SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINAR

Detekcija objekata u slikama upotrebom stabala odlučivanja

Nenad Markuš
Voditelj: *Prof. dr. sc. Nikola Bogunović*

Zagreb, siječanj 2015.

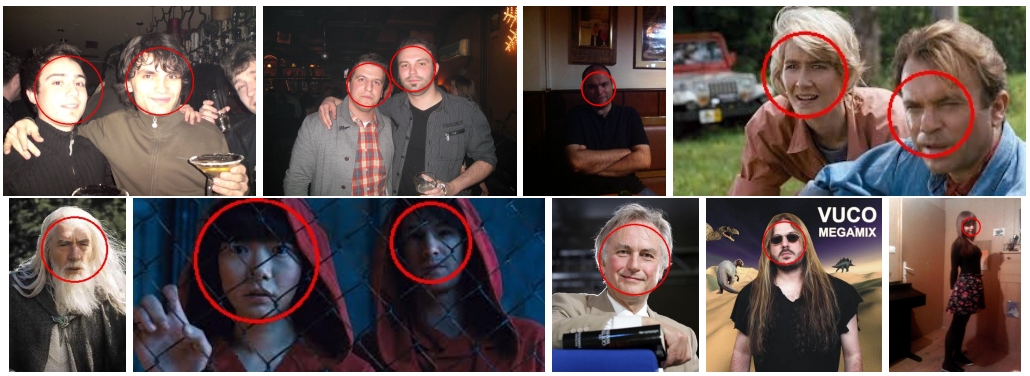
SADRŽAJ

1. Uvod

Jedna od temeljnih metoda računalnog prikaza znanja i zaključivanja su stabla odlučivanja Breiman et al. (1984). Ovaj će seminarski rad dati kratak uvod u taj značajan statistički alat kroz primjer njegove primjene u detekciji objekata u digitalnim slikama.

U području računalnog vida, detekcija jest zadaća pronalaženja pozicija i veličina svih objekata koji se nalaze u zadanoj digitalnoj slici, a pripadaju određenom razredu/klasi. Taj razred mogu biti automobili, pješaci, ljudska lica ili nešto drugo. Jasno je da matode automatske detekcije objekata nalaze veliku primjenu. Neki primjeri uključuju biometrijske aplikacije, nadzor objekata i prostora, izrada pametnih korisničkih sučelja, itd. Dakle, jasno je zašto se isplati istraživati brze i precizne algoritme za rješavanje ovih problema.

Drugo poglavlje daje kratak uvod u detalje upotrebljavanih algoritama. Treće poglavlje prikazuje rad razvijenih programa kroz primjer detekcije ljudskih lica. Četvrto poglavlje sadrži neke rezultate koji nisu detaljno opisani u ovom seminarskom radu. Opširniji tehnički izvještaj nalazi se na `http://arxiv.org/abs/1305.4537`, a ostali materijali (uključujući i programski kod) na `http://public.tel.fer.hr/odet/`. Neka lica detektirana razvijenim sustavom prikazana su na slici 1.1.



Slika 1.1: Lica detektirana programom opisanim u ovom seminarskom radu.

2. Algoritmi i metode

Korišteni pristup modifikacija je standardne metode detekcije objekata opisane u Viola i Jones (2001). Osnovna je ideja propustiti sve regije slike kroz kaskadu binarnih klasifikatora. Pritom, regija se klasificira kao lice ako prođe *sve* članove kaskade. Svaki klasifikator u kaskadi sastoji se od stabala odlučivanja. Razlika našeg algoritma u odnosu na Viola i Jones (2001) je u značajkama koje se izlučuju iz slike i upotrebljavaju u unutrašnjim čvorovim stabala.

2.1. Stabla odlučivanja

Stabla odlučivanja statistički su alat za aproksimaciju funkcija. Osnovna je ideja rekurzivno razdijeliti problem u dva potproblema, svaki od njih riješiv metodom smanjene složenosti. Dioba problema izvršava se u unutarnjim čvorovima stabla na temelju binarnih testova karakterističnih za podatke kojima baratamo. Listovi stabla sadrže jednostavne modele koji aproksimiraju traženu funkciju. U praksi, binarni testovi u unutarnjim čvorovima i modeli u listovima odabiru se na temelju konačnog broja uparenih ulaza i izlaza. U literaturi je predloženo mnogo različitih oblika stabala odlučivanja i ovdje ne možemo pokriti sve detalje (pogledati Hastie et al. (2009)). Stoga, u nastavku je opis onih koja su pogodna za naš problem.

