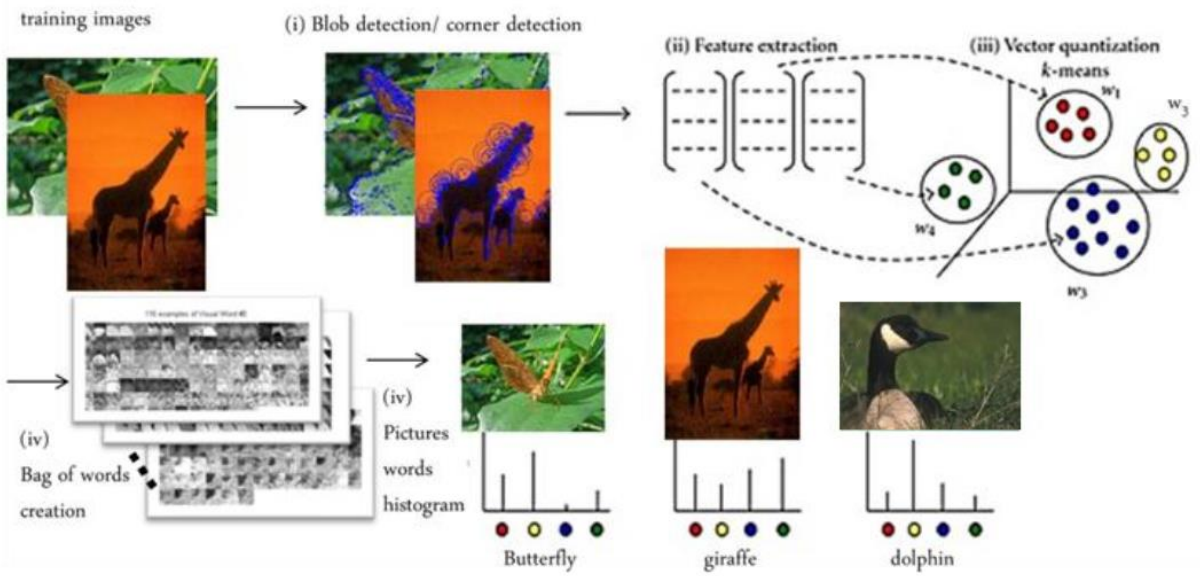
VEŽBA 12

Klasifikacija sadržaja slike

Potrebno predznanje

- Poznavanje programskog jezika C
- 2D signali
- RGB i YUV prostori boja
- Filtriranje 2D signala u vremenskom domenu
- Računanje histograma slike

ZADATAK



1.1 Zadatak 1

Završiti implementaciju funkcije

- `vector<double> getBoWDescripton(uchar input[], int xSize, int ySize, const BagOfWordsModel& model)`

Funkcija vrši računanje vektora obeležja na osnovu prosleđenog modela skupa reči (*bag of words*).

Računanje deskriptora se vrši na sledeći način:

- Za prosledjenu sliku *input* pronaći ključne tačke upotrebom SIFT algoritma..
- Za svaku izračunatu tačku pronaći kojem klasteru iz prosleđenog BoW modela pripada. Tačka pripada onom klasteru za koji je udaljenost do centra klastera najmanja.
- Uvećati polje u histogramu koje odgovara indeksu klastera za 1.
- Kada je čitav histogram izračunat, normalizovati vrednosti tako što ćete svako polje unutar histograma podeliti sa zbirom svih elemenata u histogramu.

Parametri funkcije su:

- `input` – ulazna slika u RGB formatu
- `xSize` – horizontalna dimenzija slike
- `xSize` – vertikalna dimenzija slike
- `model` – BoW model (sadrži veličinu modela, veličinu svakog vektora)

Struktura BagOfWordsModel sadrži sledeća polja:

`vector<vector<double>> model`	Vektor tačaka koje predstavljaju centralne tačke klastera. Svaka tačka je opisana sa vektorom dužine featureVectorSize.
`int numOfClusters`	Broj tačaka (klastera). Veličina vektora model.
`int featureVectorSize`	Broj obeležja sa kojim je opisana svaka tačka.

