

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 000

**Procjena gustoće mnoštva ljudi
utemeljena na matrici pojavnosti
lokalnih binarnih uzoraka**

Antonio Kuzminski

Zagreb, svibanj 2020.

*Umjesto ove stranice umetnite izvornik Vašeg rada.
Da bi ste uklonili ovu stranicu obrišite naredbu \izvornik.*

SADRŽAJ

1. Uvod	1
2. Teorijske osnove	2
2.0.1. LBP (<i>engl. Local binary pattern</i>) - lokalna binarna značajka .	2
2.0.2. Matrica pojavnosti sivih razina	5
2.0.3. Klasifikator k najbližih susjeda - k -NN	7
2.0.4. Stroj s potpornim vektorima(<i>engl. Support vector machine</i>) - SVM	9
2.0.5. Gradijent slike	13
2.0.6. Sobelov operator	14
2.0.7. Haralickove značajke	15
3. Opis postupka	16
4. Upute za korištenje i prikaz međukoraka	20
4.0.1. Podjela slika u podslike	22
4.0.2. Označavanje podslika	24
4.0.3. Stvaranje vektora značajki	25
4.0.4. Klasifikacija slike postojećim modelima	29
5. Tehnika klizećeg okna	34
6. Rezultati eksperimenta	38
7. Prikladnost korištenog postupka na drugome skupu podataka	43
8. Zaključak	44
Literatura	45

Popis slika	46
Popis tablica	48
A. Teksturne značajke	50
B. Rezultati eksperimenta	53

1. Uvod

U svjetlu problema prouzročenim slabim upravljanjem mnoštva poput gužvi ili blokada ljudi, postoji povećana potreba za računalnim modelima koji su sposobni analizirati velika mnoštva ljudi koristeći video prijenos iz sigurnosnih kamera. Brojenje mnoštva je ključna komponenta takvih automatiziranih sustava za analizu gustoće mnoštva. Sustav uključuje procjenu broja ljudi u mnoštvu kao i distribuciju gustoće mnoštva dotičnog mjesta. Identifikacija regija s gustoćom mnoštva većom od neke sigurne granice može pomoći kod izdavanja upozorenja i spriječiti potencijalne gužve. Procjena broja grupacija mnoštva može doprinijeti do bolje raspodjele ljudi, resursa i logistike kod organizacije događaja kojem prisupa veći broj ljudi ili kontroliranje već postojećih, npr. utakmica, prosvjedi ili bilo kakva druga okupljanja.

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 metoda bazirana na matrici pojavnosti lokalnih binarnih značajki (*engl.* Local binary pattern co-occurrence matrix - LBPCM).

Matrica se konstruira na svakoj ćeliji te se iz te matrice izračunaju Haralickove značajke koje se zatim dodaju u vektor značajki. Iako je gustoća mnoštva definirana kao broj ljudi po jedinici površine, brojanje ljudi nije uvijek potrebno za analizu gustoće. Polus i drugi¹ 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.

¹Polus et al. (1983)

2. Teorijske osnove

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

2.0.1. 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 ¹. 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 $V = \{V_0, V_1, \dots, V_8\}$, gdje V_0 predstavlja intenzite srednjeg piksela, a $\{V_0, V_1, \dots, V_8\}$ predstavlja entiteta susjednih piksela. Jedinica teksture je skup od osam elemenata $TU = \{E_1, E_2, \dots, E_8\}$, gdje su $E_i = (0, 1|i|2)$ i određuje se formulom

$$E_i = \begin{cases} 0, & \text{ako je } V_i \leq V_0 \\ 1, & \text{ako je } V_i = V_0 \\ 2, & \text{ako je } V_i \geq V_0 \end{cases}$$

za sve $i = 1, 2, \dots, 8$. Kako svaki element TU ima 3 mogućnosti, ukupno postoji $3^8 = 6561$ jedinica teksture. Broj N_{TU} predstavlja pojedinu jedinicu teksture, a dobiva se na sljedeći način:

$$N_{TU} = \sum_{i=0}^8 E_i \cdot 3^{i-1}$$

¹Dong-chen He i Li Wang (1990)

Moguće je elemente poredati na različite načine, ali se uzima način prikazan na slici 2.1. Prvi element je element a , zatim b i tako sve do h .

a	b	c
h		d
g	f	e

Slika 2.1: Prikaz poretka jedinica tekstone u susjedstvu srednjeg piksela.

Primjer: Dobivanje teksturne jedinice

62	20	41
79	40	30
130	40	10

2	0	2
2		0
2	1	0

Slika 2.2: Susjedstvo $V = \{40, 62, 20, 41, 30, 10, 40, 130, 79\}$ **Slika 2.3:** Jedinica tekstone $TU = \{2, 0, 2, 0, 0, 1, 2, 2\}$

$$TU = 2 \cdot 3^0 + 2 \cdot 3^2 + 3^5 + 2 \cdot 3^6 + 2 \cdot 3^7 + 2 \cdot 3^8 = 6094$$

Nakon definicije jedinice tekstone, 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. Ojala i drugih². Ovaj pristup je pogodniji jer postoji ukupno $2^8 = 256$ različitih kombinacija LPB-a, odnosno toliko koliko ima i razina svjetline pa su zbog toga slike u sivim tonovima prigodne.

1	1	1
0		1
0	0	1

71	171	190
5	55	78
24	12	78

Slika 2.4: Susjedstvo nakon usporedbe s centralnim pikselom **Slika 2.5:** Susjedstvo $V = \{55, 71, 171, 190, 78, 78, 12, 24, 5\}$

$$LBP = 2^0 + 2^1 + 2^2 + 2^3 + 2^4 = 31$$

²Ojala et al. (1994)

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 2.6: 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 ³. 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).

³Barkan et al. (2013)

2.0.2. Matrica pojavnosti sivih razina

Dotična matrica predložena je od strane Haralicka i dr. ⁴, a Marana ⁵ je koristi za procjenu gustoće mnoštva. Tipično se matrica izračunava nad crno bijelim slikama pa je poznatija pod imenom matrica pojavnosti sivih razina (*engl. Gray level co-occurrence matrix - GLCM*).

GLCM je statistička metoda procjene združenih uvjetnih vjerojatnosnih funkcija gustoće $f(i, j, d, \theta)$. Svaka $f(i, j, d, \theta)$ predstavlja vjerojatnost da se par sivih razina (i, j) javlja na udaljenosti d u smjeru θ na slici. Korištenje izvorne matrice u učenju različitih sustava klasifikacije je nepraktično jer sadrži veliku količinu informacije pa se zbog toga koriste Haralickove značajke koje su prikazane u dodatku A.

Primjer računanja funkcije gustoće vjerojatnosti za $d = 0, \theta = 0$. Slika sa sivim razinama koja služi za izračunavanje funkcije $f(i, j, 1, 0)$ nalazi se na slici 2.7.

0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	4

Slika 2.7: Primjer crno bijele slike

Postupak dobivanja elemenata matrice $f(i, j, 1, 0)$ opisan je u sljedećim koracima i zatim je prikazan izgled same matrice i izgled kada je provedena normalizacija matrice. Uzmimo da indeksi počinju od 0 i kao primjer element $(0, 0)$ koji predstavlja sva pojavljivanje dviju nula na udaljenosti 1 u smjeru od 0 stupnjeva, odnosno udesno. Iz slike je vidljivo da postoje takva dva mjesta: prvi i drugi redak. Stoga se u matricu $f(i, j, 1, 0)$ na mjestu s indeksom $(0, 0)$ upisuje broj 2.

Kao sljedeći element uzmimo $(2, 3)$. Potrebno je pronaći gdje se u izvornoj slici nalaze sive razine intenziteta 2 i 3 jedna kraj druge na udaljenosti 1 udesno. Iz slike je vidljivo da jedino mjesto pojave takvih razina u zdanjem retku, stoga u $f(i, j, 1, 0)$ na mjestu s indeksima 2, 3 upisujemo 1.

⁴Haralick (1979)

⁵Marana et al. (1997)

Daljnijim nastavkom istog postupka dolazi se do krajnjeg oblika matrice $f(i, j, 1, 0)$ koja izgleda ovako

$$\begin{bmatrix} 2 & 2 & 1 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0.167 & 0.167 & 0.083 & 0 \\ 0 & 0.167 & 0 & 0 \\ 0 & 0 & 0.25 & 0.083 \\ 0 & 0 & 0 & 0.083 \end{bmatrix}$$

Slika 2.8: Prikaz GLCM prije i nakon normalizacije

Prikazana GLCM je oblika 4, 4 jer sive razine izvorne slike mogu poprimiti 4 različite vrijednosti. U radu se koriste slike zapisane u 8-bitnom formatu što znači da se veličina matrice povećava na 256, 256 jer svaki piksel može poprimiti jednu od 256 različitih vrijednosti. Zbog veličine matrice mnoge kombinacije razina piksela se često ne pojavljuju u izvornim slikama što dovodi do toga da su matrice dosta prazne, odnosno puno indeksa imaju popunjene nulama.

Primjer: Dobivanje vrijednosti Haralickove značajke - energija. Uzmimo izvornu sliku. Energija se računa prema

$$f_1 = \sum_i \sum_j p(i, j)^2$$

gdje $p(i, j)$ označava vrijednost GLCM za udaljenost 1 i kut od 0 stupnjeva.

$$f_1 = 3 \cdot 0.167^2 + 3 \cdot 0.083^2 + 0.25^2 = 0.166834$$

U ovom radu se koristi matrica naziva LBPCM(*engl. Local binary pattern co-occurrence matrix*) koja je u suštini GLCM samo što se izračnavanje vrijednosti matrice radi nad slikom sivih razina nad kojom je primjenjen operator LBP.

2.0.3. Klasifikator k najbližih susjeda - k -NN

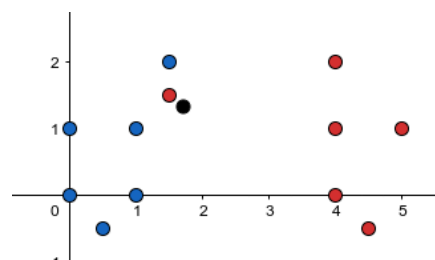
Ovaj klasifikator je jednostavan nadziran(*engl.* supervised) algoritam strojnog učenja koji može biti korišten za klasifikaciju i regresije. Problemi klasifikacije imaju kao izlaz diskretne vrijednosti dok s druge strane problemi regresije na izlazu poprimaju realne vrijednosti. k -NN algoritam pretpostavlja da se slične stvari nalaze jedne blizu drugih.

Nadzirani algoritmi strojnog učenja oslanjaju se na označene početne uzorke podataka kako bi dobili(naučili) funkciju koja daje prikladnu izlaznu vrijednost kada joj se preda neki, još do sada neviđeni, uzorak.

Primjer problema za koji bi mogli iskoristiti ovakav mehanizam je učenje male koje su boje objekti predloženi ispred nekakve vrste kamere, primjerice mobitela. Dijete bi usmjerilo kameru u neki objekt koji bi bio u fokusu i kao povratnu informaciju, aplikacija bi ispisala boju. Taj ispis bio bi pripadnost objekta nekom od predefiniраниh razreda, npr. nekoliko različitih boja. Aplikacija bi izvukla dio slike oko fokusiranog objekta, nekim funkcijama obrade slike stvorila vektor značajki i predala k -NN klasifikatoru te bi on kao izlaz dao neku labelu.