Binarni test u unutarnjim čvorovima svakog stabla definiran je kao:

$$\text{bintest}(I; \mathbf{l}_1, \mathbf{l}_2) = \begin{cases} 0, & I(\mathbf{l}_1) \leq I(\mathbf{l}_2) \\ 1, & \text{inače.} \end{cases} \tag{2.1}$$

Ulaz I predstavlja zadanu regiju slike koju želimo klasificirati, a $\mathbf{l}_1$ i $\mathbf{l}_2$ su koordinate na kojima trebamo uzorkovati pojedine piksele unutar te regije. Numeričke vrijednosti dobivene na taj način označene su s $I(\mathbf{l}_1)$ i $I(\mathbf{l}_2)$. Koordinate $\mathbf{l}_1$ i $\mathbf{l}_2$ su normalizirane, tj. obje su iz skupa $[-1, +1] \times [-1, +1]$. To nam omogućuje jednostavnu primjenu testa na regiju proizvoljne veličine. Listovi svakog stabla sadrže jedan realni broj koji modelira traženi izlaz.

Izgradnja stabla zasniva se na skupu za učenje: $\{(I_s, v_s, w_s) : s = 1, 2, \ldots, S\}$. Realni broj v_s predstavlja željeni izlaz za sliku I_s, a w_s je težina pridružena s-tom uzorku. Npr., kod binarne klasifikacije, v_s je iz skupa $\{-1, +1\}$. Težine w_s omogućuju da svakom uzorku pridružujemo drugačiju razinu važnosti (ovo će igrati ulogu kasnijem odjeljku). Binarni testovi u unutarnjim čvorovima stabla odabiru se tako da minimiziraju sljedeću kriterijsku funkciju:

$$\text{WMSE} = \sum_{(I,v,w)\in C_0} w \cdot (v - \bar{v}_0)^2 + \sum_{(I,v,w)\in C_1} w \cdot (v - \bar{v}_1)^2. \tag{2.2}$$

Skupovi C_0 i C_1 odgovaraju uzorcima u skupu za učenje za koje je izlaz 0, odnosno 1. Skalari $\bar{v}_0$ i $\bar{v}_1$ dobiveni su kao

$$\bar{v}_i = \frac{1}{\sum_{(I,v,w)\in C_i} w} \sum_{(I,v,w)\in C_i} w \cdot v \tag{2.3}$$

za $i \in \{0, 1\}$. Budući da je ukupni broj mogućih binarnih testova iznimno velik, pri izgradnji svakog čvora generira se 1024 njih i odabire se onaj koji daje najmanju vrijednost kriterijske funkcije 2.2. Uzorci za učenje rekurzivno se grupiraju na ovaj način dok ne dostignemo određenu dubinu stabla. Tako ograničavamo prostornu i vremensku složenost procesa učenja i primjene stabala. U svakom je listu pohranjena težinska srednja vrijednost svih uzoraka koji su dospjeli do njega:

$$\bar{v} = \frac{1}{\sum_{(I,v,w)\in C} w} \sum_{(I,v,w)\in C} w \cdot v. \tag{2.4}$$

Stablo dubine d treba $O(2^d)$ okteta za pohranu, a vrijeme izvođenja za zadanu regiju slike proporcionalno je s d.

2.2. Algoritam GentleBoost

Poznati je problem strojnog učenja da će svako pojedinačno stablo najčešće loše raditi Hastie et al. (2009). Ipak, zbrajanjem izlaza nekoliko stabala dobivaju se značajno bolji rezultati. U ovom ćemo radu koristiti algoritam GentleBoost Friedman et al. (1998) (inačica bolje poznatog AdaBoosta) kako bismo generirali precizne klasifikatore.

