

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

Primjena strojno naučenih klasifikatora u računalnom vidu

Mateja Čuljak
Voditelj: Siniša Šegvić

Zagreb, 3. lipnja 2009.

Sadržaj

1. Uvod	2
2. Pronalaženje objekata temeljeno na strojnom učenju	2
3. Opis metode Viole i Jonesa	4
3.1. Tri glavna doprinosa metode	4
3.2. Haarove značajke	4
3.3. Integralna slika	5
3.4. Gradnja klasifikatora strojnim učenjem	6
3.4.1. Algoritam Adaboost	6
3.5. Ulančavanje boostanih klasifikatora	8
4. Primjena metode Viole i Jonesa	9
5. Evaluiranje Lienhartovog izvornog koda	9
5.1. Biblioteka <i>OpenCV</i>	9
5.2. Korištenje biblioteke <i>OpenCV</i> za pronalaženje objekata	10
6. Zaključak	11

1. Uvod

Računalni vid je područje računarstva koje se bavi ekstrakcijom korisnih informacija iz slika ili video sekvenci, te sustavima koji takvu ekstrakciju omogućavaju. Zbog razvitka digitalnih kamera visoke rezolucije nastala je potreba za algoritmima za što bržu obradu sve masivnijih slika. Računalni vid je usko povezan s drugim područjima u računarstvu, kao npr. umjetna inteligencija, strojno učenje, robotika, raspoznavanje uzoraka, obrada signala, obrada slika itd. Vrlo je važna povezanost između biološkog i računalnog vida, tj. računalni vid često pokušava oponašati, te preuzima koncepte iz biološkog vida. Važnije teme koje istražuje računalni vid su pronalaženje dvodimenzionalnih ili trodimenzionalnih objekata, raspoznavanje objekata, detekcija događaja, rekonstrukcija scene, restauracija slika, praćenje pokreta itd.

Pronalaženje (eng. *detection*) objekata u računalnom vidu se bavi problemom nalaženja objekata koji pripadaju nekoj zadanoj klasi u slici ili video sekvenci. Dok je ljudima vrlo lako “intuitivno” detektirati objekt neovisno o njegovoj veličini, kutu gledanja, mogućoj rotaciji ili djelomičnoj zaklonjenosti, računalima to predstavlja problem, te taj problem treba predstaviti nekim algoritmom da bi on računalu bio shvatljiv. Najčešće se pronalaženje objekata primjenjuje za pronalaženje ljudskih lica, pješaka, nekih zadanih znakova itd. Postoji razlika između detekcije objekata i raspoznavanja objekata. Dok se raspoznavanje objekata (eng. *recognition*) bavi problemom određivanja koji iz niza zadanih objekata različite klase se nalazi na slici, detekcija objekata se bavi određivanjem nalazi li se na slici objekt isključivo jedne klase. Postoji više metoda kojima je moguće ostvariti učinkovite sustave za detekciju objekata.

2. Pronalaženje objekata temeljeno na strojnom učenju

Iako se sustavi za pronalaženje objekata mogu graditi koristeći doista različite pristupe, jedan od najopćenitijih pristupa je korištenje *strojnog učenja*. Ako imamo zadaću točnog detektiranja nekog objekta, i dostupne informacije o slici nad kojom radimo, susrest ćemo se s problemom kako dobro izraziti veliki skup pravila potreban za detektiranje, i hoće li biti moguće uopće ih izraziti u obliku koji računalu razumije. S druge strane, ako sustav naučimo korištenjem postupka strojnog učenja, tj. damo mu podatke o slikama koje su već prethodno obrađene i sasvim sigurno znamo sadrže li ili ne taj objekt, on će na temelju naučenog znanja uspješno detektirati objekte na novom skupu slika.

Važna prednost ovog pristupa je mogućnost primjene na razne klase objekata, samo je potrebno prikupiti podatke za učenje. Tako isti sustav možemo primijeniti za pronalaženje lica, pronalaženje prometnih znakova ili bilo koju drugu klasu objekata.

