sum_{k=1}^p b_k^2}$$

gdje je m ulazni broj značajki, p broj značajki dobiven redukcijom w_{jk} j -ti element k -tog težinskog vektora $\mathbf{w}_k \in \mathbf{W}$ te b_k regresijska težina k -te latentne varijable [7], $b_k = \mathbf{u}_k^T \mathbf{t}_k$.

Neka je blok definiran relativnom pozicijom u prozoru, svojom duljinom i širinom te vrsti značajki koje smo ekstrahirali iz njega. S obzirom da nam se prozori u skupu za treniranje međusobno ne preklapaju, možemo definirati klasu svakog bloka. Neka je klasa bloka ista klasi prozora u kojem se blok nalazi. Kako bi rangirali blokove po važnosti informacije koju unose u sustav, nad svakim blokom trenirat ćemo PLS i odabrati značajku predstavnicu tog bloka. To će ona značajka koja ima najviši VIP score između svih značajki istoga bloka.

Nakon odabira značajki predstavnica svakog bloka, konstruiramo novi vektor značajki za prozore koji se sastoji samo od značajki predstavica svakog bloka. VIP rejting značajke predstavnice bloka određuje rejting tog bloka. Ostaje nam sortirati blokove silazno po rejtingu te odabrati broj blokova za klasifikator.

Koliko ćemo blokova uzeti za klasifikaciju biramo 10-koračnom unakrsnom validacijom. Na kraju također podesimo parametre SVM-a te zapišemo parametre PLS-a i SVM-a u datoteke. Svi se parametri podešavaju na temelju 0–1 gubitka, $L = (fp + fn)/(p + n)$, na skupu za validaciju.

3. Ispitivanje rješenja

3.1. Ispitna baza

Za treniranje i verifikaciju rješenja koristit ćemo skup podataka INRIA [3]. To je skup fotografija urbanog okruženja u boji i pripadnih anotacija. Za svaku fotografiju zabilježen je skup graničnih prozora (engl. *bounding window*) unutar kojih se nalaze prikazi uspravnih osoba.

INRIA se do sada često koristio za trening detektora zbog raznolikosti pozadinskih okruženja osoba na slikama i točnosti anotacija [1]. Za razliku od ostalih javno dostupnih baza podataka za detekciju pješaka, ove fotografije nisu dobivene iz videa te su relativno visoke kvalitete i fotografirane su iz različitih točaka gledišta. U drugim bazama podataka, pješaci su većinom konecentrirani u centralnoj horizontali fotografije jer je ista dobivena iz vozačeve perspektive.



Slika 3.1: Primjeri fotografija iz baze podataka INRIA.

Baza slika INRIA sadrži slike iz nekoliko različitih izvora:

- *Graz 01* skup podataka sa pridodanim bilješkama
- slike iz osobne kolekcije samog autora koje su zbog svoje veličine u originalu izrezane tako da prikazuju samo osobu
- slike sa servisa Google Images

Za treniranje koristimo podskup baze 'train_64x128_H96' koji čini 2416 pozitivno klasificiranih slika veličine 64×128 piksela kojima je dodan crni okvir od 16 piksela te 1218 negativnih slika različitih veličina iz koji slučajno odabiremo negativne okvire za trening.

Za testiranje koristim podskup baze 'test_64x128_H96' koji čini 1132 pozitivno klasificirane slike također veličine 64×128 piksela s crnim okvirom od 16 piksela te 453 negativno klasificirane slike različitih veličina. Crne okvire ćemo rijekom treninga i testiranja na pozitivno klasificiranim slikama zanemariti.

3.2. Rezultati učenja i ispitivanja

Rezultate učenja i ispitivanja dajemo za direktnu primjenu druge faze klasifikatora. Za primjenu implementirane i trenirane prve faze klasifikatora potrebne su daljnje validacije koje nismo bili u mogućnosti izvesti.

Tijekom unakrsnog učenja i validacije, SVM sa radijalnim baznim jezgrama i 20 značajki imao je najbolje svojstvo generalizacije. Dalje povećanje prostora značajki dovodilo je do pretreniranosti. Eksperimenti sa omjerom pozitivnog i negativnog broja klasificiranih prozora na trening uputili su nas na korištenje broja negativnih slika približno jednakim kao broj pozitivnih slika (2418 : 2500) od početnog pretpostavljenog omjera (2418 : 10000) jer klasifikator tada ima tendenciju klasificirati u negativnu klasu.

Testiranje provodimo nad 1332 pozitivna i 2000 negativnih primjera. Za najbolje trenirani klasifikator:

- postotak pogrešno klasificiranih pozitivnih primjera:
- postotak pogrešno klasificiranih negativnih primjera:
- postotak pogrešno klasificiranih primjera:

Za klasifikaciju jednog prozora treba u prosjeku 0.2 sekunde.

3.3. Analiza rezultata

U tijeku klasifikacije nepoznate slike u kojoj je potrebno kliziti prozor, u slučaju pozitivne klasifikacije, više se prozora (0 do 16 piksela najbliže udaljenosti) klasificiraju pozitivno. U tom slučaju pomoću funkcije `groupRectangles` "spajamo bliske prozore" kako bi slika rezultata bila preglednija.

Za sliku nepoznate veličine kada je potrebno kliziti prozor po slici, proječno vrijeme evaluacije jednog prozora znatno usporava implementaciju što bi se riješilo dodavanjem prve faze klasifikatora. U tijeku klasifikacije slike nepoznate veličine klizimo prozor za 8 piksela po slici, prethodno je skalirajući više puta.