Osnovna je ideja iterativno dodavati stabla rješavajući niz problema najmanjih kvadrata. Za dani skup za učenje $\{(I_s, c_s) : s = 1, 2, \ldots, S\}$ i broj stabala K, algoritam možemo sažeti sljedećim koracima:

1. Inicijaliziraj težine w_s za svaku sliku I_s i pripadnu labelu $c_s \in \{-1, +1\}$ kao

$$w_s = \begin{cases} 1/P, & c_s = +1 \\ 1/N, & c_s = -1 \end{cases}$$

gdje je P broj "pozitivnih" primjera ($c_s = +1$, slike traženih objekata), a N broj "negativnih" primjera ($c_s = -1$, pozadina).

2. Za $k = 1, 2, \ldots, K$:

   (a) Nauči stablo T_k na skupu $\{(I_s, c_s, w_s) : s = 1, 2, \ldots, S\}$, kao što je opisano u prijašnjem poglavlju.

   (b) Osvježi sve težine:

$$w_s = w_s \exp\left(-c_s T_k(I_s)\right),$$

   gdje je $T_k(I_s)$ označava izlaz (realni broj u listu) stabla T_k za sliku I_s.

   (c) Normaliziraj sve težine tako da im je ukupna suma jednaka 1, tj.

$$\sum_{s=1}^{S} w_s = 1.$$

3. Rezultat učenja je skup stabala: $\{T_k : \ k = 1, 2, \ldots, K\}$.

Prilikom primjene dobivene šume na nekoj regiji slike, izlazi svih stabala se zbrajaju i dobiveni realni broj uspoređuje se s pragom τ:

$$\text{izlaz za sliku } I = \begin{cases} +1, & \sum_{k=1}^{K} T_k(I) > \tau \\ -1, & \text{inače.} \end{cases}$$

Za različite vrijednosti praga τ dobivamo drugačije omjere točno i lažno pozitivnih (engl. *true/false positive rate* — TPR i FPR). Ovo igra važnu ulogu u izgradnji učinkovitih detektora, kao što je opisano u nastavku.

2.3. Detekcija objekata kaskadom klasifikatora

Bez nekakvog *a priori* znanja, potrebno je klasificirati sve regije slike kako bismo pronašli tražene objekte. Npr., takvih regija većih od 100×100 piksela ima stotine tisuća u slici veličine 640×480. Budući da je to vrlo veliki broj, slijedimo prijedlog iz Viola i Jones (2001). Ideja je imati više stadija klasifikacije različite složenosti. Raniji stadiji sastoje se od manje stabala odlučivanja i time su brži. Svaki stadij ima postavljen

prag τ tako da TPR bude visok (≈ 0.999). Pritom se uvijek odbacuje određeni postotak pozadine. Ukupno vrijedi (umnožak ide po stadijima kaskade):

$$\text{TPR} = \prod_i \text{TPR}_i$$

$$\text{FPR} = \prod_i \text{FPR}_i$$

U praktičnim situacijama parametri kaskade mogu se postaviti tako da je TPR ≥ 0.98, a FPR $\approx 10^{-6}$. Opisanim se postupkom većina pozadine odbacuje s vrlo malo proračuna i moguće je postići detekciju objekata u stvarnom vremenu Viola i Jones (2001).

2.4. Grupiranje detekcija

Budući da je kaskada robusna na male perturbacije u položaju i veličini traženih objekata, očekujemo višestruke odzive u okolini svakog objekta. Takve se detekcije grupiraju zajedno usrednjavanjem njihovih parametara (pozicija i veličina) ako je njihovo preklapanje veće od 50% (*intersection-over-union* kriterij):

$$\frac{R_1 \cap R_2}{R_1 \cup R_2} > 0.5.$$

U gornjem su izrazu R_1 i R_2 regije (pravokutnici) u slici koje odgovaraju detekcijama.

3. Detekcija lica

Ovo poglavlje daje eksperimentalne rezultate dobivene primjenom opisane metode na detekciju frontalnih ljudskih lica.

