<Naziv projekta>

Tehnička dokumentacija

Verzija <1.0>

Studentski tim: <Ime i prezime>

<Ime i prezime>

<Ime i prezime>

**....**

Nastavnik: <Ime i prezime>

Sadržaj

1. Opis razvijenog proizvoda 4

2. Tehničke značajke 5

3. Upute za korištenje 6

4. Opis zadatka 6

5. Opis postupka 7

6. Teoretske osnove 9

6.1 LBP (*engl. Local binary pattern*) – lokalna binarna značajka 9

6.2 Matrica ko-pojava lokalnih binarnih značajki (LBPCM) 11

6.3 Klasifikator k najbližih susjeda 12

7. Literatura 13

Tehnička dokumentacija

***Na koji način koristiti predložak?***

Dokument se po potrebi može prilagoditi potrebama pojedinog projekta promjenom predloženih naslova predloženih poglavlja, kao i eventualnim dodavanjem novih poglavlja i potpoglavlja.

Cilj dokumenta je opisati rezultat rada studentskog tima, problem koji je riješen u okviru projekta, korištenu tehnologiju, mogućnosti i značajke dobivenog proizvoda i sl. Razinu detalja opisanu u ovom dokumentu studentski tim treba dogovoriti s nastavnikom.

***Literatura:***

U tekstu rada treba biti navedena literatura svugdje gdje je tekst, slika ili grafički prikaz preuzet ili se temelji na nekom pisanom predlošku. Literatura se navodi iza zaključka. U tekstu se literatura navodi unutar zagrada s navođenjem prvog autora i godine izdanja, npr. (Martinis, 1998).

***Primjer citiranja knjige:***

Prezime, inicijal(i) imena autora. Naslov: podnaslov. Podatak o izdanju. Mjesto izdavanja: Nakladnik, godina izdavanja.

***Primjer citiranja članka u časopisu:***

Prezime, inicijal(i) imena autora. Naslov članka: podnaslov. Naziv časopisa. Oznaka sveska/godišta, broj(godina), str. početna-završna.

***Primjer citiranja rada sa konferencije:***

Prezime, inicijal(i) imena autora. Naslov rada: podnaslov. Naslov zbornika, mjesto održavanja konferencije, (godina), str. početna-završna.

***Primjer citiranja doktorskog, magistarskog ili diplomskog rada:***

Prezime, inicijal(i) imena autora. Naslov. Vrsta rada. Ustanova na kojoj je rad obranjen, godina.

***Primjer citiranja www izvora:***

Ime(na) autora (ako je/su poznata), naslov dokumenta, datum nastanka (ako se razlikuje od datuma pristupa izvoru), naslov potpunog djela (italic), potpuna http adresa, datum pristupa dokumentu.

***Ostale upute***

U svim dokumentima obvezno primjenjivati SI jedinice. Slike, formule i tablice potrebno je numerirati. Opis tablice stavlja se iznad, a opis slike ispod nje. U opisu slike ili tablice pišu se samo podaci neophodni za njeno razumijevanje (npr. Slika 6. Pojačalo s promjenljivim pojačanjem). Dodatna objašnjenja daju se u tekstu uz povezivanje sa slikom ili tablicom. Osi i parametri na slikama i grafičkim prikazima trebaju biti obilježeni. Daljnji opis tog grafičkog prikaza treba se nalaziti u tekstu rada. Formule se obilježavaju brojevima u zagradi, uz desni rub stranice, a u tekstu se poziva na broj formule.

# Opis zadatka

Klasifikacija gustoće mnoštva je izuzetno važna, ali isto tako i jedna od najtežih zadaća u inteligentnom video nadzoru. Mnoštva mogu biti vrlo različita po svojoj razdiobi i paleti boja. Predloženo je puno metoda koje se zasnivaju na značajkama teksture za rješavanje ovog problema. Većina postojećih metoda procjenjuje gustoću mnoštva na razini cijele slike pritom ignorirajući lokalna područja. U ovom radu predlaže se nova metoda bazirana na matrici ko-pojava lokalnih binarnih značajka (*engl. Local binary pattern co-occurrence matrix* - LBPCM).