Slovo k u nazivu k -NN predstavlja broj susjeda koji sudjeluju u "glasanju". To glasanje predstavlja dodjeljivanje labele od strane k najbližih susjeda. Ukoliko je $k = 1$, labelu određuje samo najbliži susjed, ako je $k = 3$, razred s više glasova pobjeđuje te se kao labela novome, dosad nevđenome uzorku, dodjeljuje baš taj razred. Za broj susjeda se odabire pozitivan broj koji je tipično malen. Kako bi odabrali dobru vrijednost broja k poželjno je algoritam pokrenuti nekoliko puta.

Smanjenjem broja k odluke postaju sve nestabilnije, npr. ako postoji jedan primjer nekog drugog razreda(koji previše odskaka od vrijednosti svoga razreda) u blizini suprotnog razreda, može doći do krive klasifikacije što bi se inače izbjeglo povećanjem broja k . Crna točka iz slike 2.9. prikazuje još neklasificiran uzorak.



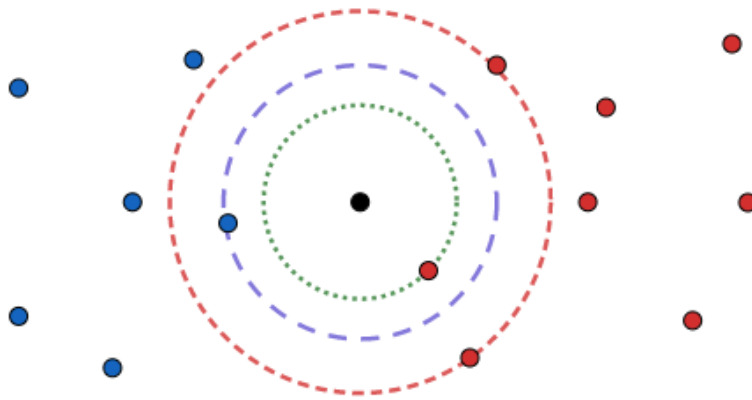
Slika 2.9: Prikaz dva razreda s jednim nestandardnim uzorkom

Ukoliko bi kao konstantu k izabrali 1, novi uzorak bi se klasificirao u crveni razred, a iz slike je jasno vidljivo da to nije poželjno. Točnost klasifikacije nije samo

uvjetovana izborom konstante k . Naime, ako su grupe pojedinih razreda raštrkane, odnosno njihov oblik je nepravilan, točnost klasifikacije se narušava. Uspješnost algoritma može biti značajno degradirana korištenjem nekih značajki u vektorima koje ne doprinose međusobnoj diskriminaciji pojedinih vektora ili su skale pojedinih komponenti vektora neprilagođene njihovoj značajnosti pa je njihov ukupan doprinos prevelik.

Primjer: Klasifikacija k -NN klasifikatorom.

Raspored uzoraka prikazan je na slici 2.10. Postoje 2 razreda - crveni i plavi. Uzorak koji se pokušava klasificirati označen je crnom bojom.



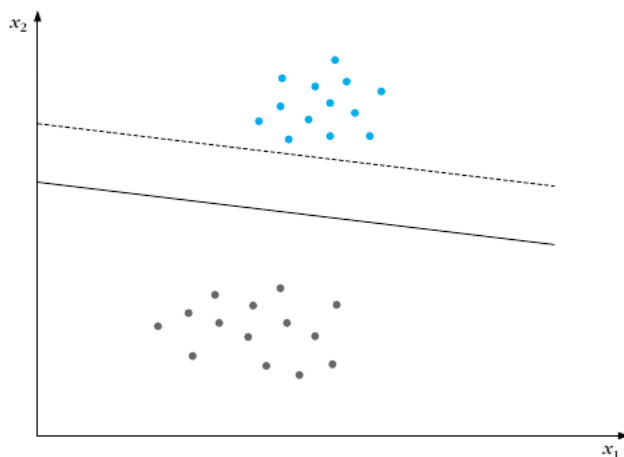
Slika 2.10: Prikaz uzoraka dva razreda i novog uzorka za klasifikaciju

Odabirom konstante k utječemo na krajnju dodjelu labele.

- $k = 1$ - Slučaj je prikazan crtkanom kružnicom zelene boje (kružnica najmanjeg radijusa). Pripadnost se određuje na temelju najbližeg susjeda pa labela novog uzorka postaje crveni razred.
- $k = 2$ - Najbliža dva susjeda. Slučaj je prikazan plavom crtkanom kružnicom. Ovakva situacija nije poželjna jer se unutar granica nalazi jednak broj pripadnika svakog od razreda. Kako bi razriješili situaciju mogli bi uzeti kao dodatnu mjeru udaljenost svakog od uzorka unutar kružnice do novog uzorka te pridodijeliti labelu bližeg. Kako bi izbjegli ovakve situacije dobra je praksa uzimati neparne vrijednosti konstante k .
- $k = 3$ - Situacija označena kružnicom najvećeg radijusa. Vidljivo je da su većinski pripadnici crvenog razreda unutar granica pa se novome uzorku dodjeljuje labela crvenog razreda.

2.0.4. Stroj s potpornim vektorima(engl. Support vector machine) - SVM

Ovaj klasifikator pripada vrsti linearnih klasifikatora što znači da ako postoji neki skup vektora X u kojem svaki od vektora $x_i, i = 0, 1, \dots, N$, pripada jednom od razreda ω_1 ili ω_2 te su ti razredi odvojivi hiperravninom u prostoru dimenzionalnosti N . Cilj klasifikatora je izgraditi hiperravninu $g(x) = \omega^T x + \omega_0 = 0$.



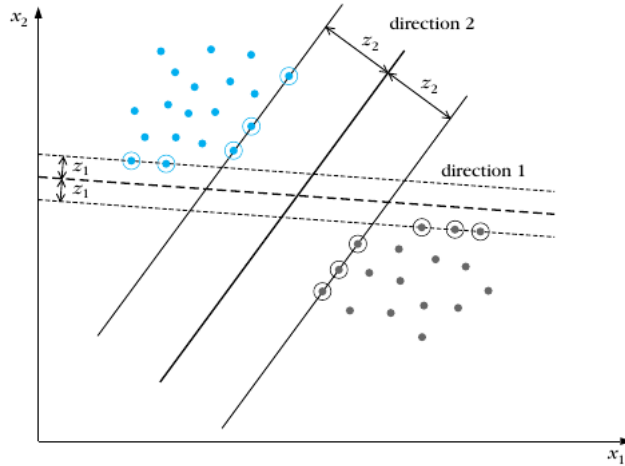
Slika 2.11: Prikaz dviju mogućih hiperravnina između dva razreda

Obje hiperravnine mogu poslužiti za odjeljivanje prikazanih razreda, međutim, uvijek je poželjnije imati čim otporniji klasifikator. Značenje otpornosti je u sljedećem. Ako se pojavi neki vektor koji je imalo ispod crtkane linije, klasificira se kao razred sivih točaka iako to ne bi bio slučaj malo pomnijim gledanjem slike. Cilj klasifikatora je izgraditi hiperravninu koja odjeljuje dva razreda, ali istovremeno ta hiperravnina mora biti što udaljenija od oba razreda čime se postiže otpornost klasifikatora na vektore koji malo odskaču od svojih razreda, bilo radi neke vrste šuma ili čega drugoga.

Svaka hiperravnina je određena svojim smjerom (određeno s w) i mjestom u prostoru (određeno s w_0). Pošto ne bismo željeli favorizirati niti jednu ravninu, trebamo odabrati jednu koja je jednako udaljena od odgovarajućih najbližih točaka razreda ω_1 i ω_2 . Odabrane hiperravnine označene su crnim linijama na slici 2. Margina udljenosti 1 je $2z_1$, a udaljenosti 2 je $2z_2$. Cilj je pronaći smjer koji daje maksimalnu moguću marginu. Margina je najmanja udaljenost između nekog vektora x_i i hiperravnine.

Želimo postići da je vrijednost $g(x)$ u blizini najbliže točke jednaka 1 za razred ω_1 , a -1 za razred ω_2 što je moguće zapisati na sljedeći način

$$y_i(w^T x_i + w_0) \geq 1, i = 1, 2, \dots, N \quad (2.1)$$



Slika 2.12: Prikaz potencijalnih hiperravnina koje dijele najbliže točke razreda

Kako bi dobili udaljenost dviju margina uzmimo jedan vektor x_p koji označava vektor iz središta koordinatnog sustava do neke točke na margini plavih točaka i drugi vektor x_s koji označava vektor iz središta koordinatnog sustava do neke točke na margini sivih točaka. Razlika tih dvaju vektora je vektor v_{ps} . Umnožak vektora v_{ps} i jediničnog vektora dao bi udaljenost između dviju margina. Kako bi dobili jedinični vektor možemo podijeliti vektor njegovom apsolutnom vrijednošću $\|w\|$.

Umnoškom vektora v_{ps} i jediničnog vektora dobiva se $\frac{2}{\|w\|}$. Kako bi maksimizirali udaljenost između margina, moramo minimizirati $\|w\|$. Kao funkciju minimizacije uzimamo $\frac{1}{2} \|w\|^2$ radi dobrih matematičkih svojstava. Za pronalazak minimuma uz ograničenja (1) možemo koristiti Lagrangeove multiplikatore.

Funkcija optimizacije se pretvara u

$$L = \frac{1}{2} \|w\|^2 - \sum_i \alpha_i [y_i (w \cdot x_i + w_0) - 1] \quad (2.2)$$

Za dobivanje ekstrema (2.2) potrebno je derivirati i izjednačiti s 0.

$$\begin{aligned} \frac{\partial L}{\partial w} &= w - \sum_i \alpha_i y_i x_i = 0 \\ w &= \sum_i \alpha_i y_i x_i \\ \frac{\partial L}{\partial w_0} &= w - \sum_i \alpha_i y_i = 0 \end{aligned} \quad (2.3)$$

$$\sum_i \alpha_i y_i = 0 \quad (2.4)$$

Uvrštavanjem (2.3) i (2.4) u (2.2) dobiva se sljedeće

$$L = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^T \cdot x_j \quad (2.5)$$

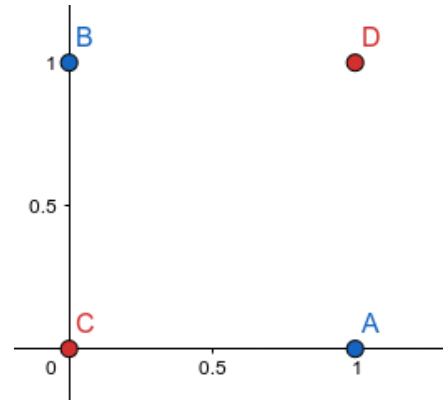
uz uvjet

$$\sum_i \alpha_i y_i = 0$$

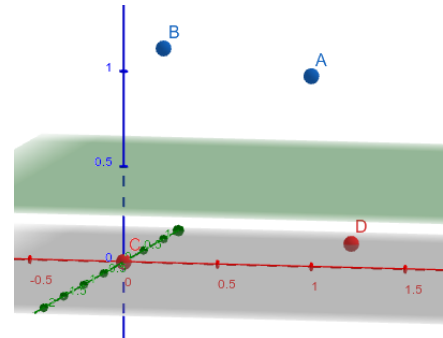
čime se dobiva krajnji izraz koji je potrebno maksimizirati po α nekim optimizacijskim algoritmom. Iz izraza je vidljivo da njegova maksimizacija ovisi o umnošku parova početnih uzoraka. Jednom kada se Lagrangeovi multiplikatori izračunati, optimalna hiperravnina se dobiva uvrštavanjem u (2.3). Prikazani postupak optimizacije vrijedi i daje dobre rezultate u slučaju kada su razredi linearno odvojivi, u slučaju kada nisu postupa se drugačijim pristupom problemu.