3.1. Učenje

Koristimo dvije baze slika za generiranje pozitivnih primjera:

- AFLW Koestinger et al. (2011)
- VT (`http://www.visagetechnologies.com`)

Obje se sastoje od nekoliko desetaka tisuća slika lica. Za svaku sliku postoji i pripadna anotacija koja, između ostaloga, sadrži i koordinate očiju. Te se koordinate upotrebljavaju za procjenu pozicije i veličine svakog lica. Za svako od $20\ 000$ frontalnih lica generiramo 15 primjera za učenje tako da na slučajan način lagano perturbiramo pravokutnik oko lica, tj. njegovu procijenju poziciju i veličinu. Preliminarni su rezultati pokazali da se na ovaj način postiže veća otpornost na *aliasing* i šum. Opisani proces vodi do $300\ 000$ pozitivnih primjera za učenje. Negativni primjeri uzorkuju se iz vrlo velikog skupa slika koje ne sadrže lica ("pozadina", engl. *background*). Svaki stadij učen je na $300\ 000$ regija pozadine koje nisu odbačene niti jednim prijašnjim stadijem. Ovim se postupkom razmatra *nekoliko stotina milijardi* regija u slikama pozadine.

U ovom eksperimentu, parametri procesa učenja empirijski su podešeni tako da vode do brzog detektora:

- Dubina svakog stabla ograničena je na 6.
- Broj stadija u kaskadi postavljen je na 20.
- Svaki stadij ima zadan broj stabala i TPR.

Neki numerički rezultati procesa učenja mogu se vidjeti u tablici 3.1. Za cijelu kaskadu vrijedi: TPR ≈ 0.92, FPR $\approx 10^{-7}$. Treba imati na umu da naizgled nizak TPR ne vodi na loše rezultate u praksi budući da je za svako lice iz baze generirano 15 primjera.

broj stabala	1	2	3	4	5	10	20	20	...	20	20
TPR [%]	97.5	98.0	98.5	99.0	99.5	99.7	99.9	99.9	...	99.9	99.9
FPR [%]	46.4	32.3	20.5	35.4	44.7	36.8	29.5	31.6	...	55.2	57.5

Tablica 3.1: Broj stabala i preciznost (TPR/FPR) nekih od stadija.

Proces učenja traje 30-ak sati na modernom računalu s četiri jezgre i 16GB RAM-a. Naučena kaskada zauzima otprilike 200kB prostora.

3.2. Analiza točnosti i brzine izvođenja

Razvijeni sustav uspoređujemo s vrlo popularnim detektorima javno dostupnim u knjžnici OpenCV (`http://opencv.org`, inačica 2.4.3). Prvi od njih temelji se na klasičnom algoritmu iz Viola i Jones (2001). Implementacija je opisana u Lienhart i Maydt (2002) i Lienhart et al. (2003). Drugi detektor temeljen je na LBP-ovima (engl. *local binary patterns*), kao što je opisano u Zhang et al. (2007).

Broj točno detektiranih lica mjerimo na dvije baze:

- GENKI-SZSL `http://mplab.ucsd.edu` (3500 lica)
- CALTECH-FACES Angelova et al. (2005) (10 000 lica)

Broj lažno pozitivnih odziva mjerimo na dva vrlo velika skupa slika koji ne sadrže lica: NO-FACES-1 i NO-FACES-2. Naravno, niti jedan od detektora nije učen na slikama koje će biti u prikazanim eksperimentima.

Zadaća je pronaći sva lica u zadanom skupu slika koja su veća od 24×24 piksela. Točnost detektora prikazujemo ROC krivuljama (engl. *receiver operating characteristic curves*). Postignuti rezultati vide se na slikama 3.1 i 3.2. Uočavamo da naš detektor postiže veću točnost u provedenim eksperimentima.