Matrica se konstruira od nekoliko preklapajućih ćelija slikovnog bloka koji se klasificira u nekoliko razreda gustoća. Iako je gustoća mnoštva definirana kao broj ljudi po jedinici površine, brojanje ljudi nije uvijek potrebno za analizu gustoće. Polus i drugi[1] prvi su predstavili tok mnoštva koji je uvelike prihvaćen. Prema njima gustoća mnoštva može se svrstati u četiri razreda: slobodan tok, ograničen tok, gusti tok i zakrčen tok.

LBPCM opisuje statistička svojstva, ali isto tako i prostornu informaciju pritom iskorištavajući puni potencijal LBP-a za lokalne značajke teksture. Dodatno, konstruiramo LBPCM na sivim i gradijentnim slikama u svrhu poboljšanja točnosti klasifikacije.









Slika 1. Rezanje izvorne slike u više slikovnih blokova



Nakon podjele u slikovne blokove slijedi podjela svakog slikovnog bloka u više preklapajućih ćelija nakon čega su izračunate značajke teksture iz svake ćelije. Značajke svih ćelija spoje se u jedan vektor značajki koji predstavlja pojedini slikovni blok.

Prikaz jedne ćelije slikovnog bloka nalazi se na slici 2. Svaka ćelija je unaprijed određene veličine (32x32, 64x64,…) i tehnikom kliznog prozora ta se ćelija pomiče po slikovnom bloku. Za pomak ćelije u horizontalnom i vertikalnom smjeru može se uzeti npr. polovina veličine ćelije (ako je 32x32 onda se ćelija pomiče za 16 piksela udesno ili dolje).

Slika 2. Prikaz jedne ćelije slikovnog bloka

Dobiveni vektor značajki preda se 1-NN klasifikatoru na treniranje.

# Teoretske osnove

U ovom poglavlju objašnjavaju se detaljnije termini i metodologije korištene u radu.

## LBP (*engl. Local binary pattern*) – lokalna binarna značajka

LBP se razvio kao poseban slučaj jedinice teksture (*engl. Texture unit*) koje je prvi put predstavljena u radu Dong-Chen He i Li Wang [2]. Jedinica teksture se definira na sljedeći način.

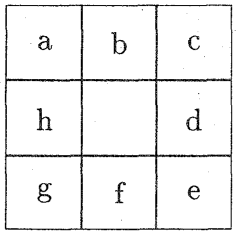
U kvadratnoj rasterskoj digitalnoj slici svaki piksel okružen je sa osam susjednih piksela (osim krajnjih piksela na rubu slike). Lokalna informacija teksture izvlači se iz susjedstva 3x3 koje predstavlja najmanju potpunu jedinicu (u smislu postojanja osam smjerova oko piksela). Obzirom na 3x3 susjedstvo koje će biti označeno skupom od devet elemenata V = {V0, V1, … , V8}, gdje V0 predstavlja intenzitet srednjeg piksela, {V1, V2, … , V8} predstavlja skup intenziteta susjednih piksela. Jedinica teksture je skup od osam elemenata TU = {E1, E2, … , E8}, gdje su

Ei = (1, 2, … , 8) i određuje se formulom:

za sve i = 1, 2, … , 8. Kako svaki element TU ima 3 mogućnosti, ukupno postoji 38 = 6561 jedinica teksture.

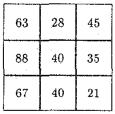
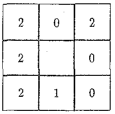
Broj NTU predstavlja pojedinu jedinicu teksture, a dobiva se na sljedeći način:

Moguće je elemente poredati na različite načine, ali se uzima način prikazan na slici 3. Prvi element je element a, zatim b i sve do h.



Slika 3. Prikaz poretka jedinica teksture u susjedstvu srednjeg piksela.