Pri korištenju ovakvog pristupa, postavljaju se pitanja kako dobro izabrati skup primjera za učenje da dobijemo što bolja pravila za pronalaženje zadanog objekta, te kako puno dobivenih “malih” pravila za pronalaženje kombinirati u jedno veće univerzalno pravilo?

Nekoliko je metoda kojima na ovo možemo odgovoriti, tj. nekoliko algoritama za gradnju *klasifikatora* kojima klasificiramo podatke. Klasifikatorski sustavi su sustavi temeljeni na skupu pravila (klasifikatora) koja određuju reakciju sustava na uvjete dane u njegovoj okolini. Ta pravila se uče na temelju primjera koji se nalaze u skupu za

učenje. Detekciju objekata tako možemo gledati kao zadaću gradnje klasifikatora, pri čemu se slika ili video sekvenca na kojoj tražimo objekte, razdvaja u više manjih podprozora, i odluka se donosi za svaki podprozor na temelju toga sadrži li on traženi objekt ili ne. Podprozori su uglavnom fiksne dimenzije (24×24 piksela najčešće).

Budući da se to pokazalo kao proces s velikom složenosti, umjesto direktnog rada s pikselima ulazne slike, ove metode rade sa određenim skupom drukčijih značajki te slike. Sustavi bazirani na tim značajkama će raditi znatno brže nego sustavi bazirani na pikselima. Vrijednosti dobivene takvim računom predstavljaju prisustvo ili odsustvo nekih karakteristika slike (kao što su rubovi, kutevi i sl.).

Navedimo neke od metoda strojnog učenja koje koriste klasifikatorske sustave:

Bayesov klasifikator: Algoritam pretpostavlja normalnu (Gaussovu) razdiobu značajki, te na temelju skupa za učenje gradi kovarijacijske matrice i koristi ih za predviđanje.

K-NN algoritam: tj. *K najbližih susjeda* ili *učenje po primjeru*. Klasifikacija se odvija na temelju usporedbe radnog uzorka i k baznih uzoraka (uzoraka koje smo koristili za *treniranje*) odabranih na temelju sličnosti s radnim uzorkom. Bazni uzorci su pohranjeni u P -dimenzijskom prostoru, pri čemu je P broj značajki (eng.: property) kojima možemo pridijeliti težine.

$$x = (\omega_0 x_0, \omega_1 x_1, \omega_2 x_2, \dots, \omega_p x_p)$$

Radni uzorak dodajemo u postojeći vektorski prostor, te tražimo K najbližih uzoraka, npr. možemo koristiti euklidsku udaljenost vektora:

$$d(x, y) = \sqrt{(x_0 - y_0)^2 + (x_1 - y_1)^2 + \dots + (x_p - y_p)^2}$$

Ako je unutar K uzoraka više pozitivnih nego negativnih radni uzorak proglašavamo pozitivnim, u suprotnog ga proglašavamo negativnim.

Metoda potpornih vektora (SVM): SVM se temelji na izgradnji hiperravnine u prostoru uzoraka koja dijeli pozitivne i negativne uzorke. Radni uzorak dodajemo u prostor uzoraka i ako se nalazi na pozitivnoj strani ravnine, uzorak proglašavamo pozitivnim, u suprotnom ga proglašavamo negativnim. Uzorke prikazujemo u P -dimenzijskom prostoru, pri čemu je P broj značajki (eng.: property) kojima možemo pridijeliti težine.

$$x = (\omega_0 x_0, \omega_1 x_1, \omega_2 x_2, \dots, \omega_p x_p)$$

Da bi potupak klasifikacije bio provediv, uzorke transformiramo tako da ih možemo podijeliti $(P - 1)$ -dimenzijskom hiperravninom. Ravninu odabiremo tako da je maksimalno udaljena od krajnjih uzoraka oba skupa.

Stabla odlučivanja: Binarna stabla, koja mogu biti korištena među ostalim za gradnju klasifikatora, pri čemu je svaki čvor označen nekom labelom i predstavlja određenu klasu.