Osim točnosti, drugi važan parametar detektora objekata je i njegova brzina. Zanimaju nas realistične situacije: zadaća je pronaći sva lica veća od 100×100 piksela u slici veličine 640×480. Ostale postavke detektora su kao i u ranija dva eksperimenta. U tablici 3.2 sažete su brzine izvođenja za razne uređaje. Vidimo da naš detektor pronalazi lica *značajno brže* od konkurencije iz eksperimenata, pogotovo na mobilnim uređajima. Treba imati na umu da je OpenCV optimiziran za PC-jeve: SIMD instrukcije, višedretvenost i strukture podataka optimizirane za priručnu memoriju (engl. *cache*). Loše performanse na mobilnim uređajima možemo barem djelomično pripi-

Device	CPU	Vrijeme [ms]		
		Naš detektor	V-J (OpenCV)	LBPs (OpenCV)
PC1	3.4GHz Core i7-2600	2.4	16.9	4.9
PC2	2.53GHz Core 2 Duo P8700	2.8	25.4	6.3
iPhone 5	1.3GHz Apple A6	6.3	175.3	47.3
iPad 2	1GHz ARM Cortex-A9	12.1	347.6	103.5
iPhone 4S	800MHz ARM Cortex-A9	14.7	430.3	129.2

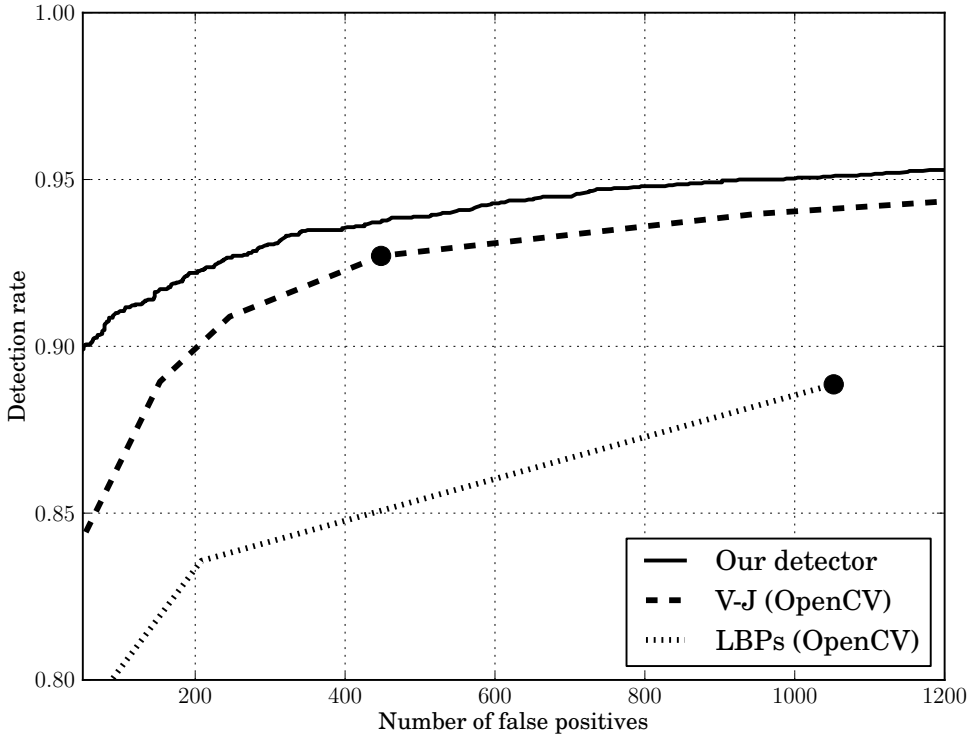
Tablica 3.2: Prosječna vremena potrebna za pronalaženje lica u slikama.

sati upotrebi *floating point* aritmetike[1]. Naša je implementacija napisana u C-u i sav se izračun vrši u jednoj dretvi, većinom pomoću *integer* aritmetike (ovo je "prirodna" implmentacija imajući na umu jednostavnost binarnih testova 2.1 u unutarnjim čvorovima stabala).