4. Opis programske implementacije rješenja

Cjelokupna implementacija rješenja može se pronaći ovdje. Programska implementacija nalazi se u repozitoriju 'src'. Implementacija ekstrakcije značajki teksture nalazi se u repozitoriju 'src/texture', a ekstrakcija značajki boje i HOG-a nalaze se u 'src/hogcolor'.

Redukcijski postupak implementiran je u datotekama 'src/pls/pls.cpp' i 'src/pls/model.cpp'. Za samu implementaciju PLS postupka koristimo izmijenjenu verziju koda korištenu za [6] dohvaćenu ovdje.

4.1. Treniranje klasifikatora

Sami postupci treniranja prve i druge faze klasifikatora dani su u datoteci 'src/pls/plsanje.cpp'. Metoda 'plsPerBlock' predstavlja postupak treniranja prve faze, a metoda 'plsFull' predstavlja postupak treniranja druge faze. Sučelje za pokretanje treniranja implementirano je u 'src/train.cpp'.

Želimo li trenirati prvu fazu klasifikatora, pokrećemo program train s parametrom '1':

```
# ./train 1
```

. Izlaz programa biti će dvije datoteke: 'plsModel1' i 'svmModel1.xml' u kojima su spremljeni parametri modela za redukciju značajki metodom PLS te parametri SVM modela.

Želimo li trenirati drugu fazu klasifikatora, pokrećemo program train s parametrom '2':

```
# ./train 2
```

. Izlaz programa biti će dvije datoteke: 'plsModel2' i 'svmModel2.xml' u kojima su spremljeni parametri modela za redukciju značajki metodom PLS te parametri SVM modela.

4.2. Program za evaluaciju istreniranog klasifikatora

Program za evaluaciju istreniranog klasifikatora implementiran je u 'src/test.cpp'. Program čita datoteke u kojima su zapisani klasifikatori za prvu i drugu fazu klasifikacije ('plsModel1', plsModel2', 'svmModel1.xml', svmModel2.xml') te provodi evaluaciju cijelokupnog procesa klasifikacije nad skupom testnim primjera iz INRIA baze podataka. Program će ispisati broj pogrešnih pozitivnih i negativnih klasifikacija, broj točnih pozitivnih i negativnih klasifikacija, omjer pogrešno negativnih klasifikacija među podskupom pozitivnih primjera (engl. *miss rate*) te omjer pogrešno pozitivnih klasifikacija među podskupom negativnih primjera (engl. *false positives per window*).

Program također stvara datoteku 'fails' sa popisom putanja do slika koje su pogrešno klasificirane.

4.3. Program za klasifikaciju pojedine slike

Program za klasifikaciju jedne slike nalazi se u datoteci 'src/classify.cpp'. Prvi argument komande linije programa jest putanja slike u kojoj se želi pronaći pješak / pješaci. Slika može biti proizvoljnog formata. Program skalira sliku više puta te pomiče prozor veličine 64×128 piksela po slici. Svaki takav prozor pokušava se klasificirati kao pješak. Na kraju izvođenja program prikazuje sliku sa prikazanim okvirima koje je klasificirao kao pješačke.

5. Zaključak

Dodavanjem značajki teksture i boje u sustav značajki podupiren HOG-om svakako dovodi nove informacije u sustav i čini model sposobnijim za generalizaciju. No, velika dimenzionalnost početnog prostora značajki dovodi do potrebe za redukcijom istog. Korištenjem metode kao što je PLS taj je postupak brz i uzima u obzir pretpostavljene klase primjera.

Trenirajući klasifikator, uočili smo pojavu znatno manje lažno negativnih klasifikacija s obzirom na problematiku klasificiranja samo HOG značajkama, no za veću preciznost, potrebno je posebno dobro podesiti parametre sustava.

Budući planirani rad svakako predstavlja dodavanje prve faze klasifikatora u konačan produkt zbog brzine same klasifikacije te dodatna ubrzanja klasifikatora druge faze. Naime, s obzirom na matrice \mathbf{P} i \mathbf{q} iz PLS komponenti, moguće je vidjeti koje se inicijalne značajke ne koriste za kreiranje značajki u reduciranom prostoru te takve značajke ne treba računati. Također, nakon potrebnih optimizacija, svakako je dobra ideja dodatno uvođenje novih značajki u sustav, s obzirom da, na primjeri, nismo koristili potpun sustav HOG značajki.

6. Literatura

- [1] Rodrigo Benenson, Mohamed Omran, Jan Hendrik Hosang, and Bernt Schiele. Ten years of pedestrian detection, what have we learned? [abs/1411.4304](#), 2014.
- [2] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.
- [3] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In Cordelia Schmid, Stefano Soatto, and Carlo Tomasi, editors, *International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 886–893, INRIA Rhône-Alpes, ZIRST-655, av. de l’Europe, Montbonnot-38334, June 2005.
- [4] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. 34(4):743–761, April 2012.
- [5] Robert M. Haralick, K. Sam Shanmugam, and Its’hak Dinstein. Textural features for image classification. (6), 1973.
- [6] William Robson Schwartz, Aniruddha Kembhavi, David Harwood, and Larry S. Davis. Human detection using partial least squares analysis.
- [7] Randall D Tobias et al. An introduction to partial least squares regression.
- [8] Paul Viola and Michael J. Jones. Robust real-time face detection. 57(2):137–154, May 2004.
- [9] S Wold, E Johansson, M Cocchi, et al. Pls: partial least squares projections to latent structures. 1993.