Neuronske mreže: Najčešće korištene neuronske mreže su višeslojni perceptroni (MLP), koji se sastoje od ulaznog sloja, izlaznog sloja, i jednog ili više skrivenih slojeva. Svaki sloj se sastoji od jednog ili više neurona koji su povezani s neuronima iz prethodnog i sljedećeg sloja. Ulazni sloj prima određene značajke, te sustav na temelju izračunatih težina određuje hoće li objekt biti prihvaćen ili ne.

Boostanje: Boostanje je algoritam strojnog učenja, koji iz skupa slabih klasifikatora gradi jedan jaki klasifikator kao linearnu kombinaciju $f(x) = \sum_{t=1}^T \alpha_t h_t(x)$ slabih klasifikatora $h_t(x)$.

3. Opis metode Viole i Jonesa

Metoda koju su predložili Paul Viola i Michael Jones je jedna od metoda pronalaženja objekata koja koristi strojno učenje, te ujedno i prva metoda koja postiže dobre rezultate u stvarnom vremenu [7]. Primarna svrha joj je detekcija lica, no može biti naučena za detekciju bilo koje vrste objekata, budući da koristi algoritme strojnog učenja. Metoda vrlo brzo procesira slike i postiže visoki broj detekcija, te je brzo nakon objavljivanja postavila standarde u području pronalaženja lica. U usporedbi s ranijim metodama, metoda Viole i Jonesa je učinkovitija najviše zbog brzine detekcije (na računalu 700 MHz Intel Pentium 3 postignuta je detekcija na 15 slika u sekundi, na slikama veličine 384×288 piksela [7]). Tako veliku brzinu ova metoda postiže među ostalim zbog rada s informacijama koje se nalaze u samom crno–bijelom području slike. Moguće je kombinirati nekoliko različitih metoda, te postići još veću učinkovitost u pronalaženju objekata. Ulazni parametar za ovaj algoritam je slika bilo koje vrste, a izlazni parametar je lista pozicija lica na slici. U nastavku su objašnjena tri glavna dijela od kojih se metoda sastoji.

3.1. Tri glavna doprinosa metode

Integralna slika Novi prikaz slike, koji omogućava detektoru koji koristimo da brzo izračuna značajke slike.

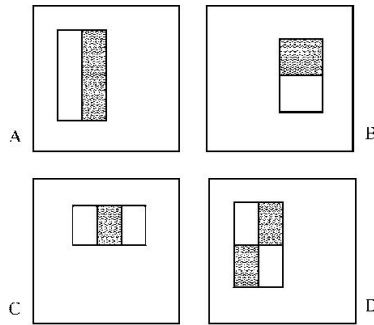
Gradnja klasifikatora Klasifikator je izgrađen korištenjem algoritma *AdaBoost* za strojno učenje koji izdvaja mali broj kritičnih značajki slike od vrlo velikog broja potencijalnih značajki slike.

Metoda kombinacije klasifikatora u kaskadu Omogućava da se brzo odbace suvišni predjeli slike, i detaljnije provede računanje nad predjelima slike koji sadrže mogući objekt koji tražimo, te time poboljšava performanse postupka.

3.2. Haarove značajke

Haarove značajke su pravokutne značajke, i računaju se kao razlika nekoliko suma piksela unutar različitih pravokutnih područja slike.

Tri su tipa značajki definiranih u metodi Viole i Jonesa:



Slika 1: Osnovni skup značajki Violen i Jonesa.

Two-rectangle: Računa razliku suma piksela u 2 pravokutna predjela istog oblika i veličine koji su spojeni jednim vertikalnim ili horizontalnim bridom

Three-rectangle: Uz 3 spojena pravokutnika istog oblika i veličine, računa razliku sume piksela 2 vanjska pravokutnika i sume piksela centralnog pravokutnika

Four-rectangle: Uz 4 spojena pravokutnika istog oblika i veličine, računa razliku suma piksela dijagonalnih parova pravokutnika

Osnovne Haarove značajke su samo *two-rectangle* značajke, a ostale dvije vrste su u svojoj metodi dodali Viola i Jones.