Slučaj kada nisu razredi linearno odvojivi prikazan je na slici 3. Koliko god se trudilil nikako nije moguće pravcem odijeliti dva razreda kako se ne bi događala kriva klasifikacija nekih budućih uzoraka. Ideja je prijeći u neki viši dimenzijski sustav u nadi da će razredi postati linearno odvojivi. Uzmimo kao primjer sliku 2.3 i prijedimo u trodimenzionalni sustav.

Novonastala situacija prikazana je na slici 2.4. U ovoj dimenziji granica koja dijeli razrede uzoraka postaje ravnina. Vidimo da je u prostoru s ovom dimenzionalnošću moguće odijeliti uzorke jedne od drugih tako da ne postoje preklapanja dvaju razreda koja bi rezultirala krivom klasifikacijom. Granica koja dijeli dva razreda prikazana je zelenkastom bojom na slici 2.4.



Slika 2.13: Linearno nerazdvojni razredi



Slika 2.14: Linearno nerazdvojni razredi u većoj dimenziji postaju linearno odvojivi

Funkcija kojom se prelazi u drugu dimenziju označava se s $\varphi(x)$. Kako bi se navedena ideja koristila u funkciji koja se maksimizira, dovoljno je samo napraviti sljedeće jer funkcija ovisi samo o polaznom skupu uzoraka, odnosno X .

$$L = \sum \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \varphi(x_i) \cdot \varphi(x_j)$$

Kraći zapis zapis korišten za $\varphi(x_i) \cdot \varphi(x_j)$ je $k(x_i, x_j)$ čime se označavaju jezgrene funkcije. Jezgrene funkcije služe za preslikavanje uzoraka iz početne, relativno niske dimenzionalnosti, u višu, moguće puno veću dimenzionalnost od izvorne kako bi uzorci u višoj dimenzionalnosti bili linearno odvojivi.

Tablica 2.1: Prikaz nekih popularnih jezgara

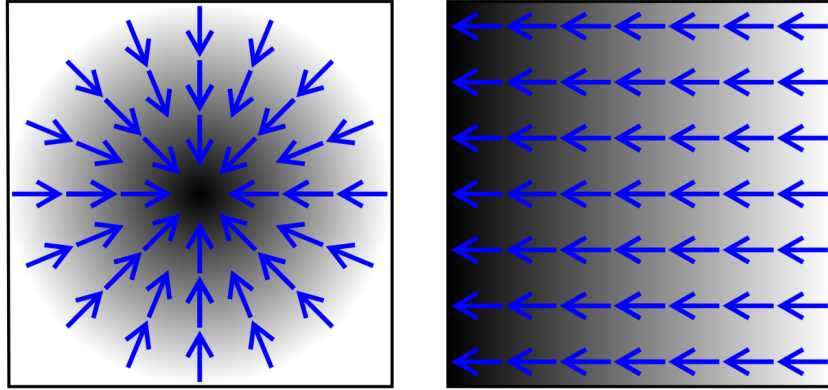
ime	formula
Linearna:	$K(x_i, x_j) = x_i^T x_j$
Polinomna:	$K(x_i, x_j) = (a \cdot x_i^T x_j + b)^d$
RBF:	$K(x_i, x_j) = e^{-\frac{\ x_i - x_j\ ^2}{\sigma^2}}$
Sigmoida:	$K(x_i, x_j) = \tanh(\sigma x_i^T x_j + r)$
Potencija:	$K(x_i, x_j) = -\ x_i - x_j\ ^\beta$
Logaritam:	$K(x_i, x_j) = -\log(1 + \ x_i - x_j\ ^\beta)$

Ova inačica SVM-a namijenjena je klasifikaciji gdje samo postoje 2 razreda. Međutim, jedan problem s više razreda može se podijeliti u više manjih binarnih problema klasifikacije. Popularne metode su:

- jedan protiv svih gdje se labela uzorku dodjeljuje prema klasifikatoru koji ima funkciju najveće vrijednosti
- jedan protiv drugog gdje se uzorak klasificira prema strategiji najvećeg broja pobjedi što znači da svaki od klasifikatora pridjeljuje pripadnost jednom od razreda i razred s najviše glasova pobjeđuje

2.0.5. Gradijent slike

Gradijent slike je promjena intenziteta ili boje u nekom smjeru na slici.



Slika 2.15: Prikaz smjera gradijenta plavim strelicama, tamnija boja označava veće vrijednosti.

Matematički, gradijent je funkcija dviju varijabli, vektor s dvije komponente koje su dobivene derivacijom u horizontalnom i vertikalnom smjeru. Na svakom pikselu vektor gradijenta je smjer najveće promjene intenziteta, a duljina toga vektora odgovara brzini promjene u tom smjeru.

$$\nabla f = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}, \text{ a iznos je dan s } \sqrt{g_x^2 + g_y^2}$$

Kako je funkcija intenziteta slike pozanta samo u diskretnim točkama, derivacija ove funkcije ne može biti definirana ako se ne pretpostavi da iza svega toga leži kontinuirana funkcija intenziteta koja je uzorkovana u dotičnim pikselima. Aproksimacije ovih funkcija derivacije mogu biti definirane s različitim stupnjevima točnosti. Jedan od popularnih načina za aproksimaciju gradijenta slike je konvolucija slike s nekom jezgrom, primjerice sa Sobelovim operatorom.

2.0.6. Sobelov operator

Ovaj operator koristi dvije 3×3 jezgre koju su konvolirane s izvornom slikom kako bi se aproksimirale derivacije – jedna za x smjer, a druga za y smjer. Označimo izvornu sliku kao dvodimenzionalnu matricu A , sliku koja je dobivena konvolucijom jezgre za x smjer i slike A kao G_x , a sliku koja je dobivena konvolucijom jezgre za y smjer i slike A s G_y .

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * A \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A$$

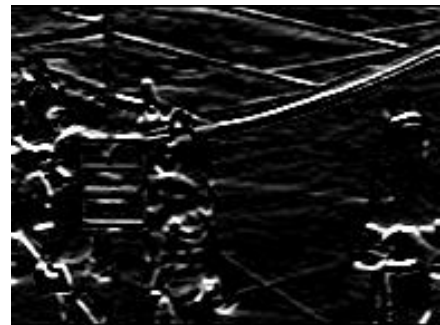
gdje $*$ označava operaciju konvolucije, a krajnja slika se dobije na sljedeći način

$$G = \sqrt{G_x^2 + G_y^2}$$

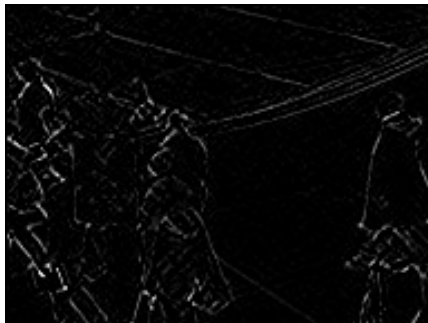
Primjer Sobelovog operatora



Slika 2.16: Izvorna slika



Slika 2.18: Prijenjen Sobelov operator u smjeru y na izvornoj slici



Slika 2.17: Prijenjen Sobelov operator na izvornoj slici



Slika 2.19: Prijenjen Sobelov operator u smjeru x na izvornoj slici

2.0.7. Haralickove značajke

Početna pretpostavka u karakterizaciji teksture slike je da matrica pojavnosti sivih tonova sadrži svu informaciju o teksturi slike. Samim time sve teksturne značajke će se izračunavati iz matrice pojavnosti sivih tonova. Funkcije koje definiraju skup od 14 mjera teksturnih značajki dane su u dodatku A. Neke od tih mjera se odnose na specifične teksturne karakteristike poput homogenosti, kontrasta i prisutnosti grupiranih ili usmjerenih struktura unutar slike. Druge mjere opisuju složenost i način promjene sivih tonova unutar slike.

Postoje neka intuitivna očekivanja u pogledu svojstava koja se odnose na neke značajke. Primjerice, bilo bi za očekivati da će vrijednost entropije poprimati veće vrijednosti za složenije slike. Možda bi također bilo moguće uočiti neke linearne zavisnosti u slikama koje imaju velike vrijednosti funkcije korelacije.

Utjecaj na vrijednosti Haralickovih značajki uvelike ima i veličina slike nad kojom se konstruira GLCM te iz koje se izračunavaju i same značajke. Ukoliko slika nije dovoljno velika, neće sadržavati dovoljno teksturnih informacija koje su zaslužne za razlikovanje različitih područja interesa dok s druge strane prevelika slika može sadržavati više različitih razreda ili regija koje pokušavamo odvojiti i razlikovati.

Iako te mjere sadrže informaciju o teksturnim značajkama slike, teško je identificirati koja je teksturna značajka karakterizirana nekom konkretnom funkcijom. Definicije funkcija iz dodatka A preuzete su iz izvornog rada Haralicka i drugih.⁶

⁶Haralick (1979)

3. Opis postupka

Predlaže se tehnika klizećeg prozora za klasifikaciju i lociranje područja mnoštva. Pripremne radnje obuhvaćaju pretvorbu izvornih slika, dimenzija 768 x 576, u sive tonove i dijeljenje njih samih u manje podslike, dimenzija 192 x 144, kojih ima 16 po svakoj izvornoj slici. Svaka podslika razlikuje se po gustoći mnoštva, pozadini i uvjetima osvjetljenja.



Slika 3.1: Prikaz pojedinih gustoća mnoštva

Sve se podslike ručno označe pripadnim razredima gustoće prema ¹ uz dotadnu oznaku „no flow” koja predstavlja podslike u kojima nema nikakvog mnoštva. Sljedeće se primjenjuje operator LBP nad svakom podslikom i pritom je potrebno napomenuti da postoje dvije vrste izračuna LBP operatora. Jedan se izračunava nad slikama sive razine, a drugi nad slikama koje su također sive razine, ali je nad njima primjenjen operator gradijenta. Nakon dobivanje slike lokalnih binarnih značajki, koristeći se tehnikom kliznog prozora, konstruira se matrica pojavnosti lokalnih binarnih značajki (LBPCM) nad svakim oknom podslike. Iz svake LBPCM izračunavaju se željene Haralickove značajke i te se vrijednosti pohranjuju u vektor značajki. Vektor značajki se sastoji od vrijednosti Haralickovih značajki svakog okna, odnosno vektor značajki neke podslike sastoji se od vrijednosti Haralickovih značajki njezinih okana. Po završetku stvaranja, vektori značajki, se predaju k -NN klasifikatoru ili SVM-u.

¹Polus et al. (1983)

Na kraju se pristupa evaluaciji samih klasifikatora i eventualnoj izmjeni samih parametara u nadi boljeg rezultata u sljedećoj iteraciji učenja. Za učenje klasifikatora koriste se 70% izvornih slika, a za testiranje ostatak.

Prethodno opisan postupak moguće je vidjeti na slici 3.2. gdje je prikazana arhitektura sustava i pojedine faze koje su kasnije u poglavlju 4 detaljnije opisane. Prva faza obuhvaća samu pripremu podslika koje se koriste za stvaranje vektora značajki. Nakon prve faze događa se grananje u dva dijela: prvi dio obuhvaća klasifikatore koji koriste slike sivih razina nad kojima se primjenjuje operator LBP, dok drugi dio predstavlja klasifikatore koji koriste slike sivih razina nad kojima se primjenjuje operator gradijenta i tek nakon toga operator LBP. Nakon što su dobiveni vektori značajki svakog, oni se prosljeđuju odgovarajućem klasifikatoru kako bi se izvršila ocjena njihove točnosti, odnosno validacija na testnom skupu podataka. Jednom kada su ocijenjeni klasifikatori moguće je napraviti prividnu fuziju u jedan klasifikator.