Primjer dobivanja TU





NTU = 6095



Slika 4. susjedstvo

V = {40, 63, 28, 45, 35, 21, 40, 67, 88}

Slika 5. jedinica teksture

TU = {2, 0, 2, 0, 0, 1, 2, 2}

Nakon definicije jedinice teksture, lako je objasniti lokalnu binarnu značajku. LBP se izračunava na potpuno jednak način osim što postoje samo dvije vrijednosti (0 i 1) umjesto 3. Ovakav način predložen je od T. Ojale i drugih [3]. Ovaj pristup je pogodniji jer postoji ukupno 28 = 256 različitih kombinacija LPB-a, odnosno toliko koliko ima i razina svjetline pa su zbog toga slike u sivim tonovima prigodne.

.

|  |  |  |
| --- | --- | --- |
| 71 | 171 | 190 |
| 5 | 55 | 78 |
| 24 | 12 | 78 |

|  |  |  |
| --- | --- | --- |
| 1 | 1 | 1 |
| 0 |  | 1 |
| 0 | 0 | 1 |

LBP = 31



Slika 6. susjedstvo

V = {55, 71, 171, 190, 78, 78 12, 24, 5}

Slika 7.

Susjedstvo nakon usporedbe s centralnim pikselom

U prethodnim primjerima razmatrano je susjedstvo udaljeno samo za jedan piksel, međutim, to ne mora uvijek biti tako. Moguće je definirati susjedstvo kao kružnicu radijusa, npr. 3, što za sobom povlači veći broj piksela koji sudjeluju u nastajanju LBP-a.

Primjer slike dimenzija 64x64 i primijenjenog LBP operatora na tu sliku s radijusom redom 1, 2 i 3.



Slika 8. Primjer dobivenih LBP slika iz izvorne slike uz različite radijuse oko centralnog piksela

Kao dodatno proširenje na osnovni operator LBP je takozvani jednolični uzorak (*engl. uniform pattern*) koji može biti korišten za smanjenje dimenzija vektora značajki, ali i kao jednostavni deskriptor otporan na rotaciju slike [4]. Ideja je motivirana činjenicom da se neki binarni uzorci pojavljuju više od nekih drugih u slikama s teksturom. LBP se naziva jednoličnim ako ima najviše dvije 0-1 ili 1-0 tranzicije. Npr. 00010000 ima dvije tranzicije, ali 01010100 ima šest tranzicija i on nije jednoličan. U izračunima LBP histograma, svaki uzorak ima svoju grupu (poziciju), a svi koji nisu jednolični uzorci se svrstavaju u zajedničku grupu. Koristeći jednolične uzorke, dimenzionalnost vektora značajki možemo smanjiti sa 256 na 59 (toliko ima jednoličnih uzoraka ako koristimo radijus jedan piksel).

## Matrica ko-pojava lokalnih binarnih značajki (LBPCM)

Matrica ko-pojava predložena je od strane Haralicka i dr. [5], a Marana [6] ju koristi za procjenu gustoće mnoštva. Tipično se matrica izračunava nad sivim slikama pa je poznatija pod imenom matrica ko-pojava sivih razina (*engl.Gray level co-occurrence matrix - GLCM*).

GLCM je statistička metoda procjene združenih uvjetnih vjerojatnosnih funkcija gustoće, .

Svaka predstavlja vjerojatnost da se par sivih razinajavlja na udaljenosti u smjeru na slici. Korištenje izvorne matrice ko-pojava je nepraktično jer sadrži veliku količinu informacija pa se zbog toga koriste deskriptori za mjere teksture. Tih deskriptora postoji više [5], a ovdje je odabrano njih 4. Korišteni deskriptori su: energija, kontrast, homogenost i entropija.

Primjer računanja funkcije gustoće vjerojatnosti za . Slika sa sivim razinama na kojoj se izračunava funkcija je na slici 9.

|  |  |  |  |
| --- | --- | --- | --- |
| 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 2 | 2 | 2 |
| 2 | 2 | 3 | 3 |

Slika 9. Primjer sivih razina neke slike

Slika 10. Prikaz matrice koja je nastala na primjeru slike 9.

Objasniti matricu možemo na sljedeći način. Uzmimo element (0,0) u matrici i potražimo sva mjesta u izvornoj slici sivih razina gdje se pojavljuju 0 jedna kraj druge (lijevo ili desno). Vidimo da ukupno ima takve četiri pozicije pošto svaka 0 ima kraj sebe lijevo ili desno jednu nulu. Kao još jedan primjer uzmimo (3,0). Pošto slika ne sadrži kombinaciju sivih razina gdje se nalazi 3 kraj 0, u matricu na mjesto indeksa (3,0) pišemo 0 i tako za sve preostale indekse.