Na detektoru razlučivosti 24×24 piksela, skup Haarovih značajki sadrži oko 160000 značajki, što je očigledno puno više od broja piksela slike. Tako nam Haarove značajke daju brojne informacije o slici, a zbog jednostavnog i učinkovitog načina kojim se one izračunavaju, vrlo su pogodan alat pri pronalaženju objekata.

3.3. Integralna slika

Značajke predstavljene u prethodnom poglavlju možemo izračunati najbrže uvođenjem novog prikaza slike, nazvanog integralna slika. Integralna slika predstavlja transformaciju slike u matricu s jednakim brojem elemenata kao i originalna slika. Na lokaciji (x, y) integralna slika sadrži sumu piksela iznad, i s lijeve strane od x, y , tj.

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

gdje je ii integralna slika, a i originalna slika.

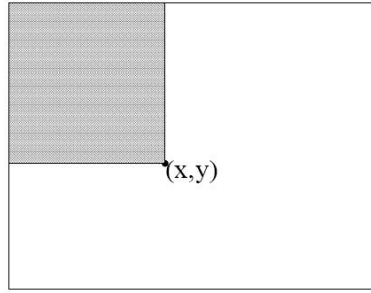
1	1	1
1	1	1
1	1	1

(a) Ulazna slika.

1	2	3
2	4	6
3	6	9

(b) Integralna slika.

Slika 2: Integralna slika.



Slika 3: Reprezentacija integralne slike.

Korištenjem sljedećih dviju jednadžbi:

$$\begin{aligned}s(x, y) &= s(x, y - 1) + i(x, y) \\ ii(x, y) &= ii(x - 1, y) + s(x, y)\end{aligned}$$

gdje s predstavlja ukupnu sumu jednog retka piksela, i gdje vrijedi da je $s(x, -1) = 0$ i $ii(-1, y) = 0$, integralna slika može biti izračunata u jednom prolazu kroz početnu sliku. Korištenjem integralne slike, bilo koja suma piksela pravokutnika može biti izračunata u 4 reference na polje podataka, te iz toga i činjenice da su neki bridovi 2 ili više pravokutnika zajednički, proizlazi da se *two-rectangle* značajke mogu izračunati u 6 referenci, *three-rectangle* značajke u 8 referenci, te *four-rectangle* značajke u 9 referenci.

3.4. Gradnja klasifikatora strojnim učenjem

Svaku od izračunatih značajki možemo predstaviti kao jedan slabi klasifikator. Matematički, značajka kao slabi klasifikator je opisana kao:

$$h_j(x) = \begin{cases} 1 & \text{ako } p_j f_j(x) < p_j \theta_j \\ 0 & \text{inače.} \end{cases}$$

gdje je x podprozor nad kojim radimo, veličine 24×24 piksela, f je korištena značajka, p je polaritet, a θ prag na temelju kojeg odlučujemo treba li x biti klasificiran kao pozitivan (sadrži objekt) ili negativan (ne sadrži objekt).

Kako je provođenje operacija nad cijelim skupom značajki također iznimno sporo (podsjetimo se, broj značajki znatno nadmašuje broj piksela), a eksperimentalno je utvrđeno da vrlo mali broj značajki može formirati klasifikator, potrebno je pronaći taj mali podskup kritičnih značajki. Metoda Viola i Jonesa koristi *boostanje* tj. varijantu *AdaBoost* algoritma strojnog učenja za odabir značajki.

Boostanje je meta-algoritam strojnog učenja, koji iz skupa slabih klasifikatora gradi jedan jaki klasifikator. Naime, slabi klasifikator uspješno klasificira podatke u >50% slučajeva, te je zbog toga potrebno izgraditi jaki klasifikator koji bi postizao veći broj pogodaka.