Sjedinjenje dvaju različitih klasifikatora obavlja se pomoću nove vrste klasifikatora - glasačkog klasifikatora (*engl. voting classifier*). Poželjno je da se koristi jedan koji koristi slike sivih razina, a jedan slike nad kojima je primijenjen operator gradijenta kako bi tako stvoren klasifikator bio robusniji te dao potencijalno bolji rezultat. Ovakva vrsta klasifikatora pogodna je zbog načina na koji daje važnost pojedinim klasifikatorima od kojih se sastoji.

Na odluku klasifikatora može se utjecati na više načina. Prvi je „tvrdo glasanje” (većinsko glasanje), gdje svaki od klasifikator glasa za neku labelu, a najbrojnija labela pobjeđuje i dodjeljuje se novome uzorku koji se klasificira. Ovakvo glasanje je moguće i modificirati na način da se svakome klasifikatoru dodijeli neka težina, npr. ovisno o točnosti njegove klasifikacije na setu slika za učenje.

Drugi način je „meko glasanje” (*engl. soft voting*) gdje se dohvate vjerojatnosti klasifikacije svake labele svakog od klasifikatora, uprosječe se, pa se na temelju tih novih vjerojatnosnih vrijednosti određuje pripadnost dosad neviđenog uzorka. Pošto su poznate vjerojatnosti klasifikacije moguće je množiti svaku vjerojatnost nekom težinom i time utjecati na ishod odluke.

Odabrana je druga vrsta glasanja, tzv. „meko glasanje” jer je eksperimentom utvrđeno da daje bolje rezultate nad skupom podataka koji se koristi u radu.

Primjer dobivanja vjerojatnosti glasačkog klasifikatora

$$y = \arg \max_i \sum_{j=1}^m w_j p_{ij}$$

gdje w_j predstavlja težinu pridijeljenu j -tom klasifikatoru.

Uzmimo 3 klasifikatora i dva moguća razreda pripadnosti svakog uzorka

$$C_1(x) \rightarrow [0.9, 0.1],$$

$$C_2(x) \rightarrow [0.8, 0.2],$$

$$C_3(x) \rightarrow [0.4, 0.6]$$

Koristeći unofomne težine dobivamo sljedeće srednje vjerojatnosti

$$p(i_0|x) = \frac{0.9+0.8+0.4}{3} = 0.7$$

$$p(i_1|x) = \frac{0.1+0.2+0.6}{3} = 0.3$$

$$y = \arg \max_i [p(i_0|x), p(i_1|x)] = 0$$

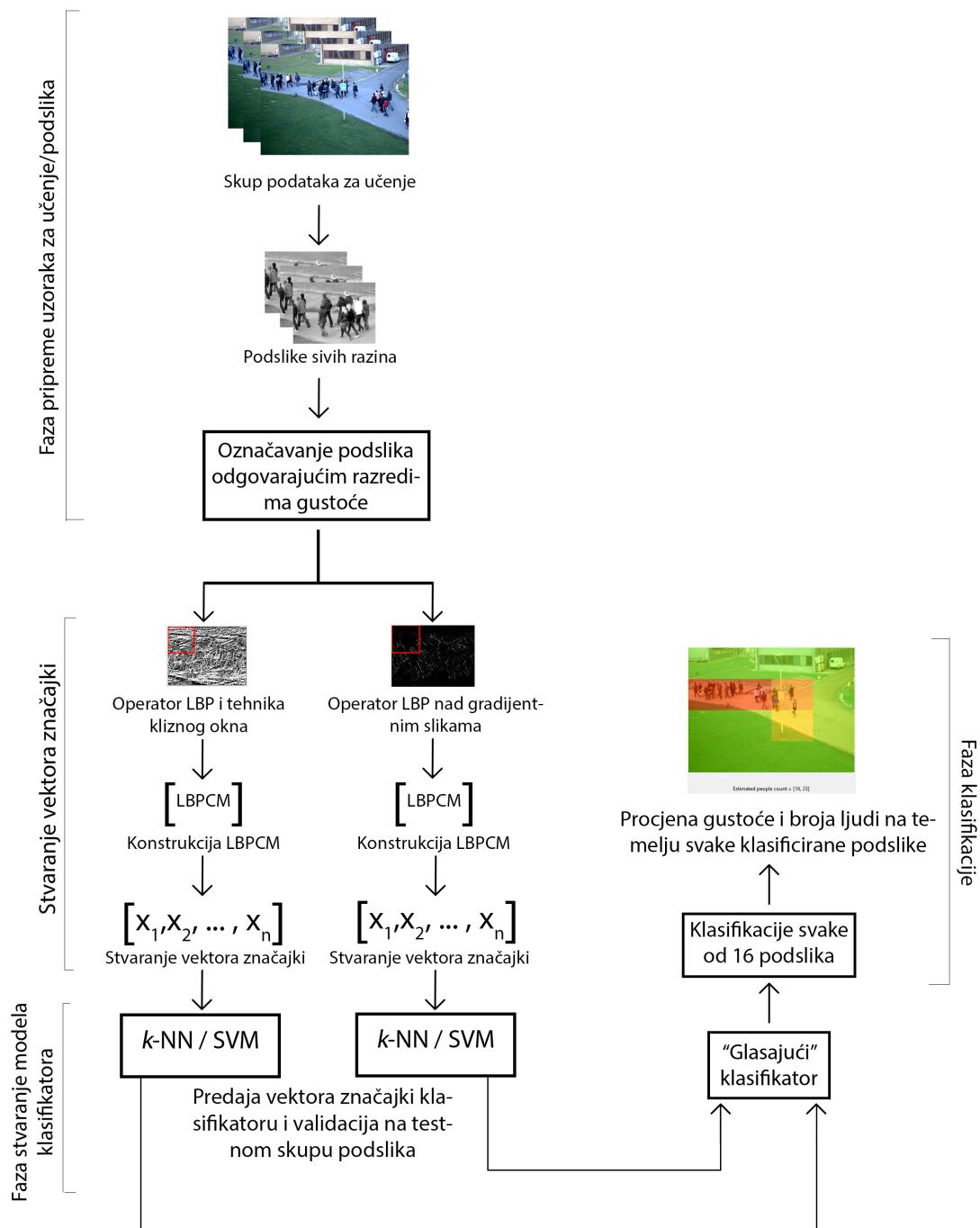
Promjenimo iznose težina svakog klasifikatora sljedećim $\{0.1, 0.1, 0.8\}$. Sada izračun vjerojatnosti izgleda sljedeće

$$p(i_0|x) = 0.1 \cdot 0.9 + 0.1 \cdot 0.8 + 0.8 \cdot 0.4 = 0.49$$

$$p(i_1|x) = 0.1 \cdot 0.1 + 0.2 \cdot 0.1 + 0.8 \cdot 0.6 = 0.51$$

$$y = \arg \max_i [p(i_0|x), p(i_1|x)] = 1$$

Iz primjer je jasno vidljivo koliku važnost imaju pridijeljene težine svakom od klasifikatora i ukoliko nisu ispravno dodijeljene, rezultat klasifikacije može biti značajno lošiji od svakog pojedinog klasifikatora.

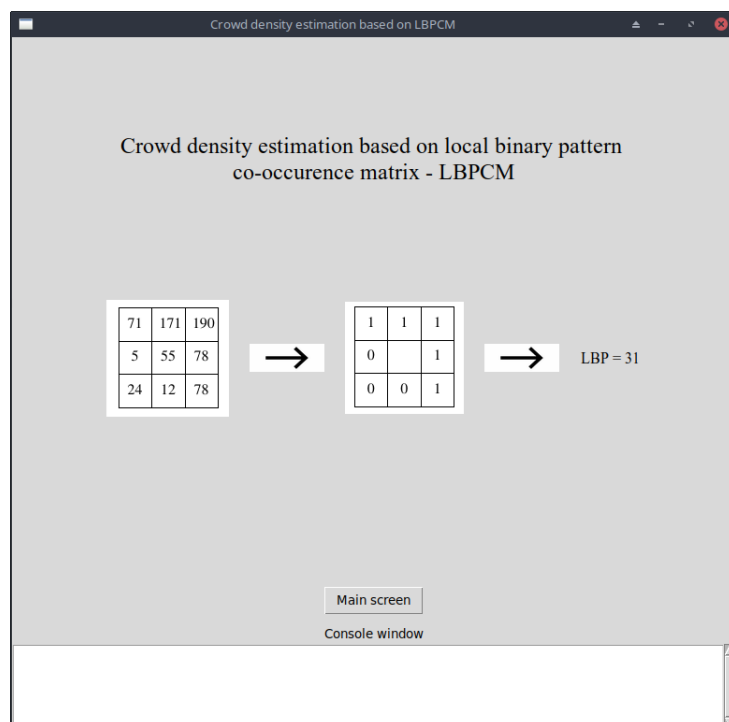


Slika 3.2: Prikaz arhitekture sustava

4. 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: OpenCV¹, scikit-learn² i PIL³.

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.



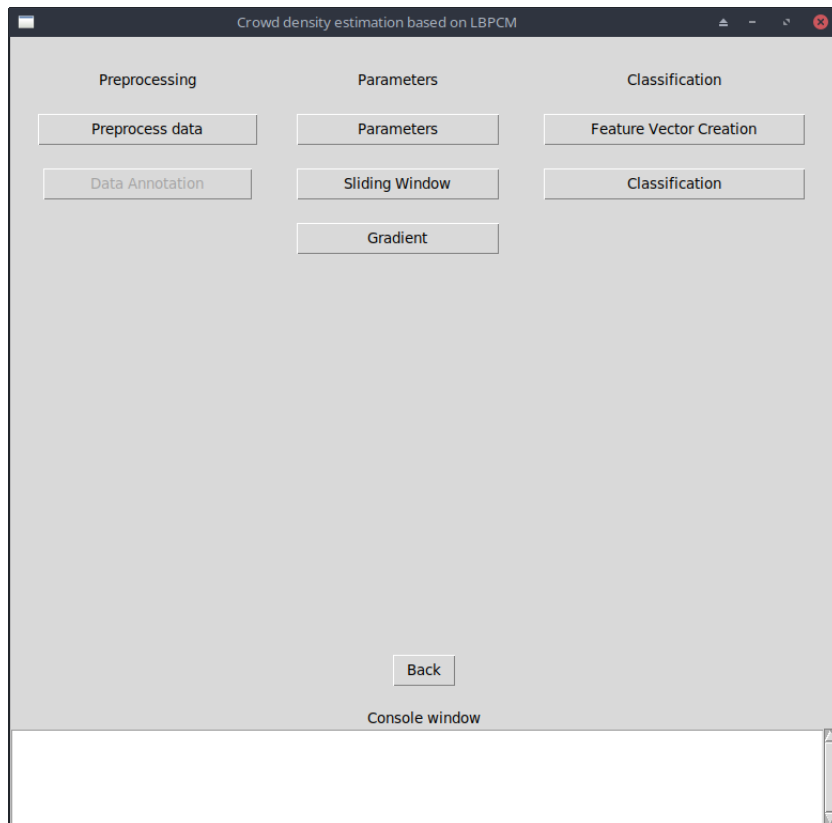
Slika 4.1: Prikaz početnog prozora nakon pokretanja aplikacije

¹Intel (2000)

²?

³?