Kako je matrica , matrica vjerojatnosti, potrebno je provesti normalizaciju. Ukupno postoji 256x256 vrijednosti, a kao vrijednosti su uzete sljedeće vrijednosti: 0, 45, 90 i 135.

## Klasifikator k najbližih susjeda

U raspoznavanju uzoraka klasifikator k najbližih susjeda(*k*-NN) je neparametarska metoda korištena za klasifikaciju i regresiju. Kod *k-*NN klasifikacije izlaz klasifikatora je pripadnost nekom od razreda. Objekt se klasificira na način da svaki od *k* najbližih susjeda „glasa“ za neki od razreda, a pripadnost je onda onaj razred koji je većinski izabran od strane susjeda. Za broj susjeda se odabire pozitivan broj koji je tipično malen. Ako je *k* = 1 tada je pripadnost objekta pridružena razredu najbližeg susjeda.

*k*-NN je vrsta učenja koje se temelji na primjerima gdje se funkcija aproksimira lokalno i svo se računanje odgađa do faze klasifikacije. Primjeri su spremljeni u memoriji te se svaki novi objekt uspoređuje s njima ne bi li se utvrdila pripadnost nekom od razreda.

Algoritam

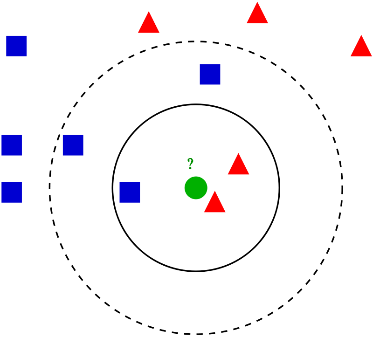
Primjeri za treniranje su vektori u n-dimenzionalnom prostoru značajki, svaki s oznakom pripadnosti nekom od razreda. Faza treniranja sastoji se od spremanja vektora značajki i oznaka pripadnosti razredu. U fazi klasifikacije, neoznačenom vektoru se dodjeljuje oznaka koja je najčešća među *k* najbližih susjeda.

Često korištena mjera udaljenosti za kontinuirane varijable je euklidska udaljenost. Za diskretne varijable, npr. kod klasifikacije teksta, kao udaljenost se može koristiti Hammingova udaljenost.

Nedostatak standardnih metoda gdje se „većinski glasa“ je problem određivanja pripadnosti razredu u slučaju kada su distribucije razreda nepravilne. Primjer takve nepravilnosti je prevelika zastupljenost nekog od razreda u skupu za treniranje. Jedan od načina kako prevladati ovaj problem je dodijeliti svakom od susjeda neku težinu koja je obrnuto proporcionalna udaljenosti do uzorka koji se klasificira.

Izbor parametara

Najbolji izbor za konstantu *k* ovisi o podatcima nad kojima se vrši treniranje. Velike vrijednosti *k* smanjuju efekte šuma u klasifikaciji, ali znatno smanjuju granice između pojedinih razreda. Poseban slučaj kada se pripadnost određuje na temelju samo jednog susjeda je klasifikator najbližeg susjeda. Točnost klasifikacije algoritma može biti značajno degradirana korištenjem nekih značajki u vektorima koje ne doprinose međusobnoj diskriminaciji pojedinih vektora ili su skale pojedinih vrijednosti vektora neprilagođene njihovoj značajnosti pa je njihov ukupan doprinos prevelik.



Slika 11. Primjer *k*-NN klasifikacije. Testni primjer(zeleni krug) trebao bi biti klasificiran u crvene trokute ili plave kvadrate. Ako je *k* = 3 (kružnica s punom crtom), razred kojem se zeleni krug pridjeljuje je razred crvenih trokuta jer unutar kruga ima dva trokuta, a samo jedan plavi kvadrat. Ukoliko promijenimo *k* = 5, odluka klasifikatora se mijenja jer unutar radijusa toga kruga ima više kvadrata. Ovaj primjer pokazuje koliko je važno izabrati pravu vrijednost parametra *k*.