3.4.1. Algoritam Adaboost

Algoritam *AdaBoost* u originalnoj formi obavlja funkciju *boostanja* performansi jednostavnog algoritma strojnog učenja. To čini kombiniranjem skupa slabih klasifika-

tora tako da tvore jaki klasifikator, tj. gradi jaki klasifikator kao linearnu kombinaciju $f(x) = \sum_{t=1}^T \alpha_t h_t(x)$ slabih klasifikatora $h_t(x)$. Algoritam izgleda ovako :

- Uzmemo uzorke za učenje $(x_1, y_1), \dots, (x_n, y_n)$ pri čemu je $y_1 = 0$ za negativne, odnosno $y_1 = 1$ pozitivne uzorke.
- Inicijaliziramo težine $w_{1,i} = \frac{1}{2m}$ za $y_1 = 0$ i $w_{1,i} = \frac{1}{2l}$ za $y_1 = 1$ pri čemu su m i l broj pozitivnih, odnosno negativnih uzoraka.
- za $t = 1, \dots, T$:

1. Normaliziraj težine $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$
2. Za svaku značajku j treniraj klasifikator h_j , te od svih klasifikatora nađi onaj s najmanjim postotkom grešaka ϵ_t :

$$\epsilon_t = \min_{f,p,\theta} \sum_i w_i |h(x_i, f, p, \theta) - y_i|$$

3. Odredi $h_t(x) = h(x, f_t, p_t, \theta_t)$ pri čemu su f_t , p_t i θ_t minimalne vrijednosti za koje se postiže ϵ_t .
4. Ponovo izračunaj težine:

$$w_{t+1,i} = w_{t,i} \cdot \beta^{1-e_i}$$

pri čemu je $e_i = 0$ ako je uzorak x_i proglašen ispravnim, a $e_i = 1$ u suprotnom; $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$

- Završni jaki klasifikator je:

$$C(x) = \begin{cases} 1 & \text{ako } \sum_{t=1}^T \alpha_t H_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{inače.} \end{cases}$$

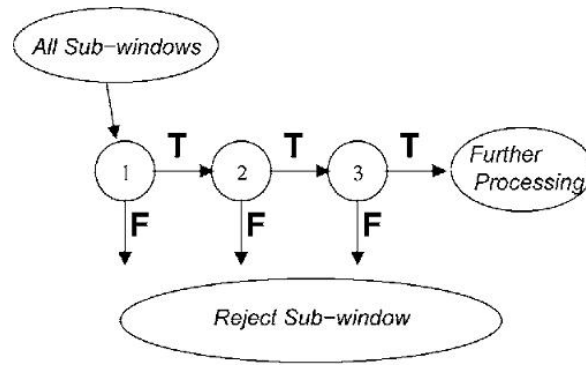
pri čemu je $\alpha_t = \log \frac{1}{\beta_t}$

Iz gornjeg algoritma možemo iščitati kako zapravo *boostamo* slabe klasifikatore. Svaki put naglašavamo one primjere koji su bili neispravno klasificirani, tako da im održavamo konstantne težine, dok primjerima koji su ispravno klasificirani smanjujemo težine. Svaki klasifikator pokušava bolje klasificirati one primjere koje je njegov prethodnik loše klasificirao. Tako tvorimo jaki klasifikator. Broj grešaka takvog klasifikatora se eksponencijalno približava nuli s povećanjem broja primjera za učenje koje smo mu zadali.

U jednom dijelu gornjeg algoritma je potrebno odrediti najbolju značajku, polaritet i prag. Postavlja se pitanje kako to uspješno i brzo riješiti. Viola i Jones su predložili jednostavnu *brute-force* metodu pretraživanja. To znači da za određivanje jednog slabog klasifikatora moramo evaluirati svaku značajku na svakom primjerku za učenje, da bi našli značajku s najboljim performansama. Ovaj postupak je vremenski najzahtjevniji dio cijele metode.