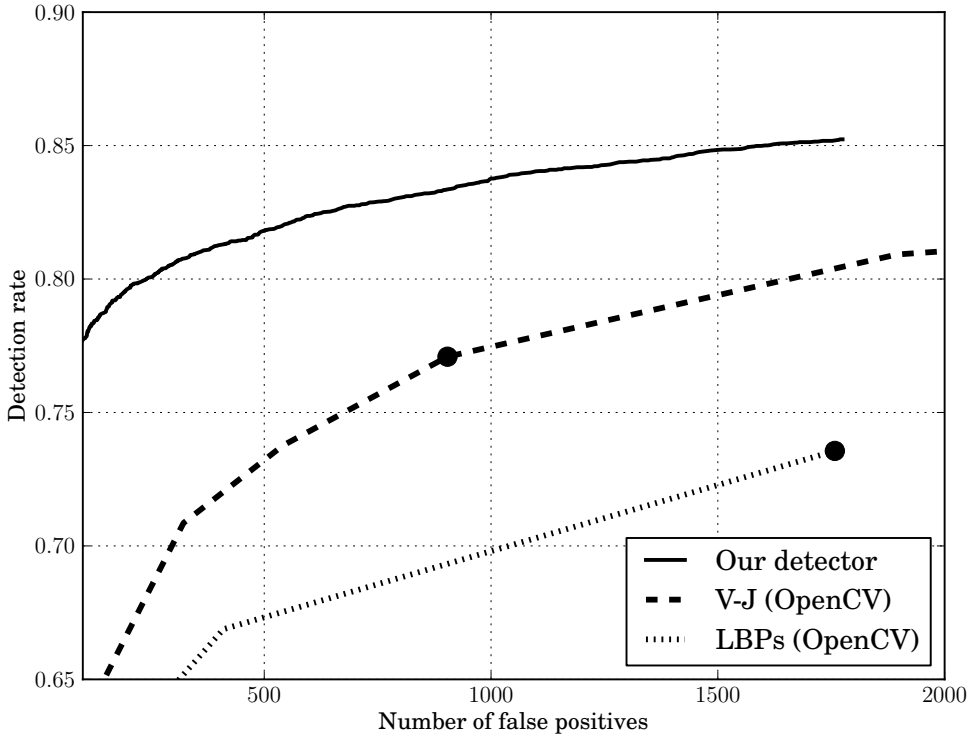
3.3. Zaključak

Pokazali smo da je moguće izgraditi učinkovit i precizan detektor lica pomoću vrlo jednostavnih binarnih testova ako ih složimo u stabla odlučivanja. Prednost nad konkurencijom iz široko upotrebljavane knjižnice OpenCV pogotovo je vidljiva na uređajima ograničenih računalnih resursa.

[1]Taj su problem uočili i drugdje: `http://www.computer-vision-software.com/blog/2009/04/fixing-opencv/`



Slika 3.1: Rezultati na GENKI/NO-FACES-1 eksperimentu.



Slika 3.2: Rezultati na CALTECH/NO-FACES-2 eksperimentu.

4. Ostali rezultati

Algoritam je primijenjen na detekciju različitih objekata, kao što je prikazano na slikama 4.1, 4.2 i 4.3.

Slika 4.1: Detekcija lica.

Slika 4.2: Detekcija automobilskih svjetala.

Slika 4.3: Detekcija ruku.

5. Literatura

A. Angelova, Y. Abu-Mostafa, i P. Perona. Pruning training sets for learning of object categories. U *CVPR*, 2005.

L. Breiman, J. Friedman, C.J. Stone, i R.A. Olshen. *Classification and Regression Trees*. Chapman and Hall, 1984.

J. Friedman, T. Hastie, i R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28, 1998.

T. Hastie, R. Tibshirani, i J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

M. Koestinger, P. Wohlhart, P. M. Roth, i H. Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. U *First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, 2011.

R. Lienhart i J. Maydt. An extended set of haar-like features for rapid object detection. U *IEEE ICIP*, svezak 1, stranice 900–903, 2002.

R. Lienhart, A. Kuranov, i V. Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. U *Proceedings of the 25th DAGM-Symposium*, stranice 297–304, 2003.

`http://mplab.ucsd.edu`. The mplab genki database, genki-szsl subset.

P. Viola i M. Jones. Rapid object detection using a boosted cascade of simple features. U *CVPR*, 2001.

L. Zhang, R. Chu, S. Xiang, S. Liao, i S. Z. Li. Face detection based on multi-block lbp representation. *Advances in Biometrics*, 4642:11–18, 2007.