

# Klasifikacija slika korištenjem lokalnih deskriptora

## 1 Kernel funkcije

Linearna SVM koristi linearni kernel

$$K(\mathbf{h}, \mathbf{h}') = \sum_{i=1}^d h_i h'_i \quad (1)$$

za određivanje sličnosti između vektora  $\mathbf{h}$  i  $\mathbf{h}'$ .

U nekim slučajevima bolje performanse se dobijaju korištenjem nelinearnih kernela. U ovoj vježbi ćemo eksperimentirati i sa Hellingerovim kernelom koji je oblika

$$K(\mathbf{h}, \mathbf{h}') = \sum_{i=1}^d \sqrt{h_i h'_i}. \quad (2)$$

## 2 Uputstva za praktičan rad

### 2.1 VLFeat biblioteka

Pored ranije korištene biblioteke LIBSVM u ovoj vježbi će biti korištena i biblioteka VLFeat koju je moguće preuzeti na adresi <http://vlfeat.org>. Preuzetu ZIP arhivu raspakujte na pogodno mjesto, a zatim izvršite skript `vl_setup` iz foldera `toolbox`. Ovaj skript će u MATLAB-ovu listu putanja dodati direktorijume u kojima je instalirana biblioteka. Iz ove biblioteke ćemo iskoristiti funkcije za izračunavanje lokalnih deskriptora i formiranje BoW reprezentacije slike.

Za izračunavanje lokalnih SIFT deskriptora u pravilno raspoređenim lokacijama koristi se funkcija `vl_phow` čija je sintaksa:

```
[frames, descrs] = vl_phow(im, options)
```

- `im` je ulazna slika za koju se izračunavaju deskriptori;
- `options` su opcije kojima se podešavaju npr. veličina prozora u kojem se izračunava SIFT deskriptor i korak sa kojim se pomjera prozor. Za kompletan spisak opcija i detalje sintakse pogledati dokumentaciju;
- `frames` je matrica kod koje svaka kolona odgovara jednoj lokaciji na kojoj je izračunat SIFT deskriptor. `frames(1:2, :)` sadrže x i y koordinatu lokacije, `frames(3, :)` sadrži kontrast SIFT deskriptora, a `frames(4, :)` veličinu prostorne ćelije deskriptora;
- `descrs` po kolonama sadrži 128-dimenzionalne SIFT deskriptore izračunate u lokacijama u `frames`.

Kodna knjiga se formira k-means klasterizacijom SIFT deskriptora. Koristićemo implementaciju k-means algoritma iz VLFeat biblioteke čija je sintaksa:

```
[c, a, energy] = vl_kmeans(data, numcenters, options);
```

- `data` po kolonama sadrži vektore podataka koji se klasterizuju;
- `numcenters` je broj centroida;
- `options` su opcije algoritma. Za spisak opcija i detalje sintakse pogledati dokumentaciju;
- `c` po kolonama sadrži centroide klastera;
- `a` je vektor pripadnosti podataka centroidima;
- `energy` je energija klasterizacije;
- `options` omogućava izbor različitih opcija inicijalizacije, metoda optimizacije i različitih metrika za računanje udaljenosti. Za kompletan spisak opcija i detalje sintakse pogledati dokumentaciju.

Pošto je klasterizacija velikog broja lokalnih deskriptora vrlo spora, kodnu knjigu ćemo formirati klasterizacijom podskupa deskriptora, a ostale lokalne deskriptore ćemo kvantovati pridruživanjem klasteru sa najbližim centroidom. Za efikasno približno određivanje najbližih susjeda koristićemo k-d stablo koje se formira pomoću funkcije

```
forest = vl_kdtreebuild(x)
```

- `x` je matrica u čijim kolonama se nalaze vektori podataka;
- `forest` je indeksna struktura.

Približni najbliži susjedi se određuju pomoću funkcije

```
[index, dist] = vl_kdtreequery(kdtree, x, y)
```

koja za svaku kolonu iz matrice `y` pronalazi najbližu kolonu iz matrice `x` koristeći indeksnu strukturu `kdtree`.

## 2.2 Formiranje reprezentacije slika

MATLAB kod za formiranje BoW reprezentacije slika i njihovu klasifikaciju dat je na web stranici predmeta <http://dsp.etfbl.net/pms/bow.zip>. Kod je organizovan na sljedeći način:

- `make_codebook`  
Formiranje kodne knjige korištenjem `conf.num_patches` deskriptora izdvojenih sa svake od `conf.num_training_codebook` slika. Dobijena kodna knjiga se nalazi u kolonama matrice `vocab`.
- `extract_features`  
Izračunavanje reprezentacije za slike koje se nalaze na putanji zadatoj u `conf.images_filepath`. Dobijene reprezentacije su u kolonama matrice `hist`. Imena slika su u ćelijskom nizu `images`.
- `aerial_svm_nfold`  
Podjela kolekcije slika na trening i test skup, obučavanje SVM klasifikatora i njegovo testiranje.

## 3 Zadaci

1. Upoznati se sa arhitekturom klasifikatora i strukturama podataka. Preuzmite kolekciju aero-slika sa adrese <http://dsp.etfbl.net/pms/bl.zip>. Podesiti putanju do kolekcije slika i ostale parametre. Pošto faze formiranja kodne knjige i izračunavanja reprezentacija slika mogu dugo trajati, kodna knjiga i reprezentacije aero-slika su već pripremljene i nalaze se u `mat` fajlovima.
2. Testirati klasifikator aero-slika pokretanjem programa `aerial_svm_nfold`. Implementiran je SVM klasifikator sa linearnim kernelom. Kolika tačnost klasifikacije se dobija obučavanjem klasifikatora korištenjem po 10 slika iz svake klase?

