

SVEUČILIŠTE U ZAGREBU

**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

**SEMINAR**

**Kategorizacija objekata zbirakama značajki**

*Alan Sambol*

Voditelj: *Doc. dr. sc. Siniša Šegvić*

Zagreb, travanj, 2011.

## Sadržaj

1. Uvod.....	1
2. Detekcija i izlučivanje zbirke značajki.....	2
2.1 Model zbirke značajki u procesiranju prirodnog jezika.....	2
2.2 Model zbirke značajki u računalnom vidu.....	3
2.2.1 Detekcija značajki.....	5
2.2.2 Formiranje opisnika značajki.....	7
2.2.3 Formiranje rječnika.....	8
3. Učenje i kategorizacija korištenjem modela zbirka značajki.....	10
3.1 Generativni modeli.....	10
3.1.1 Naivni Bayesov klasifikator.....	11
3.1.2 Hijerarhijski Bayesovi modeli.....	12
3.2 Diskriminativni modeli.....	13
4. Demonstracija funkcionalnosti.....	14
5. Zaključak.....	17
6. Literatura.....	18
Sažetak .....	20

## 1. Uvod

U područjima računalnog vida i strojnog učenja koriste se razni načini ekstrakcije bitnih dijelova ulaznih podataka, odnosno slika. Objekti se u računalnom vidu kao i u stvarnom životu jednoznačno mogu odrediti opisujući određene karakteristike (npr. geometrijski oblik, veličina, tekstura površine itd.), odnosno značajke.

Model zbirke značajki (engl. *Bag of words model*, *Bag of features model* [3]) vuče korijene iz područja procesiranja prirodnog jezika, gdje su dokumenti reprezentirani riječima, međutim, redoslijed tih riječi je nebitan. Sličan pristup primjenjuje se na slike – one se se reprezentiraju kao skupine regija ili objekata (u daljnjem tekstu: riječi), pri čemu se zanemaruje prostorni odnos tih objekata.

U ovom seminarskom radu opisan je postupak korištenja modela zbirke značajki pri kategorizaciji objekata. U prvom poglavlju opisan je postupak izlučivanja značajki i formiranja opisnika značajki. Nakon toga opisani su neki generativni i diskriminativni modeli koje možemo koristiti za klasifikaciju te je na kraju dana kratka demonstracija funkcionalnosti.

## 2. Detekcija i izlučivanje zbirke značajki

Prije nego što pojasnimo primjenu modela zbirka značajki u računalnom vidu, kratko ćemo se osvrnuti na model zbirke značajki u procesiranju prirodnog jezika, od kuda taj pristup i potječe.

### 2.1. Model zbirke značajki u procesiranju prirodnog jezika

Model zbirke značajki se koristi kako bi se pojednostavili problemi procesiranja prirodnog jezika (engl. *natural language processing, NLP*) i traženja informacija (engl. *information retrieval*) [4]. Tekst se reprezentira kao kolekcija riječi čiji je redosljed nebitan, čime se zanemaruje i gramatika.

Pogledajmo, za primjer, dva jednostavna tekstualna dokumenta:

- Marko voli operacijski sustav Windows. Pero ga također voli.
- Marko također voli planinariti.

Od ta dva dokumenta konstruiramo rječnik od 9 riječi:

- rječnik = {1: Marko, 2: voli, 3: operacijski, 4: sustav, 5: Windows, 6: Pero, 7: ga, 8: također, 9: planinariti}

Sada prijašnja dva dokumenta možemo prikazati vektorima s 9 članova, koji zapravo predstavljaju histograme pojavljivanja pojedinih riječi:

- [1 2 1 1 1 1 1 1 0]
- [1 1 0 0 0 0 0 1 1]

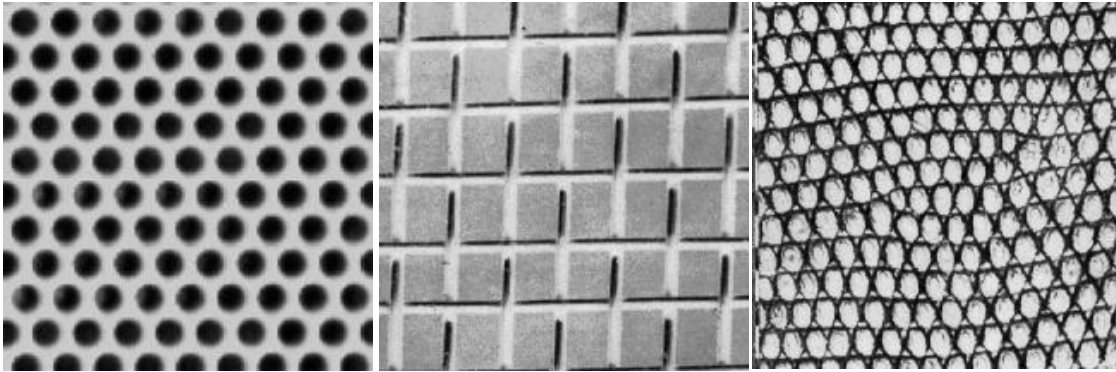
Kao što vidimo, u prikazima histograma smo izgubili informaciju o redosljedu riječi, međutim, ti histogrami se uspješno koriste u nekoliko statističkih metoda, npr. naivni

Bayesov klasifikator, metoda LDA (engl. *latent Dirichlet allocation* [5]) i metoda pLSA (engl. *probabilistic latent semantic analysis* [6]).