# Opis postupka

Predlaže se tehnika klizećeg prozora za klasifikaciju i lociranje područja mnoštva. Prvo se sve slike pretvore u sive tonove, a zatim se svaka izvorna slika izreže u više blokova, npr. 16, koji variraju u gustoći mnoštva, pozadini i uvjetima osvjetljenja. Ti blokovi su zatim označeni s odgovarajućim razinama gustoće prema [1] uz dodatnu oznaku „no flow“ koja predstavlja blokove u kojima nema nikakvog mnoštva. Nakon dobivenih blokova, nad svakim se primjenjuje operacija LBP-a. Nad tim LBP blokovima se tehnikom kliznog prozora generira LBPCM koja služi za dobivanje vrijednosti funkcija: energije, entropije, kontrasta i homogenosti. Svaka od tih vrijednosti se doda u vektor značajki, jedan poslije druge i taj se proces ponavlja za svaku ćeliju u bloku, što na kraju dovodi do vektora veličine.

DODATI IZVOR I FORMULU Po završetku stvaranja, vektora značajki se predaju *k*-NN klasifikator koji ih interno sprema za daljnju klasifikaciju još neviđenih uzoraka. Na kraju slijedi evaluacija klasifikatora i moguće naknadne izmjene parametara te pregled krajnjih rezultata.

Označavanje pripadnosti razredu svakog bloka



Set slika za treniranje



Blokovi u sivim razinama



Tehnika klizećeg prozora

LBPCM svake ćelije

Izračun Haralickovih funkcija i stvaranje vektora značajki

*k*-NN klasifikator

Treniranje i klasifikacija

Slika 1. Prikaz dijagrama tijeka kroz pojedine faze prilikom postupka treniranja



Model *k*-NN klasifikatora

LBPCM svake ćelije

Izračun Haralickovih funkcija i stvaranje vektora značajki



Tehnika klizećeg prozora



Blokovi u sivim razinama

Klasificiranje svakog bloka slike u neki od razreda

Slika koju treba klasificirati

Slika 2. Prikaz dijagrama tijeka kroz pojedine faze prilikom klasifikacije dosad neviđene slike

# Upute za korištenje i prikaz međukoraka

Programska podrška ovdje obrađivanog postupaka pisana je u Python 3.6 programskom jeziku. Usto je potrebno instalirati dodatne Python knjižnice: OpenCV, scikit-learn i PIL.

Za pokretanje aplikacije putem grafičkog sučelja potrebno je kliknuti na app.py ili putem terminala upisati naredbu „python app.py“ iz vršnog direktorija projekta. Nakon što je aplikacija pokrenuta otvoren je sljedeći prozor.

# Literatura

[1] A. Polus, J. Schofer and A. Ushpiz, „Pedestrian Flow and Level of Service,“ J. Transportation Eng. 109(1), 46-56, 1983.

[2] DC. He and L. Wang (1990), "Texture Unit, Texture Spectrum, And Texture Analysis", Geoscience and Remote Sensing, IEEE Transactions on, vol. 28, pp. 509 - 512.

[3] T. Ojala, M. Pietikäinen and D. Harwood, "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions," Proceedings of the 12th IAPR International Conference on Pattern Recognition, vol. 1, pp. 582 – 585, 1994.

[4] Barkan et. al "Fast High Dimensional Vector Multiplication Face Recognition." Proceedings of ICCV 2013

[5] Robert M Haralick; K Shanmugam; Its'hak Dinstein (1973). "Textural Features for Image Classification", IEEE Transactions on Systems, Man, and Cybernetics. SMC-3 (6): 610–621.

[6] A. N. Marana, S. A. VelaStin, L. F. Costa and R. A. Lotufo., "Estimation of crowd density using image processing," Proc. IEE Colloquium Image Processing for Security Applications, 11:1-8, 1997.

[7] OpenCV, <https://opencv.org/>

[8] scikit-learn, <https://scikit-learn.org/stable/>

[9] PIL, <https://pypi.org/project/Pillow/>