Početni prozor prikazuje naziv rada na kojem se temelji ovaj završni rad, slikovni prikaz postupka lokalnih binarnih značajki i gumb „Main screen” koji otvara stranicu s glavnim funkcionalnostima aplikacije. U podnožju svakog prozora nalazi se prazan prozor koji „Console window” koji ima ulogu mjesta u koji se ispisuju važnije informacije poput upozorenja, eventualnih grešaka i rezultata izvođenja nekih drugih funkcija. Pritiskom na gumb „Main screen” otvara se glavni prozor koji sadrži sve funkcionalnosti aplikacije.



Slika 4.2: Prikaz glavnog prozora

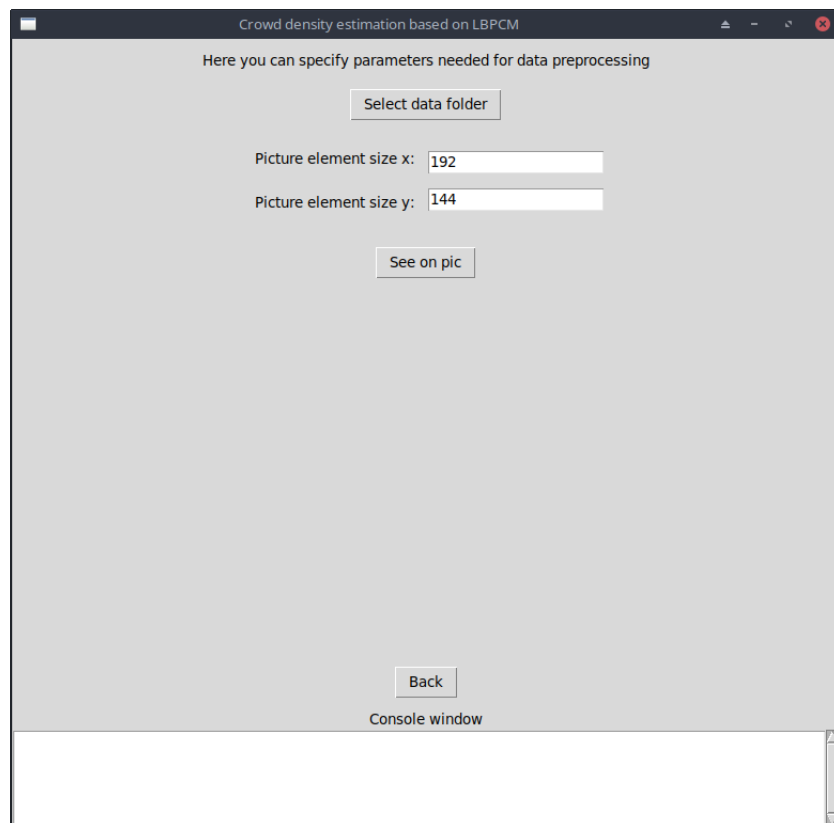
Prostor glavnog prozora podijeljen je u tri dijela. Prvi dio se sastoji od radnja koje je potrebno obaviti prije samog postupka učenja, a to su: priprema izvornih slika podjelom u manje podslike i označavanje svake podslike odgovarajućim razredom gustoće mnoštva. Po završetku navedenih radnja moguće je nastaviti s daljnjim postupkom.

Drugi dio prozora služi za promjenu parametara aplikacije, uvid u tehniku klizećeg okna gdje je moguće vidjeti kako se mijenjaju iznosi Haralickovih značajki prilikom putovanja klizećeg okna po pojedinoj slici i na kraju korištenje operatora gradijenta na odabranoj slici.

Zadnji dio prozora obuhvaća samu fazu stvaranja vektora značajki (učenje klasifikatora) gdje se postavljaju svi parametri pojedinog modela i pokreće postupak učenja i drugi dio koji je zaslužan za postupak klasifikacije dosad neviđenih uzoraka nekim od već naučenih modela.

4.0.1. Podjela slika u podslike

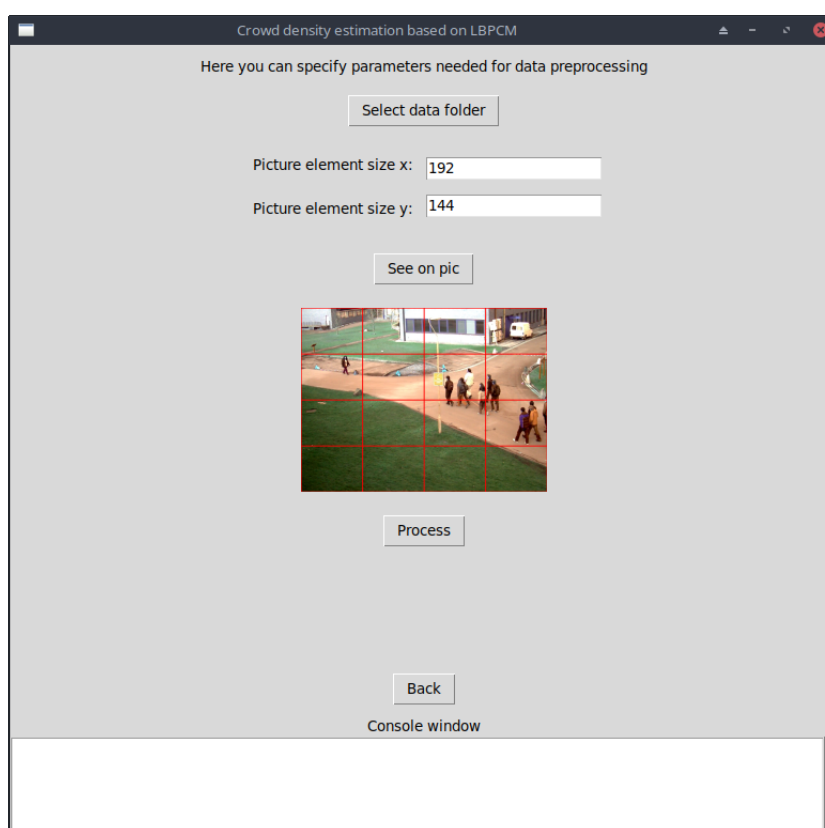
Prije svega potrebno je pripremiti skup slika koji se koristi za učenje i kasnije za verifikaciju klasifikatora. Izvorni skup se sastoji od 221 slike koje su izvučene iz video zapisa jedne nadzorne kamere. Svaka slika je dimenzija 768 x 576 te se podijeli u 16 jednakih podslika 192 x 144. Ova konkretna dimenzija podslika je odabrana kako ne bi ostalo neiskorištenih piksela na rubovima i kako dimenzije novonastalih podslika ne bi previše odskakala od veličine mnoštva koje je prisutno u izvornim slikama. Nakon što su slike podijeljene na manje dijelove, novonastalih podslika ima 3536 koje se zatim pretvaraju u slike sivih razina kako bi se prilikom računanja Haralickovih značajki tijekom učenja klasifikatora taj korak mogao preskočiti pa bi se uštedjelo vrijeme.



Slika 4.3: Prikaz stranice za obradu izvornih slika

Aplikacija nudi potporu za izvođenje navedenog postupka. Kako bi se izvorne slike podijelile u manje podslike potrebno je pritisnuti na gumb „Preprocess data” koji otvara stranicu izgleda na slici 4.3.

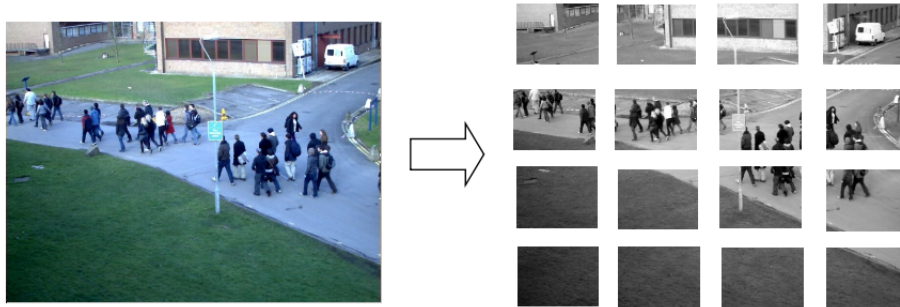
Prozor na samom vrhu ima gumb za odabir direktorija koji sadrži izvorne slike za učenje i testiranje. Po odabiru navedenog direktorija mogu se upisati željene dimenzije podslike koje će nastati dijeljenjem izvorne slike. Prvi element unosa sadrži veličinu $s \times x$ smjeru, a drugi u y smjeru. Nakon unešenih dimenzija moguće je vidjeti kako bi izgledale konkretne dimenzije na izvornoj slici pritiskom na gumb „See on pic”. Nakon pritiska na gumb pojavljuje se sljedeći dio prozora.



Slika 4.4: Prikaz izgleda podslike s nešenih dimenzijama

Novi dio prozora prikazuje jednu izvornu sliku iz odabranog direktorija s podjelom na manje podslike koje su dimenzije unešene iznad. Svaka podslika prikazana je crvenim linijama i ako nismo zadovoljni njihovom dimenzijom moguće je upisati neke druge vrijednosti i vidjeti novi rezultat. Ukoliko nam odgovara veličina podslika potrebno je pritisnuti gumb „Preprocess” koji pokreće postupak dijeljenja slika iz odabranog direktorija i zapisuje ih u direktorij „preprocessedData”. Taj direktorij nakon završenog postupka sadrži 3536 podslika u sivim razinama, ukoliko su korištene

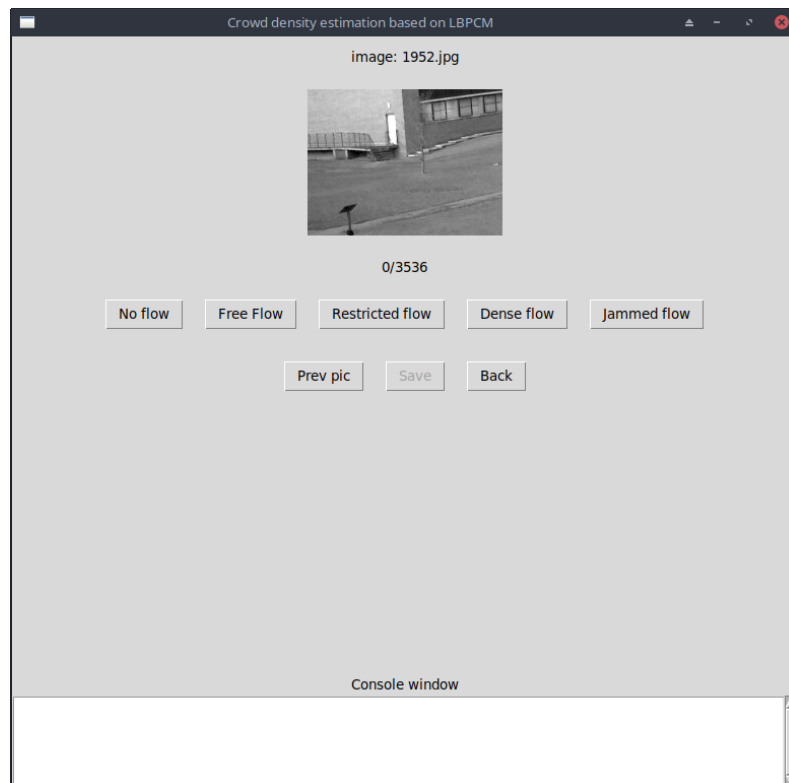
dienzije 192 x 144, pa je moguće prijeći u sljedeću fazu, a to je označavanje svake pojedine podslike.



Slika 4.5: Prikaz podijeljene izvorne slika na manje podslike

4.0.2. Označavanje podslika

Po završetku dijeljenja izvornih slika u manje podslike pristupa se postupku označavanja podslika. Kako bi mogli označiti svaku pojedinu podsliku na što jednostavniji način potrebno je pritisnuti gumb „Data Annotation” u glavnom prozoru aplikacije. Nakon pritiska na gumb otvara se sljedeći prozor.