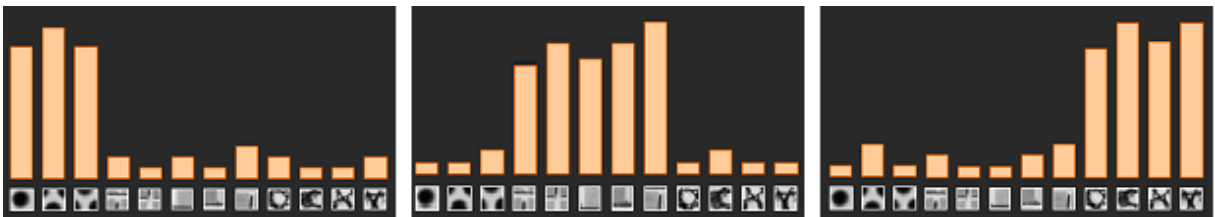
Česta primjena ovog modela je u Bayesovom filtru neželjene pošte (engl. *Bayesian spam filter* [7]), gdje se svaka e-mail poruka modelira kao kolekcija riječi iz jedne od dvije vjerojatnosne distribucije – jedna koja predstavlja neželjenu poštu (engl. *spam*) i druga koja predstavlja legitimnu poštu (engl. *ham*). Jedna "vreća" riječi sadrži riječi koje se nalaze u neželjenoj pošti, dok druga sadrži riječi koje se nalaze u legitimnoj pošti. Iako se bilo koja proizvoljna riječ vrlo vjerojatno nalazi u obje "vreće", "vreća" s riječima neželjene pošte sadrži puno više riječi povezanih s neželjenom poštom, kao što su "stock" i "buy", dok druga "vreća" sadrži više riječi povezanih uz kontakte korisnika ili uz njegovu radnu površinu. Kako bi klasificirao novu poruku, Bayesov filter pretpostavlja da je ona nakupina riječi nasumično odabrana iz neke "vreće", te koristi Bayesovu formulu da odredi kojoj "vreći" poruka vjerojatnije pripada.

## **2.2. Model zbirki značajki u računalnom vidu**

U području računalnog vida, model zbirki značajki se isprva koristio za rješavanje problema klasifikacije tekstura, čija je glavna karakteristika ponavljanje osnovnih elemenata (engl. *texton*). Kod tekstura, bitan je identitet osnovnih elemenata teksture, a ne njihov prostorni razmještaj, što je pogodno za primjenu modela zbirki značajki. Slika 2.1 prikazuje tri različite teksture, dok Slika 2.2 prikazuje histograme njihovih riječi [1].



Slika 2.1



Slika 2.2

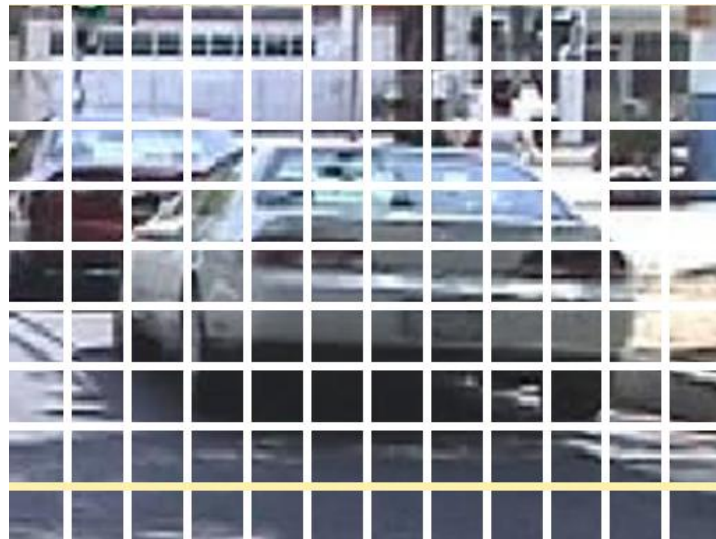
Ulaznu sliku možemo smatrati dokumentom te moramo definirati značenje pojma "riječ" u slici. Problem je što riječ u slici nije tako jednostavna stvar kao riječ u tekstu. Postoje tri koraka kojima to možemo postići:

1. detekcija značajki (engl. *feature detection*)
2. formiranje opisnika značajki (engl. *feature description*)
3. generiranje rječnika (engl. *codebook generation*)

### 2.2.1. Detekcija značajki

Postupak detekcije značajki podrazumijeva pronalazak nekoliko regija unutar slike koje su potencijalni kandidati za riječi. Postoji više načina na koje nalazimo i odabiremo značajke.

Regularna mreža (engl. *regular grid*) je najjednostavnija, no efikasna metoda za detekciju značajki. Slika se segmentira horizontalnim i vertikalnim linijama te se dobivena polja odabiru kao riječi. Ova metoda daje dobre rezultate pri rješavanju problema kategorizacije prirodnih scena [8]. Slika 2.3 prikazuje primjenu regularne mreže na ulaznu sliku.



Slika 2.3

Detektor interesnih točaka (engl. *interest point detector*) je metoda koja pokušava pronaći istaknute dijelove slike, kao što su rubovi i kutevi. Jedan od načina dobivanja potencijalnih interesnih točaka jest primjena SIFT (engl. *Scale-invariant feature transform* [11]) algoritma.

Prvi korak algoritma je konvolucija ulazne slike s Gausovim filtrima različitih veličina, na različitim veličinama slike te računanje razlike između tako dobivenih

slika (engl. *difference of Gaussians, DoG* [10]). Maksimumi ili minimumi tih razlika su potencijalne interesne točke.