3.5. Ulančavanje boostanih klasifikatora

Nakon što je završena gradnja klasifikatora, performanse se mogu poboljšati kombiniranjem više klasifikatora. Naime, jedan klasifikator dobiven prethodnim algoritmom učenja ne može davati rezultate u realnom vremenu, pa se klasifikatori kombiniraju u *kaskadu* poredano po kompleksnosti, tako da je svaki klasifikator kompleksniji od svog prethodnika. Prvi klasifikator u kaskadi dobija na evaluaciju sve podprozore slike, a svaki sljedeći klasifikator evaluira samo one primjere koje prethodni klasifikator nije odbacio kao negativne (ne sadrže zadani objekt). Ako je neki podprozor u tom procesu odbačen od strane nekog klasifikatora, on se potpuno odbacuje i na njega se više ne primjenjuju nikakve operacije, i pretraživanje kreće do idućeg podprozora. Kaskada tako ima oblik degeneriranog stabla odlučivanja, kao što je prikazano na slici 4.



Slika 4: Kaskada klasifikatora.

Takvim postupkom kaskada povećava brzinu cijelog detektora, jer fokusira pažnju na predjele slike koji “obećavaju”, a kompleksnije procesiranje se čuva samo za te predjele (podprozore). Kada se radi o detekciji lica, prvi klasifikator u kaskadi koristi samo 2 značajke, i postiže postotak greške od 0% kada postupak nalazi objekte koje nismo tražili (false positive rate) i 40% kada postupak ne nalazi tražene objekte (false negative rate). Dakle, prvi klasifikator reducira broj značajki koje prolaze kroz kaskadu za gotovo jednu polovinu. Kako učinak bilo kojeg klasifikatora u kaskadi ovisi o ponašanju njegovog prethodnika, ukupan broj grešaka kada postupak nalazi objekte koji nisu traženi (false positive rate) jest:

$$F = \prod_{i=1}^K f_i$$

gdje je f_i broj grešaka za svaki i -ti klasifikator u kaskadi. Analogno tome, broj ispravnih detekcija jest:

$$D = \prod_{i=1}^K d_i$$

gdje je d_i broj detekcija za svaki i -ti klasifikator u kaskadi. Kako to poboljšava performanse, vidljivo je na primjeru: Ako kaskada koja sadrži 32 klasifikatora postiže postotak greške veličine $F = 10^{-6}$, svaki pojedini klasifikator unutar kaskade može postizati

postotak greške $F = 65\%$. Međutim, svaki klasifikator mora imati vrlo visok postotak ispravnih detekcija, npr. $D = 99.7\%$ da bi cijela kaskada imala postotak detekcija veličine $D = 90\%$.

4. Primjena metode Viole i Jonesa

Budući da je metoda brza i postiže dobre rezultate u realnom vremenu, moguće su mnoge primjene uz eventualne male izmjene i poboljšanja osnovnog algoritma, koje zahtijevaju detekciju lica, kao npr. za detekciju profila lica, nekih karakteristika lica (oči, nos, oblici lica itd), u demografskim analizama, u telekonferencijama, biometrici (kombinirano detekcija lica i prepoznavanje različitih lica), za nadzorne video-kamere, za razvoj korisničkog sučelja između čovjeka i računala, u slikovnim bazama podataka itd. Neke od novijih digitalnih kamera također koriste detekciju lica za autofokusiranje.

Također, iako je pronalaženje lica osnovna namjena ove metode, može se koristiti i za pronalaženje objekata bilo koje druge klase (pronalaženje prometnih znakova, pronalaženje pješaka na cesti i sl.). U aplikacijama koje ne zahtijevaju veliku brzinu, metoda dozvoljava značajna dodatna post-procesiranja i analize, te kombinaciju s drugim metodama detekcije objekata. Takvi sustavi mogu biti implementirani na širokom spektru ugrađenih računala, uređaja male snage (eng. *low power devices*), u mobilnim, hand-heldovima itd. Recimo, uspješno je implementiran detektor lica na StrongARM procesoru od 200 MIPS, i postignuto je vrijeme obrade od 2 slike po sekundi. [7]

5. Evaluiranje Lienhartovog izvornog koda

Implementacija metode koju je predložio Rainer Lienhart se razlikuje od prvotne zamisli Viole i Jonesa u dvije stvari:

1. Osim osnovnih Haarovih značajki, korištene su i one rotirane za 45° , pa je time dodano malo poboljšanje u odnosu na osnovnu metodu
2. Klasifikatori u kaskadi su predstavljeni cijelim stablima, a ne samo jednim čvorom (slika 5).