1.2 Zadatak 2

Dopuniti funkciju:

- `void trainSVM(const ImgDataBase& dataBase, bool createNewBoWModel)` – koja vrši obuku SVM za klasifikaciju sadržaja slike u prosleđenoj bazi podataka.

Parametri funkcije su:

- `dataBase` – Baza slika nad kojom je potrebno uraditi obuku
- `createNewBoWModel` – Naznaka da li je potrebno praviti novi BoW model ili učitati postojeći iz datoteke.

Baza podataka predstavlja listu slika predstavljenih strukturom DBImage. Struktura DBImage data je sa:

`QImage* image`	Slika u RGB formatu predstavljena QT klasom Qimage
`int labelNumber;`	Redni broj klase kojoj slika pripada. Koristi se u fazi treniranja i za proveru uspešnosti klasifikacije u fazi prepoznavanja
`std::string label`	Tekstualni opis klase.

Funkcija TrainSVM u prvoj fazi priprema ulaz za SVM u formi strukture svm_problem. Parametri SVM su dati u fromi strukture svm_param. Funkcija koja vrši postavljanje parametara na podrazumevane vrednosti naziva se setDefaultParams i u njoj je moguće praviti izmene parametara.

Nakon toga vrši se pravljenje novog ili učitavanje BoW modela iz datoteke. Koristeći dobijeni model potrebno je za svaku sliku u bazi izračunati vektor obeležja koristeći funkciju getBoWDescripton. Vektor obeležja se potom zajedno sa rednim brojem klase dodaje u strukturu svmProblemSet.

Kada je problem upsešno specifiran neophodno je pozvati funkciju za obuku SVM. Ukoliko je treniranje uspešno vrši se čuvanje dobijenog modela u tekstualni fajl vezba12.model.

Deklaracije funkcije svm_train je data sa:

```
svm_model *svm_train(const svm_problem *prob, const svm_parameter *param);
```

gde prob predstavlja zadati problem (vektor ulaznih obeležja i indeks klase za svaku sliku), a param skup parametara za SVM: Povratna vrednost je obučeni model za klasifikaciju.

1.3 Zadatak 3

Dopuniti funkciju:

- `int predictSingleImage(uchar input[], int xSize, int ySize, BagOfWordsModel& bowModel, svm_model* svmModel))` – koja vrši klasifikaciju jedne slike koriste'i prosle]eni BoW model i SVM model

Parametri funkcije su:

- `input` – ulazna slika u RGB formatu
- `xSize` – horizontalna dimenzija slike
- `xSize` – vertikalna dimenzija slike
- `bowModel` – BoW model (sadrži veličinu modela, veličinu svakog vektora)
- `svmModel` – Unapred obučen model SVM

Koristeći prosleđeni BoW model potrebno je za ulaznu sliku izračunati vektor obeležja koristeći funkciju getBoWDescripton. Na osnovu vektora obeležja se priprema odgovarajući ulaz za SVM (realizovano).

Nakon toga potrebno je pozvati funkciju svm_predict i proslediti joj SVM model i pripremeljeni ulazni vektor obeležja.

Deklaracije funkcije svm_predict je data sa:

```
double svm_predict(const struct svm_model *model, const struct svm_node *x);
```

gde model predstavlja unapred obučeni SVM model, x vector ulaznih obeležja za zadatu sliku. Povratna vrednost funkcije je redni broj klase kojoj slika pripada.

1.4 Zadatak 4

U projektu ImageDSP, unutar funkcije `imageProcessingFun` dodati poziv funkcije za predikciju klase kojoj odabrana slika pripada. Na osnovu rezultata ispisati preko slike kojoj klasi pripada.