Konvolucija slike  $I(x, y)$  s Gausovim filtrom  $G(x, y, k\sigma)$ , gdje je  $k\sigma$  veličina tog filtra, iznosi je  $L(x, y, k\sigma) = G(x, y, k\sigma) * I(x, y)$ .

Razlika konvoluiranih slika je  $D(x, y, \sigma) = L(x, y, k_i\sigma) - L(x, y, k_j\sigma)$ .

Rezultat razlike tih slika je nova slika iste veličine na kojoj se zatim traže kandidati. Svaki piksel te slike se uspoređuje sa svojih 8 susjeda te sa još 9 susjeda prve veće (ili manje) slike obrađene istim postupkom. Ako je promatrani piksel najmanji ili najveći u tom svom susjedstvu, onda se proglašava kandidatom.

Nakon ovog koraka potrebno je smanjiti broj kandidata za interesne točke, odnosno odbaciti one koje su podložne šumu na slici, te niskog kontrasta u odnosu na okolinu. Potrebno je, dakle, izmjeriti stabilnost svih kandidata.

Prvo se provodi interpolacija kandidata  $D(x, y, \sigma)$  Taylorovim redom drugog stupnja:  $D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x$ , gdje je  $x$  odmak od kandidata. Ako taj odmak iznosi više od 0.5, sigurno je da se taj ekstrem nalazi bliže nekoj drugoj kandidatnoj točki. Ako je vrijednost  $D(x)$  manja od 0.03, taj kandidat odbacujemo, jer je niskog kontrasta.

Funkcija  $D(x, y, \sigma)$  ima jak odziv uz rubove objekata, stoga trebamo odbaciti kandidate koji imaju loše određene položaje a jak odziv. Kako bi smo to postigli, potrebno je riješiti matricu  $H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix}$ . Možemo izbjeći traženje svojstvenih vrijednosti time što nas jedino zanima omjer tih dviju vrijednosti. Veću od tih vrijednosti nazivamo  $\alpha$ , a manju  $\beta$ .

$$D_{xx} + D_{yy} = \alpha + \beta.$$

$$Det(H) = D_{xx} + D_{yy} - D_{xy}^2 = \alpha\beta$$



$$\frac{D_{xx} + D_{yy}}{\text{Det}(H)} = \frac{\alpha + \beta}{\alpha\beta}$$

Ako je determinanta od  $H$  negativna, taj se kandidat odbacuje. U suprotnom se uspoređuje omjer  $\frac{\alpha + \beta}{\alpha\beta}$  s određenim pragom. Ukoliko je taj omjer veći od praga, kandidat se također odbacuje.

Sljedeći korak je ključan za postizanje invarijantnosti na rotaciju i uvećanje. Svakoj interesnoj točki koja je prošla selekciju dodjeljuje se orijentacija na temelju značajki iz okoline te točke, odnosno orijentacija gradijenata u tom dijelu. Prvo se uzima slika  $L(x, y, \sigma)$  zamučena Gausovim filtrom veličine  $\sigma$ , te se računaju amplituda i orijentacija gradijenta:

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2}$$

$$\theta(x, y) = \arctan \frac{L(x, y + 1) - L(x, y - 1)}{L(x + 1, y) - L(x - 1, y)}$$

Izračuni ove dvije jednadžbe vrše se za svaki pixel u okolini interesne točke. Formira se orijentacijski histogram s 36 smjerova (svaki po 10 stupnjeva). U histogramu nalazimo točku najveće vrijednosti, te se od nje i točaka do 80% vrijednosti najveće točke određuje orijentacija interesne točke.

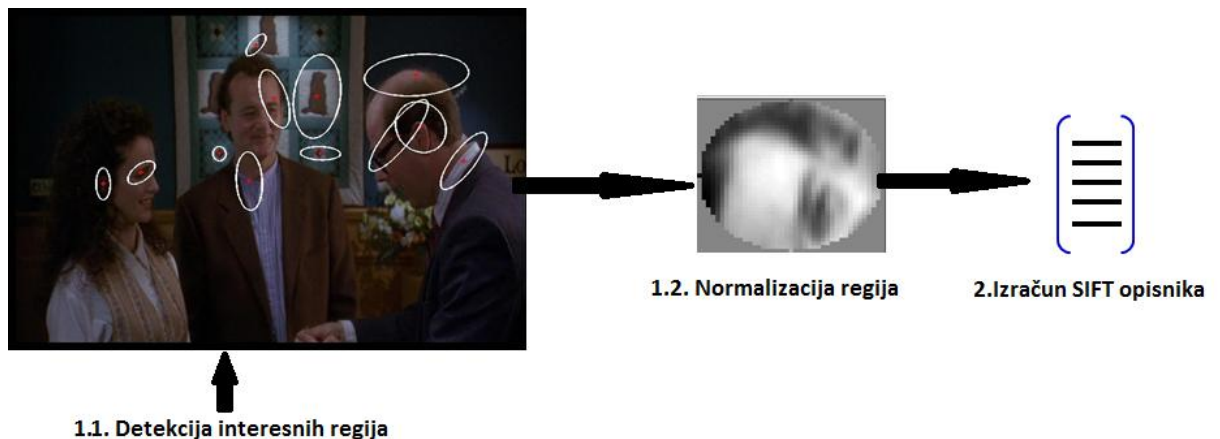
Zadnji korak algoritma SIFT je formiranje opisnika interesne točke, što je ujedno i drugi korak pronalaženja riječi u slici.