Lienhartova implementacija se može pronaći u biblioteci *OpenCV*.

5.1. Biblioteka *OpenCV*

Open Computer Vision library je *open source* biblioteka pod *BSD* licencom¹ koja sadrži preko 500 algoritama iz područja računalnog vida.

Algoritmi rade na obradi slika u stvarnom vremenu. Dostupna je na Linux, Windows i MacOS X operacijskim sustavima, a pisana u programskim jezicima C i C++ te postoje programska sučelja prema *Pythonu*, *Rubyu*, *Matlabu*, i drugim jezicima.

¹URL: <http://www.opensource.org/licenses/bsd-license.php>

Sadrži module za rad sa matričnim strukturama podataka, rad sa video sadržajem iz datoteka ili kamera, procesiranje i transformaciju slika, izgradnju histograma i kon-tura, alate za praćenje pokretnih objekata, kalibriranje kamera, izgradnju projekcije i 3D pogleda, algoritme za strojno učenje, te modul za izgradnju grafičkog sučelja [1].

Službeni *OpenCV* resursi na Internetu su:

- <http://sourceforge.net/projects/opencvlibrary/>
Izvorni kod biblioteke Open Computer Vision.
- <http://tech.groups.yahoo.com/group/OpenCV/>
Mailing lista i grupa zajednice *OpenCV*.
- <http://opencv.willowgarage.com/wiki/>
Wiki stranice zajednice *OpenCV*.

5.2. Korištenje biblioteke *OpenCV* za pronalaženje objekata

Biblioteka *OpenCV* sadrži gotove module za pronalaženje objekata preko metode Vi-ole i Jonesa s treniranim klasifikatorima za pronalaženje ljudskog lica. Da bi izgradili klasifikator za pronalaženje drugog tipa objekta trebamo:

1. **Sastaviti bazu uzoraka.** Kolekcija slika na kojima se pojavljuje zadani objekt (pozitivni primjer) i kolekcija slika na kojima se taj objekt ne pojavljuje (negativni primjer, tzv. “pozadina” (eng.: *background*)). Kolekcije moraju biti “*velike*” (npr. iznad 5000 slika za negativne primjere i npr. 1000 slika za pozitivne primjere).
2. **Podijeliti uzorke.** Slike na kojima se objekt pojavljuje dijelimo u dvije skupine:
 - (a) Pozitivne uzorke za treniranje (eng. : *positives*)
 - (b) Pozitivne test uzorke za ispitivanje

Za treniranje uzimamo 90 – 95% svih pozitivnih uzoraka, a za testiranje ostatak pozitivnih uzoraka. Pozitivni uzorci za treniranje i za testiranje moraju biti dis-junktni. Istu podjelu radimo i za negativne uzorke.

3. **Označiti objekte na svim pozitivnim uzorcima.** Označavanje vršimo bilježe-njem koordinate gornjeg lijevog kuta objekta, te njegove visine i širine.
4. **Skalirati i isjeći kolekciju pozitivnih uzoraka za učenje.** Iskoristimo *info* da-toteku i alat *createsamples* koji dolazi sa bibliotekom *OpenCV*. Poziv može biti:

```
createsamples -info positives/imgs/positives.dat \  
-vec pos_samples.vec
```

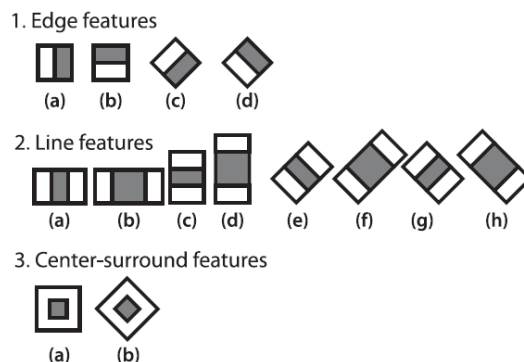
Sam opis parametara može se dobiti naredbom *createsamples* bez argumenata. Bitno je napomenuti da se u *positives.dat* datoteci nalaze *relativne* putanje do slika u odnosu na *positives.dat* datoteku, a ne *createsamples* program. Time dobivamo *pos_samples.vec* paket.