3. BoW deskriptori su histogrami pojavljivanja kodnih riječi i ukoliko se kao mjera sličnosti koristi kosinusna metrika prikladno je izvršiti njihovu  $L_1$  normalizaciju. S druge strane, klasifikator korišten u ovom primjeru koristi linearni kernel (1). U ovom slučaju je prikladnije koristiti  $L_2$  normalizaciju histograma. Zašto? Modifikujte `aerial_svm_nfold` tako da umjesto  $L_2$  normalizacije koristi  $L_1$  normalizaciju ili ne koristi normalizaciju uopšte i uporedite dobijene rezultate.
4. U mnogim slučajevima korištenjem nelinearnih kernela se dobijaju bolji rezultati nego korištenjem linearnog kernela. Modifikovati program `aerial_svm_nfold` tako da se umjesto linearnog kernela koristi Hellingerov kernel (2). Ovaj kernel se može efikasno realizovati izračunavanjem kvadratnog korijena pojedinih komponenata deskriptora, a zatim korištenjem linearnog kernela. Kolika tačnost klasifikacije se dobija u ovom slučaju? Obratite pažnju na adekvatnu normalizaciju deskriptora. Zašto je izračunavanje kvadratnog korijena pojedinih komponenata deskriptora i korištenje linearnog kernela ekvivalentno korištenju Hellingerovog kernela?
5. Varirajte broj trening slika od 5-20 slika iz svake klase. Kako se mijenjaju performanse klasifikatora?
6. Napisati kod u MATLAB-u kojim će se za zadatu klasu prikazivati određen broj najviše rangiranih testnih slika. Upotrebiti priloženu funkciju `thumbs`.
7. Promijenite veličinu kodne knjige na 1000 kodnih riječi, formirajte kodnu knjigu i izračunajte reprezentacije slika. Kako se mijenjaju performanse klasifikatora?

## 4 Domaći zadatak

1. Corel 1k kolekcija slika se sastoji od 1000 slika podijeljenih u 10 kategorija: Afrika, plaže, spomenici, autobusi, dinosaurusi, slonovi, cvijeće, konji, planine i hrana. Zadatak je da se realizuje sistem za pretraživanje baze koji će moći da vrati slike koje pripadaju zadatoj klasi, npr. slonovi. Kako bi se ovo postiglo potrebno je obučiti klasifikator koji će klasifikovati slike na osnovu pripadnosti određenoj klasi i optimizovati njegove performanse u pretraživanju. Za obučavanje klasifikatora potrebno je formirati trening skup. Kao pozitivne primjere iskoristite slike koje ste dobili pretraživanjem Interneta po zadatom ter-

minu. Pronađite, npr. pet slika slonova koristeći Google Image Search i sačuvajte ih. Kao negativne primjere iskoristite slike iz arhive <http://dsp.etfbl.net/pms/background.zip>.

2. Napisati program u MATLAB-u kojim će se formirati kodna knjiga korištenjem pozitivnih i negativnih trening slika iz prethodne tačke. Pošto veličine trening slika mogu biti različite, a veličine test slika su fiksne i iznose  $256 \times 384$  piksela korisno je standardizovati veličine trening slika tako da duža ivica bude 384 piksela. Formirati kodnu knjigu sa 1000 kodnih riječi. Kako bi se izbjegli problemi sa memorijom pri formiranju kodne knjige nije potrebno koristiti sve slike, niti sve lokalne deskriptore izdvojene za korištene slike. Testne slike se ne koriste za formiranje kodne knjige.
3. Napisati program u MATLAB-u kojim će se, na osnovu kodne knjige iz prethodne tačke, formirati reprezentacije svih slika (trening i test) kao skupova riječi. I ovdje je korisno standardizovati veličine slika tako da duža ivica bude 384 piksela.
4. Obučiti binarni SVM klasifikator sa linearnim kernelom korištenjem reprezentacija slika iz trening skupa. Napisati program u MATLAB-u za pretraživanje Corel 1k kolekcije slika korištenjem SVM klasifikatora tako što će se slike klasifikovati korištenjem obučenog klasifikatora, a zatim se rezultati pretraživanja sortirati na osnovu odziva SVM klasifikatora. Prikazati prvih 24 rezultata pretraživanja. Izračunati koliko relevantnih slika se nalazi u prvih 24 rezultata pretraživanja – preciznost sistema za 24 rezultata  $P(24)$ . Izračunati preciznost i odziv za ovaj sistem i nacrtati P-R grafik. Koristiti funkciju `v1_pr` iz `VLFeat` biblioteke. Odrediti tačnost klasifikatora.
5. Poboljšati performanse sistema za pretraživanje, odnosno, klasifikatora dodavanjem novih trening primjera, po mogućnosti što raznovrsnijih.
6. Varirati veličinu kodne knjige i ispitati njen uticaj na performanse sistema. Performanse sistema ocijeniti kao u prethodnoj tački.
7. Umjesto linearnog kernela obučiti SVM klasifikator sa Hellingerovim kernelom i uporediti rezultate sa rezultatima dobijenim korištenjem linearne SVM.
8. Na opisani način realizovati sisteme za pretraživanje slika iz tri različite kategorije. Dati primjere rezultata pretraživanja i izračunati performanse dobijenih sistema.