### 2.2.2. Formiranje opisnika značajki

U ovom koraku potrebno je regije slike reprezentirati numeričkim vektorima. U slučaju SIFT opisnika, ti vektori su veličine 128. Uzimamo područja veličine 4x4 pixela oko interesne točke te se za svako od tih područja stvara vektor s 8 potencijalnih smjerova. Ovakav vektor potrebno je normalizirati kako bi, uz već postignute invarijantnosti na položaj, skaliranje i rotaciju, poboljšali invarijantnost na osvjetljenje.

Nakon ovog koraka, svaka je ulazna slika je zapravo kolekcija SIFT opisnika, čiji poredak je nebitan.

Slika 2.4 prikazuje pronalazak značajki te izračun njihovog SIFT opisnika [9].



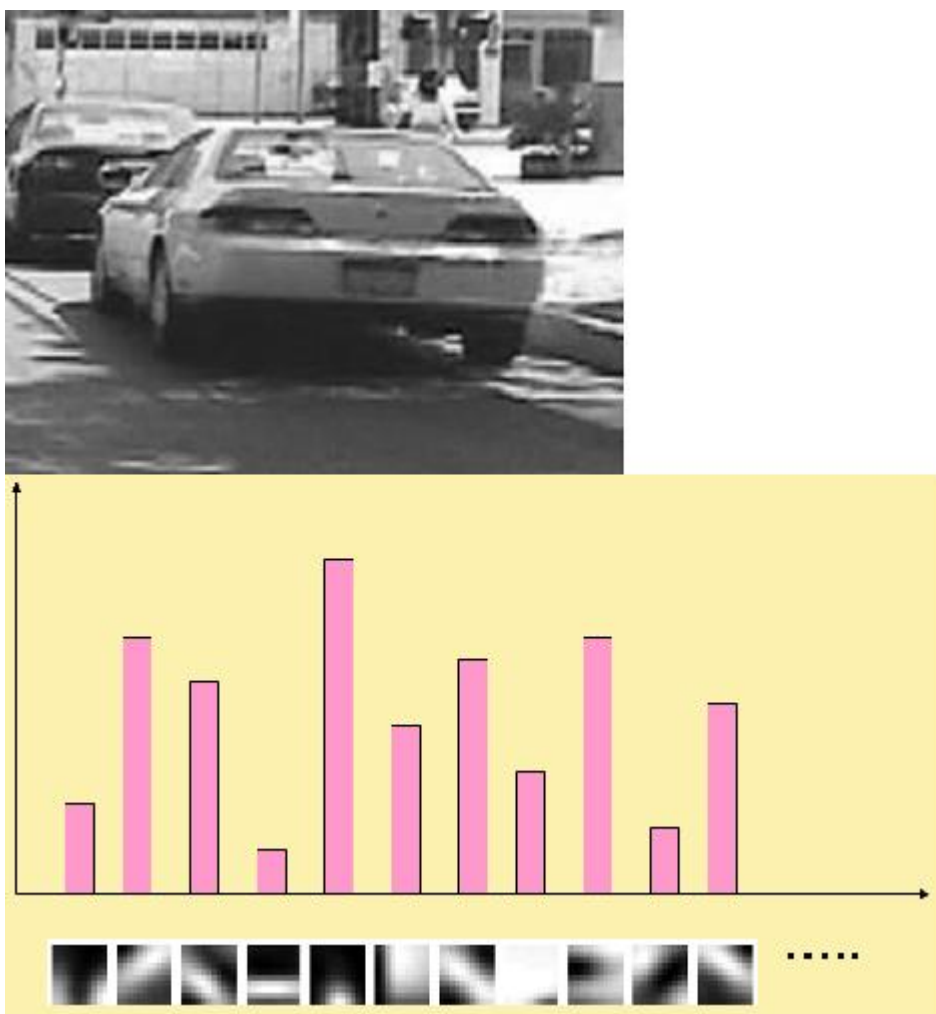
Slika 2.4

### 2.2.3. Formiranje rječnika

Postupak formiranja vizualnog rječnika (engl. *visual dictionary*, *codebook*) svodi se na određivanje kodnih riječi, odnosno riječi koje su reprezentativne za nekoliko sličnih riječi. Jedna od jednostavnijih mogućnosti jest provođenje K-means algoritma grupiranja nad svim riječima. Kodne riječi možemo definirati kao centre dobivenih grupa. Broj grupa K kod algoritma grupiranja odgovara veličini rječnika. Sada se svaka riječ unutar slike može mapirati u jednu kodnu riječ postupkom grupiranja te se slika može prikazati kao histogram kodnih riječi.

Također, vrlo je važno odabrati pravu veličinu rječnika. Ako odaberemo premalu veličinu, kodne riječi neće biti reprezentativne za sve riječi iz slike. Ako pak odaberemo preveliku veličinu rječnika, dolazi do prenaučivosti.

Slika 2.5 prikazuje histogram kodnih riječi jedne slike automobila.



Slika 2.5

### 3. Učenje i kategorizacija korištenjem modela zbirka značajki

Za sada smo uspjeli sliku prikazati kao histogram riječi, bazirajući se na modelu zbirki značajki. Postoji mnogo metoda strojnog učenja koje koriste taj model za razne zadatke vezane uz slike, npr. kategorizacija objekata. Te metode možemo ugrubo podijeliti u dvije skupine: generativne i diskriminativne metode.