Slika 4.6: Prikaz prozora za označavanje podslika

Najgornji dio prozora prikazuje ime trenutne podslike koju je potrebno označiti. Ispod naziva se nalazi konkretna slika koja se označava, a i prisutan je brojač koji prikazuje broj označenih slika. Ispod brojača postoji 5 gumba koji označavaju razine gustoće mnoštva. Podslika se označava pritiskom na jedan od gumba koji odgovara gustoći mnoštva zatečenoj na trenutnoj slici. Nakon pritiska na gumb pojavljuje se sljedeća od podslika koju je potrebno označiti, a labela prethodne podslike se zapisuje u interni spremnik.

Nakon što su sve podslike označene odgovarajućim labelama, interni spremnik je moguće spremiti na disk u obliku tekstualne datoteke pritiskom na gumb „Save”. Nakon pritiska na gumb „Save” stvara se tekstualna datoteka imena „labeledData.txt”, a svaki redak je oblika 0.jpg:2, gdje 0.jpg predstavlja ime podslike koje je jedinstveno, a 2 označava pripadni razred gustoće mnoštva. Ukoliko je došlo do pogreške prilikom označavanja neke podslike moguće je pritisnuti gumb „Prev pic” čime se vraća prethodna slika kojoj je moguće pridijeliti drugačiju labelu. Potrebno je napomenuti da se jednom označene podslike ne trebaju više označavati prilikom svakog procesa učenja nekog klasifikatora nego je samo potrebno učitati postojeću datoteku.

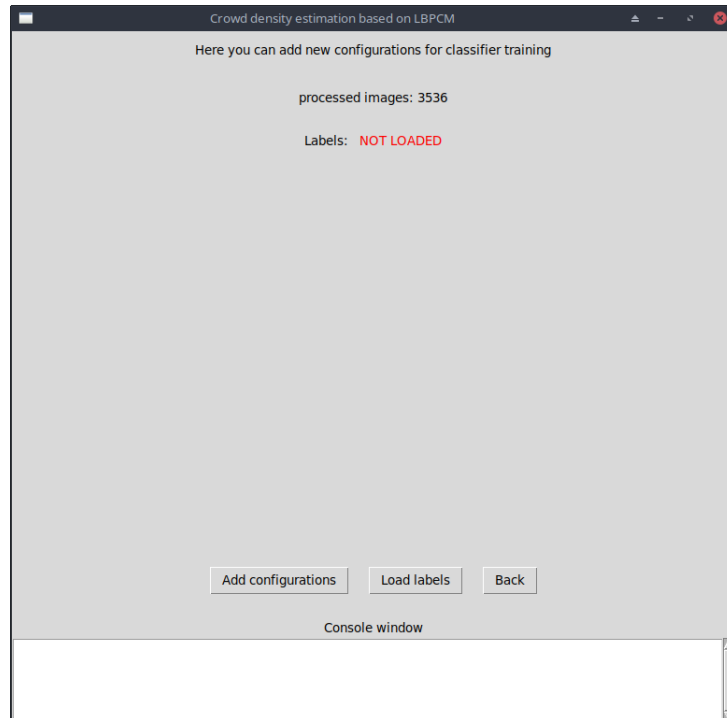
4.0.3. Stvaranje vektora značajki

Jednom kada se slike podijeljene u manje podslike i nakon završetka označavanja svih podslika, moguće je pristupiti postupku stvaranja vektora značajki. Kako bi pristupili tome prozoru, potrebno je pritisnuti gumb „Feature Vector Creation” nakon čega se otvara prozor prikazan na slici 4.7.

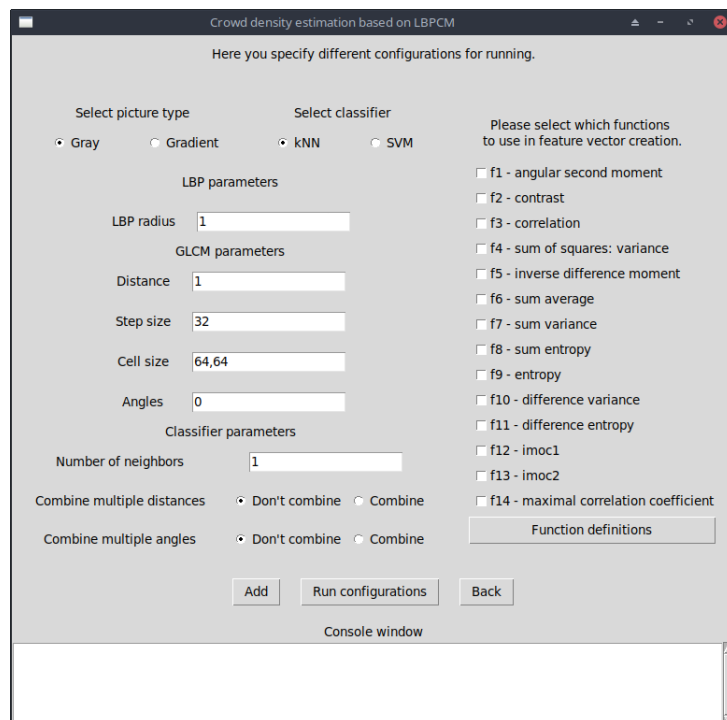
Prozor ne prikazuje puno osim broja podslika i informacije jesu li labela podslika učitane u memoriju ili nisu. Kako bi ih učitati potrebno je pritisnuti na gumb „Load labels” nakon čega se otvara prozor za odabir tekstualne datoteke s labelama. Ako su labela uspješno učitane, mijenja se tekst ispod broja podslika u tekst zelene boje „LOADED”. Podnožje prozora još sadrži gumb za povratak na prethodnu stranicu i gumb za dodavanje novih konfiguracije koji će biti ubrzo objašnjen.

Središnji dio prozora trenutno ne prikazuje ništa jer nije dodana niti jedna konfiguracija. Jednom kada se doda nova konfiguracija, u središnjem dijelu prozora stvori se traka napretka koja prikazuje koliko još ima do završetka te pojedine konfiguracije.

Stvaranju vektora značajki prethodi stvaranje neke konfiguracije. Kako bi se stvorila nova konfiguracija potrebno je pritisnuti na gumb „Add configurations”. Nakon pritiska na gumb otvara se prozor prikazan na slici 4.8.



Slika 4.7: Početni prozor za stvaranje vektora značajki



Slika 4.8: Izgled prozora za stvaranje konfiguracije

Prozor za stvaranje konfiguracije sastoji se od lijevog i desnog dijela. Lijevi dio služi za unos parametara dok je u desnom dijelu moguće odabrati Haralickove značajke koje se izračunavaju iz GLCM. Na vrhu prozora potrebno je odabrati vrstu slike nad

kojom se primjenjuje LBP operator: sliku sivih razina (*gray*) ili sliku sivih razina nad kojom je primijenjen operator gradijenta (*gradient*).

Za odabir klasifikatora postoje dvije mogućnosti: k -NN klasifikator i SVM. Odabranom klasifikatoru se prosljeđuju stvoreni vektori značajki gdje se onda interno sprema u slučaju k -NN ili upotrijebe u fazi učenja klasifikatora u slučaju SVM.

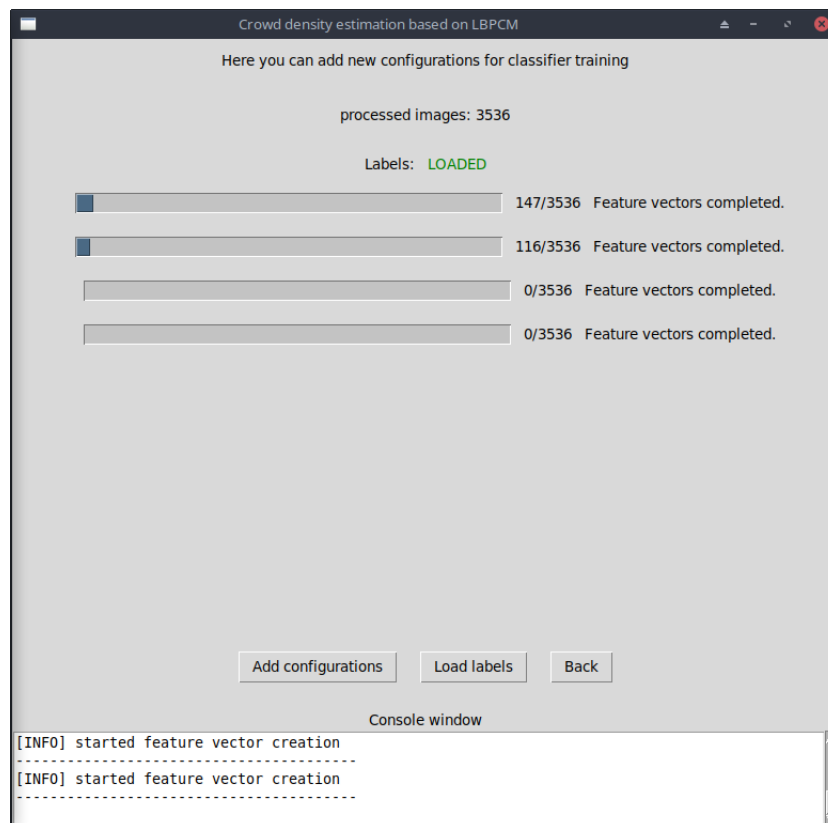
Desna strana prozora prikazuje svih 14 Haralickovih značajki. Jednom kada su odabrani svi parametri s lijeve strane, desna strana se koristi za odabir značajki koje se izračunavaju prilikom kreiranja vektora značajki svake pojedine konfiguracije. Svaka konfiguracija može imati broj značajki veći ili jednak jedan. Ovisno o broju odabranih značajki i broju različitih udaljenosti i kutova, vrijeme stvaranja vektora značajki može biti značajno dugačko stoga se u fazi eksperimentiranja stvara više konfiguracija koje se mogu izvoditi istovremeno ukoliko je računalo višeprosorsko. Definicije Haralickovih značajki moguće je vidjeti u Dodatku A ili pritiskom na gumb „Function definitions” nakon čega se otvara novi prozor u kojem je naveden algebarski oblik svake Haralickove značajke kao i opisi drugih oznaka korištenih u formulama.

Nakon odabira željenih Haralickovih značajki moguće je pritisnuti na gumb „Add” čime se stvara nova konfiguracija. Ukoliko želimo dodati još konfiguracija, potrebno je samo označiti ili izmijeniti postojeće parametre i opet pritisnuti gumb „Add”. Svakim pritiskom gumba stvara se nova konfiguracija i interno pohranjuje te čeka na pokretanje postupka stvaranja vektora značajki. Kada smo dodali sve konfiguracije koje želimo, za početak stvaranje vektora značajki potrebno je pritisnuti gumb „Run configurations” čime se onoliko konfiguracije izvodi paralelno koliko naše računalo raspoloživih procesora ima.

Za praćenje napretka pokrenutih konfiguracija može se pritisnuti gumb „Back” koji otvara prethodnu stranicu sa slike 4.7. samo što su sada u središnjem dijelu stranice trake napretka svake od dodanih konfiguracije od maloprije. Kraj svake konfiguracije je brojač koji prati broj do sada obrađenih, odnosno stvorenih vektora značajki. Dotična stranica prikazana je na slici 4.9., a napreduju samo dvije konfiguracije jer računalo ima samo dva raspoloživa procesora. Po završetku jedne od prve dvije se pokreće treća i tako sve do završetka svih dodanih konfiguracija. Nakon završetka svake konfiguracije objekt koji predstavlja model klasifikatora sprema se na disk kako i konfiguracija samog modela kako bi se kasnije prilikom faze klasifikacije mogao učitati ispravan model.