5. **Izgraditi (naučiti) klasifikator.** Gradimo ga pomoću *pos_samples.vec* paketa, kolekcije negativnih uzoraka i haartraining programa koji također dolazi sa bibliotekom *OpenCV*. Poziv može biti:

```
haartraining -data positives/classifier \
-vec positives/data/pos_samples.vec \
-bg negatives/imgs/negatives.dat -npos 330 \
-nneg 97 -mem 800
```

Za detalje pogledajte službenu dokumentaciju [1] ili pokrenite haartraining program bez parametara. Treniranje se odvija kroz nekoliko prolaza, a cilj nam je izgraditi klasifikator čija je vjerojatnost greške ispod $5 \cdot 10^{-6}$ [4].

6. **Provjeriti rezultate.** Za potrebe testiranja izgrađenog klasifikatora koristimo program *performance* koji prima izgrađeni klasifikator i opisnik skupa test podataka. Kao rezultat dobivamo danu test kolekciju s označenim područjem objekta iz podataka dobivenih obradom.



Slika 5: Lienhartovo proširenje haarovih značajki u *OpenCV* biblioteci.

Za računalno najzahtjevniji dio je treniranje klasifikatora, a za čovjeka označavanje objekata na pozitivnim uzorcima.

Za potrebe označavanja razvijeni su alati *Object Marker*² i *MASTIF Marker*³. Oni omogućavaju vizualni odabir koordinata te visine i širine te. Rezultati se spremaju u “*info*” datoteku, pri čemu je format *Object Marker* alata spreman za korištenje sa *createsamples* programom, dok *MASTIF Marker* sadrži dodatne informacije o tipu objekta koji se nalazi na slici. Opis formata *MASTIF Markera* možete naći u [2], a *Object Markera* u [4].

6. Zaključak

U ovom radu je dan prikaz primjene strojnog učenja na pronalaženje objekata u računalnom vidu, uz detaljno objašnjenje algoritma i implementacije jedne od najučinkovitijih

²URL: <http://www.cse.unsw.edu.au/~gherman/ObjectMarker.zip> (Improved Object Marker)

³URL: <http://www.zemris.fer.hr/~ssegvic/mastif/marker/marker.zip>

i najbržih metoda, koju su predložili Paul Viola i Michael Jones. Ovaj postupak je postavio standarde u računalnom vidu, budući da je oko 15 puta brži nego bilo koji prijašnji postupak detekcije objekata.

Primjene su brojne, kako danas, tako i u budućnosti. Budući da je postupak relativno nov, kao i cijelo područje računalnog vida, razumno je u skoroj budućnosti očekivati dodatna poboljšanja metode, nove metode temeljene na opisanoj metodi, te brojne primjene u tehnologiji.

Literatura

- [1] Gary Bradski i Adrian Kaehler. *Learning OpenCV*. O'Reilly Media, Sebastopol, 1. izdanje, 2008.
- [2] Fakultet Elektrotehnike i Računarstva. *MASTIF Marker – upute za korištenje*, 1. izdanje, 2009.
- [3] Y. Freund, R. Schapire, i N. Abe. A short introduction to boosting. *Japanese Society For Artificial Intelligence*, 14:771–780, 1999.
- [4] Inflomatik. *How-to build a cascade of boosted classifiers based on Haar-like features*, 1. izdanje, 2003.
- [5] Ole Helvig Jensen. Implementing the viola-jones face detection algorithm. Tehnički izvještaj, Technical University of Denmark, Informatics and Mathematical Modelling, 2008.
- [6] P. Viola i M. Jones. Rapid Object Detection using a Boosted Cascade of Simple. U *Proceedings of CVPR2001*, svezak 1.
- [7] Paul Viola i Michael Jones. Robust real-time object detection. U *International Journal of Computer Vision*, 2001.