#### 3.1. Generativni modeli

Generativni modeli su probabilistički modeli kojima generiramo distribucije ulaznih primjera  $x$  i razreda  $C_j$ , a tako ih nazivamo jer omogućavaju generiranje sintetičkih primjera u ulaznom prostoru (uzorkovanjem iz zajedničke distribucije  $P(x, C_j)$ ). [12]

Neki od generativnih modela su naivni Bayesov klasifikator, model LDA (engl. *latent Dirichlet allocation* [5]) i model pLSA (engl. *probabilistic latent semantic analysis* [6]).

Za početak, korištena notacija:

- $M$  – veličina rječnika
- $w_n$  –  $V$ -dimenzionalni vektor koji predstavlja riječ te ima sve komponente, osim jedne, jednake nuli. Komponenta koja nije jednaka nuli označava kojoj grupi (kodnoj riječi) taj vektor pripada npr.  $w_n = [0, 0, \dots, 1, \dots, 0]$
- $\mathbf{w}$  – kolekcija svih riječi u jednom dokumentu (slici), npr.  $\mathbf{w} = [w_1, w_2, \dots, w_{58}]$
- $d_j$  –  $j$ -ti dokument (slika) u skupu svih ulaznih dokumenata
- $c$  – kategorija dokumenta (slike)
- $z$  – tema dijela slike (riječi), odnosno kodna riječ u rječniku kojoj ta riječ pripada

### 3.1.1. Naivni Bayesov klasifikator

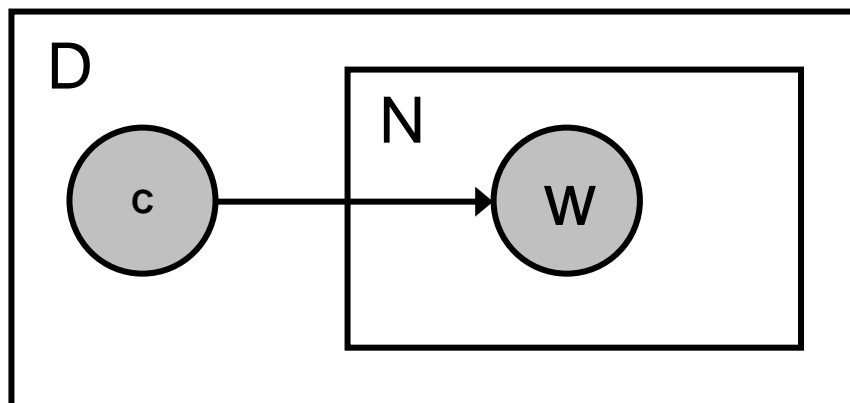
Naivni Bayesov klasifikator je najjednostavniji probabilistički generativni model, čija je glavna pretpostavka da svaka kategorija ima vlastitu distribuciju nad rječnikom te da se te distribucije znatno razlikuju. Uzmimo za primjer kategoriju „lice“. Ta kategorija naglašava kodne riječi koje predstavljaju „nos“, „oko“ i „usta“, dok kategorija „auto“ naglašava kodne riječi koje predstavljaju „kotač“ i „prozor“. Naivni Bayesov klasifikator na temelju ulaznih primjera uči distribucije za različite kategorije.

Odluka kategorizacije je predstavljena sljedećim izrazom:

$$c^* = \underset{c}{\operatorname{argmax}} p(c|\mathbf{w}) = \underset{c}{\operatorname{argmax}} p(c)p(\mathbf{w}|c) = \underset{c}{\operatorname{argmax}} p(c) \prod_{n=1}^N p(w_n|c)$$

Slika 3.1 prikazuje grafički model naivnog Bayesovog klasifikatora. Općenito, za grafičke prikaze vrijedi:

- čvorovi unutar kvadrata se ponavljaju onoliko puta koliko je navedeno u gornjem lijevom kutu kvadrata
- ispunjeni krugovi predstavljaju slučajne varijable
- prazni krugovi predstavljaju skrivene (latentne) slučajne varijable



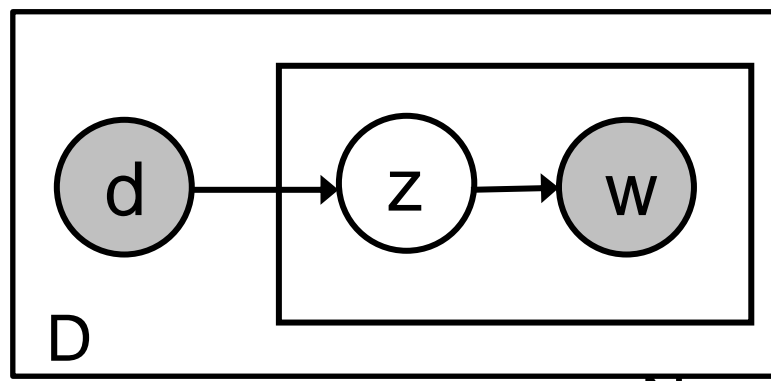
Slika 3.1

### 3.1.2. Hijerarhijski Bayesovi modeli

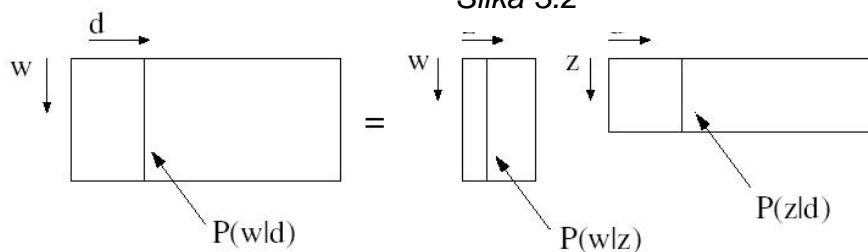
Osnovna pretpostavka modela naivnog Bayesovog klasifikatora ponekad nije ispunjena, odnosno neka slika može sadržavati riječi koje se mapiraju u teme karakteristične različitim kategorijama. Dva najpoznatija takva modela su model LDA (engl. *latent Dirichlet allocation* [5]) i model pLSA (engl. *probabilistic latent semantic analysis* [6]). U nastavku je opisan model pLSA [2].