Parametar	Opis
LBP radius	označava veličinu radijusa oko središnjeg piksela na temelju kojeg se izračunava LBP neke slike
Distance	predstavlja udaljenost u GLCM na temelju koje se gledaju pojavnosti sivih razina. Moguće je istovremeno izračunavati vrijednosti za više udaljenosti, a u tom slučaju ih je potrebno odvojiti zarezom
Step size	veličina koraka u x i y smjeru prilikom tehnike klizećeg okna
Cell size	veličina putujućeg okna u smjeru x, y
Angles	su svi kutovi za koje se konstruira GLCM, odvajaju se zarezom i upisuju u stupnjevima
Number of neighbors	koristi se kod k -NN klasifikatora gdje označava broj susjeda na temelju čijeg se glasanja daje labela novome uzorku
Combine multiple distances	predstavlja opciju kombiniranja više udaljenosti u jednu, npr. ako je GLCM izračunavana za udaljenosti 1, 2 i 3, matrice se zbroje i podijele s 3 ako bi se odabrao slučaj s kombinacijom udaljenosti „combine” u suprotnom se svaka interpretira za sebe
Combine multiple angles	jednako kao i za udaljenosti

Tablica 4.1: Opis parametara konfiguracije



Slika 4.9: Prikaz napretka pojedinih konfiguracija

4.0.4. Klasifikacija slike postojećim modelima

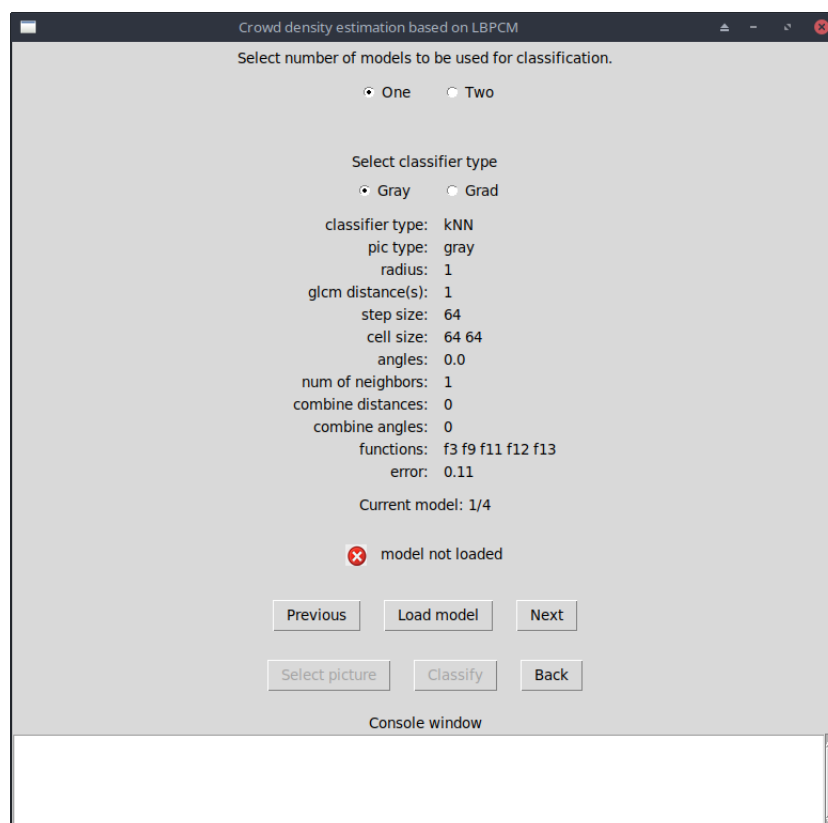
Jednom kada je stvaranje vektora značajki završeno i kada je model spremljen moguće je pristupiti postupku klasifikacije dosad neviđene slike.

Kako bi se pristupilo stranici za klasifikaciju potrebno je pritisnuti gumb „Classification” koji se nalazi na glavnoj stranici aplikacije. Izgled stranice za klasifikaciju prikazan je na slici 4.10.

Na stranici je prikazana konfiguracija nekog modela koji se nalazi na disku računala. Prikazane su sve komponente konkretne konfiguracije trenutnog modela, a uz to je i prikazana greška klasifikacije pod nazivom „error”. Ta greška je omjer točno klasificiranih podslika i ukupnog broja podslika koji se koristi za testiranje klasifikatora na podslikama koje nije vidio tijekom faze učenja. Ukoliko nije niti jedan model klasifikatora dostupan, prikazana je labela „NO MODEL” koja javlja korisniku kako nije moguće klasificirati slike pošto nikakvog modela nema.

Trenutno je prikazan model koji je učen na slikama sive razine i moguće je vidjeti da trenutno takvih modela na disku ima 4 (redni broj modela „Current model”). Kako

bi vidjeli moguće modele koji su učeni na gradijentnim slikama potrebno je pritisnuti gumb „Grad”. Za dohvat sljedećeg ili prethodnog dostupnog modela potrebno je pritisnuti na gumbe „Next” ili „Previous”. Jednom kada je odabran željeni model za klasifikaciju potrebno je pritisnuti gumb „Load model” čime se prikazani model učitava u memoriju i moguće ga je koristiti za klasifikaciju. Ako je model ispravno učitano, tekst ispod trenutnog modela se mijenja u „model loaded” sa zelenom kvačicom.



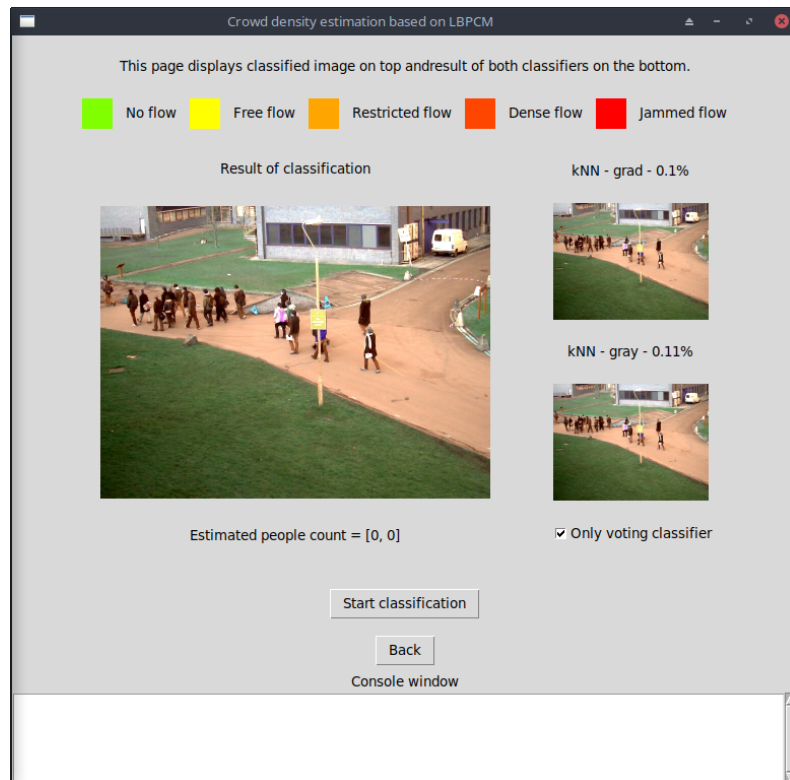
Slika 4.10: Prikaz stranice za odabir jednog modela

Opisani postupak se primjenjuje kada želimo klasificirati novu sliku koristeći samo jedan model. U slušaju da želimo koristiti kombinaciju dva modela potrebno je pritisnuti gumb „Two” na vrhu stranice. Nakon pritiska na gumb i učitavanja oba modela izgled stranice je prikazan na slici 4.11. Jednom učitane modele moguće je koristiti za klasifikaciju. Kako bi se odabrala željena slika za klasifikaciju potrebno je pritisnuti gumb „Select picture” nakon čega se otvara prozor za odabir slike. Ako slika nije učitana, korisniku se ispisuje prikladna poruka te nije moguće nastaviti na klasifikaciju sve do kada se ne učitava neka slika. Nakon jednom učitane slike potrebno je pritisnuti na gumb „Classify” koji otvara prozor prikazan na slici 4.12.



Slika 4.11: Prikaz stranice za odabir dva modela

Na slici 4.12. prikazane su 3 identične kopije odabrane slike sljedećeg značenja. Na mjestu najveće slike će se prikazati rezultat kombinacije obaju modela dok su slike s desne strane rezultati klasifikacije svakog pojedinog klasifikatora. Razlog za to je uvid u utjecaj pojedinih težina svakog klasifikatora na krajnji rezultat klasifikacije. Vrh stranice prikazuje paletu boja koje su dodijeljene pojedinom razredu gustoće mnoštva radi vizualnog prikaza rezultata klasifikacije. Klasifikacija se napokon može pokrenuti pritiskom na gumb „Start classification” nakon čega je potrebno pričekati neko vrijeme koje ovisi o složenosti učitano modela. Ovdje je moguće zamijetiti koliki utjecaj imaju različiti parametri i broj parametara na performanse programa i koje značajke se „isplati” koristiti za stvaranje vektora značajki. Neke od značajki su izuzetno skupe za izračunavati i njihov doprinos možda i nije znatan u odnosu na onu drugu koja se brže izračunava stoga je potrebno naći nekakav kompromis između procesorskog vremena i točnosti klasifikatora jer nikome nije u cilju čekati neko duže vrijeme na rezultat klasifikacije pogotovo ukoliko se radi o nekakvoj video sekenci. Podno najveće slike se nakon gotove klasifikacije prikazuje interval procjene broja ljudi. Ako nije potrebno vidjeti rezultate svakog od klasifikatora moguće je označiti „Only voting classifier” čime se samo klasificira najveća slika. Jednom kada je klasifikacija završila moguće je vidjeti rezultat na slici 4.13.