Cilj pLSA modela je pronalazak tema  $z_k$  za svaki ulazni dokument, odnosno sliku. Pretpostavimo da imamo  $N$  dokumenata koji sadrže riječi iz rječnika veličine  $M$ . Kolekciju dokumenata tada predstavlja matrica dimenzija  $M \times N$  gdje element  $n(w_i, d_j)$  predstavlja broj pojavljivanja riječi  $w_i$  u dokumentu  $d_j$ . Postoji i dodatna, latentna varijabla  $z_k$  koja predstavlja temu riječi te je povezana uz svako pojavljivanje riječi  $w_i$  u dokumentu  $d_j$ .

Slika 3.2 prikazuje grafički model pLSA, dok Slika 3.3 prikazuje odnos vjerojatnosti.



Slika 3.2



Slika 3.3

Vrijedi  $p(w_i|d_j) = \sum_{k=1}^K P(z_k|d_j)P(w_i|z_k)$ , gdje je  $P(z_k|d_j)$  vjerojatnost pojavljivanja teme  $z_k$  u dokumentu  $d_j$ , a  $P(w_i|z_k)$  je vjerojatnost pojavljivanja riječi  $w_i$  u temi  $z_k$ .

Klasifikacija se sada svodi na:

$$z^* = \underset{z}{\operatorname{argmax}} p(z|d)$$

### 3.2. Diskriminativni modeli

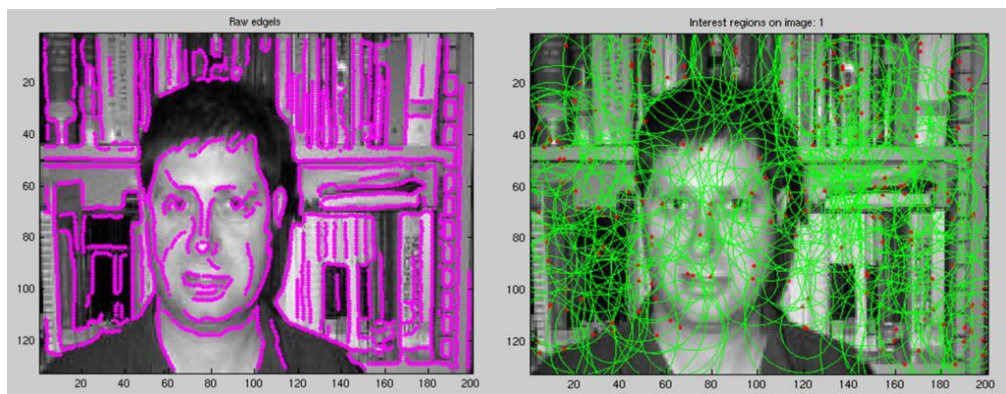
Diskriminativne metode definiraju funkciju  $f(x)$  koja za ulazni podatak  $x$  direktno računa razred kojem on pripada, odnosno izravno modeliraju granicu između razreda. Kod modela zbirke značajki, kao ulazni podaci se uzimaju ranije formirane riječi. Neke od diskriminativnih metoda su:

- klasifikator „najbliži susjed“ (engl. *nearest neighbor*) pridjeljuje razred podatka za učenje koji je u prostoru značajki najbliži svakom podatku za testiranje
- klasifikator  $k$ -najbližih susjeda (engl. *k-nearest neighbors*,  $k$ -NN) za novu točku traži  $k$  najbližih točaka u prostoru značajki te zatim razredi tih  $k$  točaka „glasuju“ za razred nove točke
- Stroj s potpornim vektorima (engl. *Support vector machine*) je binarni klasifikator koji maksimizira marginu između pozitivnih i negativnih primjera za učenje

## 4. Demonstracija funkcionalnosti

U ovom poglavlju prikazani su rezultati eksperimenta iz [1], kojim smo rješavali problem detekcije lica na slici. Koraci algoritma su sljedeći:

1. Prvo se poziva skripta *do\_random\_indices.m* koja nasumično dijeli ulazni skup slika na skup za učenje i skup za testiranje.
2. Skripta *do\_preprocessing.m* skalira sve ulazne slike na 200 pixela širine.
3. Zatim se poziva skripta *do\_interest\_op.m*. Za razliku od ranije opisanog detektora interesnih točaka, korišten je jednostavniji detektor koji se bazira na detekciji rubova Canny-evom metodom, te nasumičnim odabirom točaka na tim rubovima. Sljedeća slika prikazuje rubove dobivene Canny-evom metodom i odabrane interesne točke:



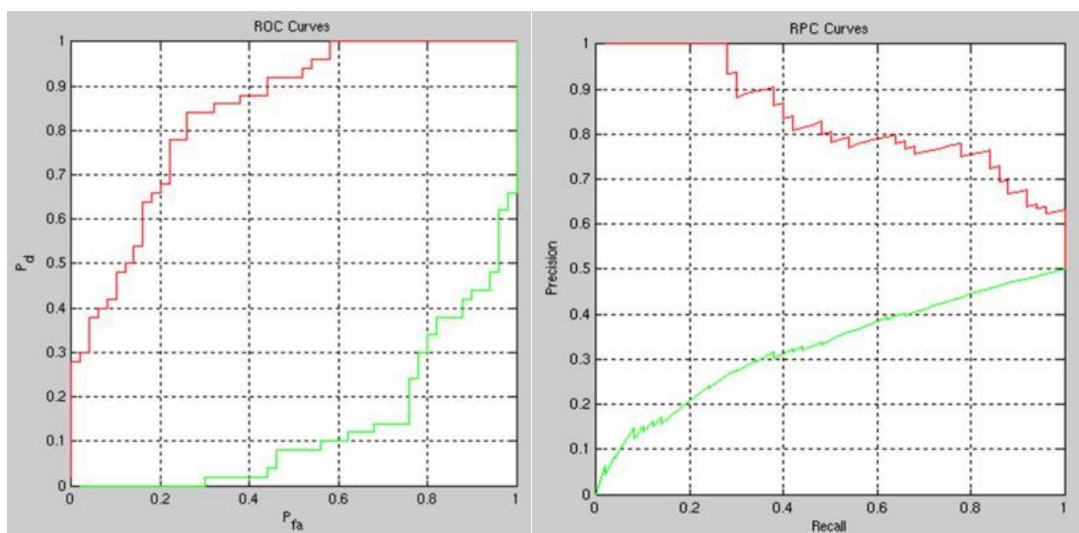
4. Skripta *do\_representation.m* stvara SIFT opisnike za svaku od interesnih točaka.
5. Skripta *do\_form\_codebook.m* provodi *k-means* grupiranje nad svim SIFT vektorima svih ulaznih slika te stvara rječnik.



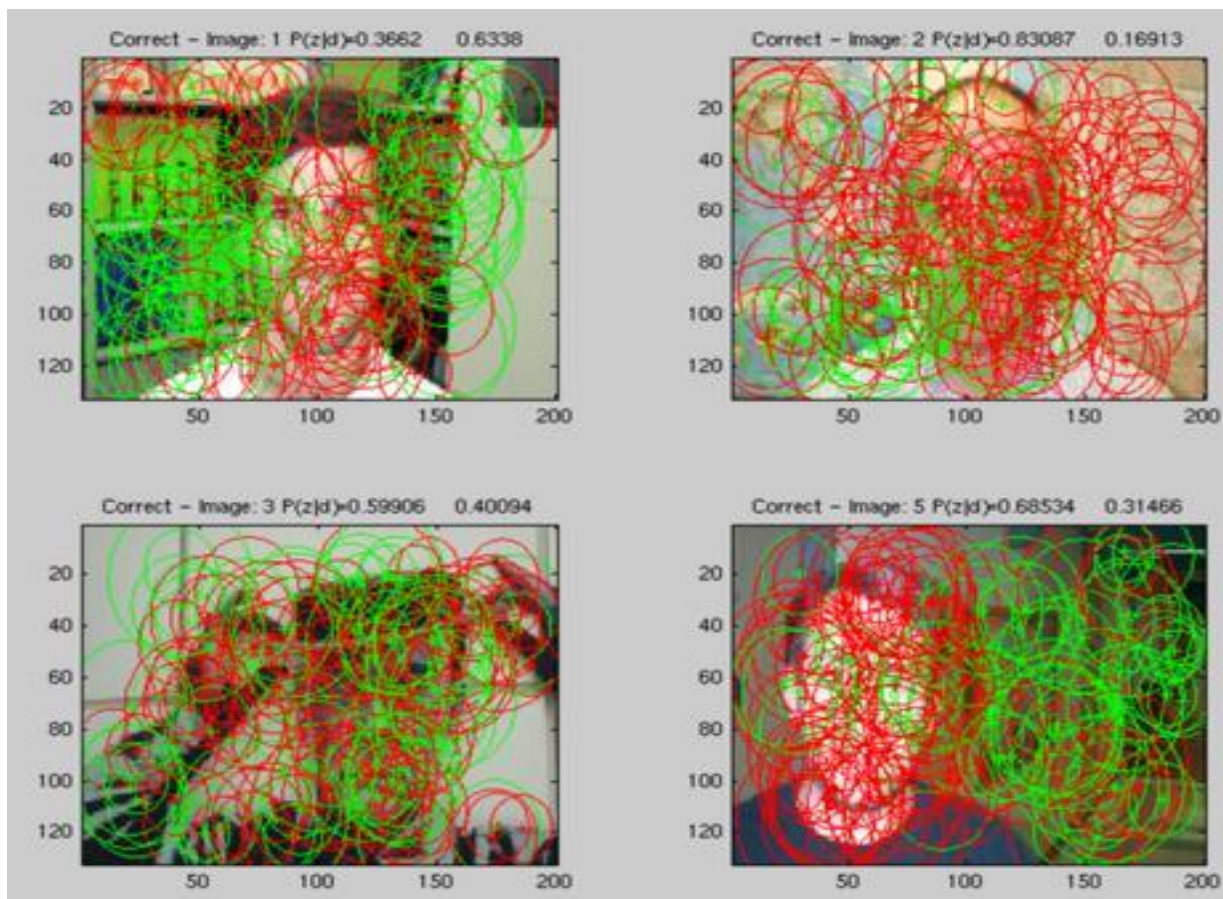
6. Skripta `do_vq.m` na temelju interesnih točaka i rječnika stvara vektor koji sadrži sljedeće varijable, za svaku interesnu točku:

```
>> load interest_0001
>> whos
Name          Size          Bytes  Class
angle         1x154         1232   double array
descriptor    128x154       19712  uint8 array
descriptor_vq 1x154         1232   double array
histogram     1x500         4000   double array
scale         1x154         1232   double array
score         1x154         1232   double array
x             1x154         1232   double array
y             1x154         1232   double array
```

7. Skripta `do_plsa.m` uči pLSA model na temelju ulaznih slika.
8. Skripta `do_plsa_evaluation.m` crta grafove za  $P(w_i|z_k)$  i  $P(d_j|z_k)$ , zatim ROC krivulju (engl. *receiver operating characteristic*), odnosno *true positive* stopu u ovisnosti o stopi *false positive*, te RPC krivulju (engl. *recall precision curve*). Primjer ROC i RPC krivulja dan je na sljedećoj slici (zeleno označava temu „lice“, dok crveno označava temu „ne\_lice“:



Na sljedećoj slici prikazano je nekoliko slika s njihovim interesnim točkama obojanima bojom odgovarajuće teme:



Osim pLSA modela, ova implementacija nudi i mogućnost korištenja naivnog Bayesovog klasifikatora, koji daje nešto lošije rezultate [1].

## 5. Zaključak

Model zbirke značajki nudi osiromašenu reprezentaciju podataka, zato što ignorira bilo kakve prostorne odnose značajki. Unatoč tome, taj model pokazao se vrlo uspješnim u području procesiranja i analize prirodnog jezika, ponajviše radi visokih diskriminativnih vrijednosti određenih riječi u tekstu. Pokazuje se da se isti model može uspješno primijeniti i u području računalnog vida uparivanjem s bilo kojim klasifikatorom, kao što je naivni Bayesov klasifikator ili Stroj s potpornim vektorima.

Većina pristupa prepoznavanju objekata uključuje neku vrstu nadgledanog učenja – od specificiranja same lokacije objekta do jednostavnog označavanja razreda kojem objekt pripada. Pritom je vrlo važna veličina skupa podataka za učenje, a vrlo često se nalazimo u situaciji da je veličina skupa označenih podataka premala ili da ih uopće nemamo. U slučaju nenadgledanog učenja model zbirke značajki u kombinaciji s hijerarhijskim Bayesovim modelima klasifikatora (pLSA i LDA) se pokazao odličnim izborom.

## 6. Literatura

1. Autori: Li Fei-Fei, Rob Fergus  
Naziv: Two bag-of-words classifiers  
<http://people.csail.mit.edu/fergus/iccv2005/bagwords.html>
2. Autori: Josef Sivic, Bryan C. Russell, Alexei A. Efros, Andrew Zisserman, William T. Freeman  
Naziv: Discovering objects and their location in images  
Godina: 2005.
3. Bag of words model in computer vision  
[http://en.wikipedia.org/wiki/Bag\\_of\\_words\\_model\\_in\\_computer\\_vision](http://en.wikipedia.org/wiki/Bag_of_words_model_in_computer_vision)
4. Bag of words model  
[http://en.wikipedia.org/wiki/Bag\\_of\\_words\\_model](http://en.wikipedia.org/wiki/Bag_of_words_model)
5. Latent Dirichlet allocation  
[http://en.wikipedia.org/wiki/Latent\\_Dirichlet\\_allocation](http://en.wikipedia.org/wiki/Latent_Dirichlet_allocation)
6. Probabilistic latent semantic analysis  
[http://en.wikipedia.org/wiki/Probabilistic\\_latent\\_semantic\\_analysis](http://en.wikipedia.org/wiki/Probabilistic_latent_semantic_analysis)
7. Bayesian spam filtering  
[http://en.wikipedia.org/wiki/Bayesian\\_spam\\_filtering](http://en.wikipedia.org/wiki/Bayesian_spam_filtering)
8. Autori: Li Fei-Fei, P. Perona  
Naziv: A Bayesian Hierarchical Model for Learning Natural Scene Categories  
Godina: 2005.
9. Autor: Li Fei-Fei  
Naziv: Bag-of-words models
10. Difference of Gaussians  
[http://en.wikipedia.org/wiki/Difference\\_of\\_Gaussians](http://en.wikipedia.org/wiki/Difference_of_Gaussians)

11. Scale-invariant feature transform

[http://en.wikipedia.org/wiki/Scale-invariant\\_feature\\_transform](http://en.wikipedia.org/wiki/Scale-invariant_feature_transform)

12. Autori: Bojana Dalbelo Bašić, Jan Šnajder

Naziv: Strojno učenje, bilješke za predavanja 1

Godina: 2010.

## Sažetak

U ovom seminarskom radu opisan je pristup kategorizaciji objekata korištenjem zbirke značajki (engl. *bag-of-words*). Postupak vuče korijene iz područja procesiranja prirodnog jezika. U tom području, dokumenti su reprezentirani riječima, međutim, redoslijed tih riječi je nebitan. Sličan pristup primjenjuje se na slike – one se reprezentiraju kao skupine regija ili objekata, pri čemu se zanemaruje prostorni odnos tih objekata. Osim detekcije i izlučivanja zbirke značajki, opisan je postupak učenja i klasifikacije objekata različitim klasifikatorima, kao što su naivni Bayesov klasifikator i metoda pLSA (engl. *probabilistic latent semantic analysis*). Na kraju je dana demonstracija funkcionalnosti modela zbirke značajki pri rješavanju problema detekcije lica.