Slika 4.12: Prikaz stranice za klasifikaciju slika



Slika 4.13: Prikaz klasificirane slike i procjene broja ljudi na temelju gustoće mnoštva

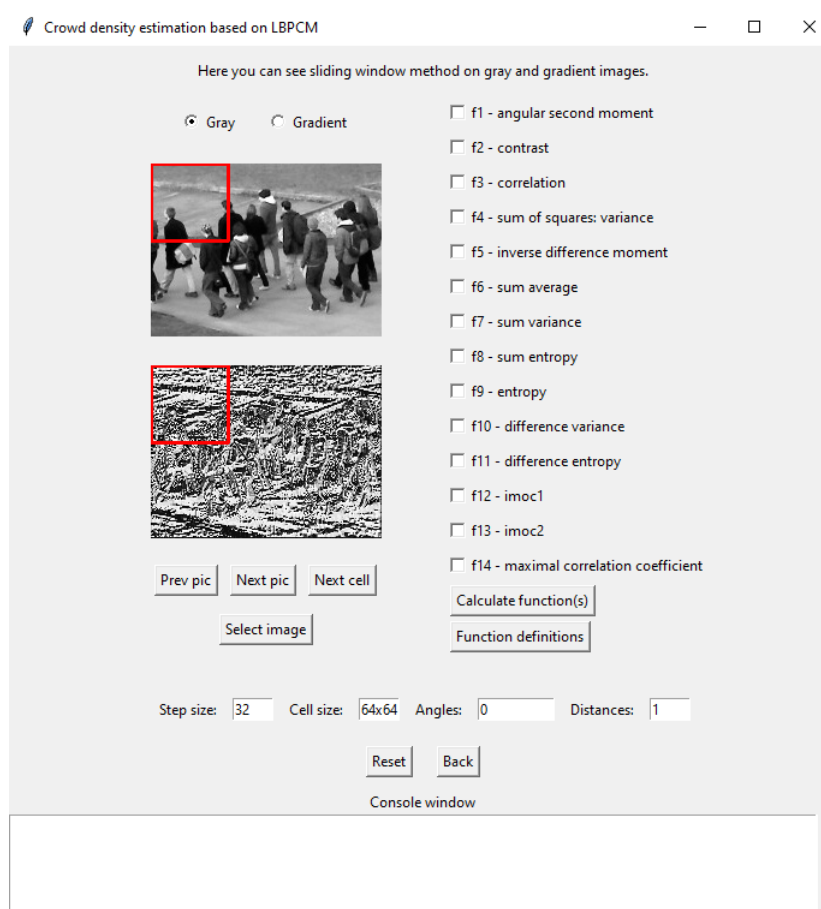
Na slikama koje pokazuju rezultate klasifikacije pojedinih klasifikatora s desne strane može se zamijetiti kako točnosti klasifikacije utječu na krajnji rezultat. Klasifikatori iz primjera nisu najbolje točnosti, međutim kada su upareni zajedno, mogu dati bolje rezultate kako je i ilustrirano na slici 4.13. To je i cijela ideja korištenja skupnog klasifikatora koji spaja više njih u jednu cjelinu, u ovom slučaju dva.

Iako se čini iz navedenog primjera da je rezultat samo preslika klasifikatora koji ima veću točnost klasifikacije, to ne mora uvijek biti slučaj. Neka može fokus jednog klasifikator biti usmjeren na određene dijelove slike koji je možda uvjetovan odabranim Haralickovim značajkama, uvjetima osvjetljenja, kutom nagiba kamere ili nečim drugim dok drugi klasifikator može imati fokus na sasvim drugi dio slike.

Opisani problem bi se manifestirao na način da su neka područja slike, koja su u fokusu klasifikatora, ispravno klasificirana dok druga nisu te jednako tako i za drugi klasifikator, ali kada su „spojena” zajedno, daju jedan rezultat koji može davati veoma dobre rezultate. Pozitivne strane jednog klasifikatora mogu nadjačati negativne strane drugog klasifikatora i na kraju rezultirati uspješnom procjenom gustoće mnoštva. Ta simbioza klasifikatora uvelike je određena težinama koje množe pojedine vjerojatnosne vrijednosti svakog klasifikatora.

5. Tehnika klizećeg okna

Kao jednu od osnovnih alata postupka prikazanog u ovom radu, tehniku kliznog okna moguće je vidjeti pritiskom na gumb „Sliding window” na glavnoj stranici aplikacije.



Slika 5.1: Prikaz prozora tehnike klizećeg okna

Na slici 5.1. prikazane su dvije podslike koje su izravno učitane iz direktorija s već obrađenim podslikama za ilustraciju postupka. Ukoliko je prikazan tekst „NO IMAGE” na mjestu namijenjenom za slike, to znači da još nisu izvorne slike obrađene i da podslike još ne postoje pa je potrebno učitati sliku iz nekog drugog direktorija pritiskom na gumb „Select Image” nakon čega se otvara prozor za odabir željene podslike.

Na vrhu prozora postoje mogućnosti prikaza postupka nad podslikama sive razine i nad slikama koje su također sive razine, ali se nad njima još dodatno primijeni operator gradijenta. Za promjenu iz jednog stanja u drugo potrebno je samo pritisnuti na željeni gumb. Za pomicanje okna na sljedeću poziciju zaslužan je gumb „Next cell”. Pritiskom na dotični gumb se okno, koje je na slici prikazano crvenim rubovima, pomiče na sljedeću poziciju. Sljedeća slika iz direktorija obrađenih podslika dohvaća se pritiskom na gumb „Next pic”, a prethodna gumbom „Prev pic”. Podnožje prozora zaokupljaju područja za upis parametara vezanih uz LBP i GLCM. Pretpostavljene vrijednosti su prikazane ulaskom u prozor i ukoliko se želi vidjeti utjecaj drugačijih parametara, potrebno je upisati novu vrijednost parametra u odgovarajuće područje za upis i pritisnuti na gumb „Reset” čime se ažurira podslika i resetira pozicija klizećeg okna.

Želimo li izračunati vrijednosti konkretnih Haralickovih značajki trenutnog okna potrebno je odabrati sve željene značajke u desnom dijelu prozora te pritisnuti gumb „Calculate function(s)”. Ukoliko nije jasno kako su neke značajke definirane, moguće je vidjeti njihov algebarski zapis pritiskom na gumb „Function definitions”. Nakon pritiska na gumb za izračun Haralickovih značajki, njihove vrijednosti se pojavljuju u konzoli aplikacije. Izračun Haralickovih značajki se ovdje ne izračunava svakim pomakom okna jer su neke značajke iznimno računski zahtjevne, a i možda nam neka područja podslike nisu zanimljiva pa se značajke samo izračunavaju pritiskom na odgovarajući gumb.

Pogledom u izračune različitih Haralickovih značajki može se zamijetiti da vrijednosti funkcija odskaku u nekoliko redova veličina jedne od drugih, stoga ih je potrebno normalizirati prije nego što se vektori značajki predaju klasifikatoru. Normalizacija je nužan postupak iz razloga što ovdje korišteni klasifikatori koriste mjeru udaljenosti kao funkciju koja određuje pripadnost nekom razredu te ako bi ostavili vrijednosti ovakvima, neke od njih uopće ne bi imale ikakvog utjecaja zbog svoje veličine dok bi druge, koje su možda manje važne u nekim slučajevima, dominirale i previše utjecale na krajnji rezultat klasifikacije. Postoje više različitih metoda kojom se vektori značajki normaliziraju, npr. min-max, oko srednje vrijednosti, normalizacija z-ocjenom, skaliranje na jedinični vektor i dr.

U ovom radu je odabrana normalizacije kojom se vektori značajki svode na srednju vrijednost oko 0 i jediničnu varijancu.

Postupak normalizacije:

N - ukupan broj vektora značajki, k - k -ta komponenta vektora značajki

1. Izračun srednje vrijednosti svake komponente vektora značajki

$$x_k = \frac{1}{N} \sum_{i=1}^N x_{ik}$$

2. Izračun standardne devijacije svake komponente vektora značajki

$$\sigma_k^2 = \frac{1}{N-1} \sum_{i=1}^N (x_{ik} - x_k)^2$$

3. Od vrijednosti svake komponente vektora značajki se oduzme srednja vrijednost te komponente i podijeli drugim korijenom iz srednje vrijednosti devijacije te komponente

$$x_{ik} = \frac{x_{ik}}{\sigma_k}$$

Nakon postupka normalizacije svi vektori imaju srednje vrijednosti oko nule i jediničnu varijancu.

Primjer: Stvaranje vektora značajki

Kao primjer uzmimo 4 Haralickove značajke: kontrast, energiju, homogenost i entropiju.

Uvedimo oznake radi jednostavnijeg prikaza:

$X_{i,j}$ predstavlja vrijednost funkcije kontrasta za i -ti kut i j -tu udaljenost u GLCM

$Y_{i,j}$ predstavlja vrijednost funkcije energije za i -ti kut i j -tu udaljenost u GLCM

$Z_{i,j}$ predstavlja vrijednost funkcije homogenosti za i -ti kut i j -tu udaljenost u GLCM

$W_{i,j}$ predstavlja vrijednost funkcije entropije za i -ti kut i j -tu udaljenost u GLCM

Ukoliko bi uzeli udaljenost $d = 1$ i kutove $\frac{\pi}{4}, \frac{\pi}{2}, \pi$ kao parametre GLCM, vektor značajki jednog okna bi izgledao na sljedeći način:

Radi jednostavnosti zapisa uvedimo indeks 1 za kut $\frac{\pi}{4}$, 2 za $\frac{\pi}{2}$ i 3 za π .

$$c = (X_{1,1}, X_{2,1}, X_{3,1}, Y_{1,1}, Y_{2,1}, Y_{3,1}, Z_{1,1}, Z_{2,1}, Z_{3,1}, W_{1,1}, W_{2,1}, W_{3,1})$$

Kada je dobiven vektor značajki jednog okna, taj se vektor doda već postojećem vektoru značajki podslike te se postupak ponavlja za svako okno podslike.

Dimenzionalnost vektora značajki pojedinog okna dobije se umnoškom:

A - broj Haralickovih značajki

B - broj kutova u GLCM

C - broj različitih udaljenosti u GLCM

$$n = A \cdot B \cdot C = 4 \cdot 3 \cdot 1 = 12$$

Broj okna u svakoj podslici može se izračunati prema formuli:

X - širina podslike u pikselima Y - visina podslike u pikselima d_1 - širina okna u pikselima d_2 - visina okna u pikselima t - korak okna

$$N = \left(\left\lfloor \frac{X - d_1}{t} \right\rfloor + 1 \right) \cdot \left(\left\lfloor \frac{Y - d_2}{t} \right\rfloor + 1 \right)$$

Ukupna dimenzionalnost vektora značajki pojedine podslike dobije se umnoškom:

$$dim = N \cdot n$$

Za konkretan slučaj s parametrima sa slike 5.1. i korištenim Haralickovim značajkama iz primjera iznad, dimenzionalnost vektora značajki bi iznosila:

$$dim = 15 \cdot 12 = 180$$

Iz primjera je vidljivo kako dimenzionalnost pojedine podslike može postati iznimno velika. Izbor broja Haralickovih značajki, udaljenosti i broja kutova u GLCM ima velik utjecaj na krajnju dimenzionalnost vektora značajki stoga je od velike važnosti promišljanje o izboru parametara svakog modela.

6. Rezultati eksperimenta

U svrhu boljeg razumijevanja dobivenih rezultata eksperimenta napravljena je manja analiza prostorne raspodjele uzoraka za svaki razred. Za analizu su uzeti sljedeći parametri

parametar	vrijednost
vrsta klasifikatora	k -NN
broj susjeda	1
vrsta slike	sivih razina
LBP radijus	1
korak okna	32
veličina okna	64x64
GLCM udaljenosti	1
GLCM kutovi	0, π

Tablica 6.1: Parametri primjera

Analiza se također vrši nad skupom slika koji je korišten u radu. Stvoreni su vektori značajki svake podslike, koji s navedenim parametrima poprimaju dimenzionalnost 240, normlizirani i raspoređeni na način da je svaki od vektora značajki zbrojen zajedno s vektorima razreda kojem pripadaju. Sljedeće su ti vektori uprosječeni čime je dobiven vektor značajki koji ne pripada niti jednoj podslici, a označava zamišljeno središte pojedinog razreda (prototip), u fizici bi to bilo težište.

Nakon dobivenih središta razreda izračunate su euklidske udaljenosti između svakog od razreda kako bi se dobila neka predodžba o razdiobi uzoraka u prostoru. Oznake u tablici 6.2. predstavljaju razrede gustoće: $c_0 \rightarrow$ „no flow”, $c_1 \rightarrow$ „free flow”, $c_2 \rightarrow$ „restricted flow”, $c_3 \rightarrow$ „dense flow”, $c_4 \rightarrow$ „jammed flow”.

Euklidska udaljenost se računa prema:

$$d(x_1, x_2) = \sqrt{\left(\sum_{i=1}^{240} (x_{1i} - x_{2i})^2 \right)}$$

udaljenost između	iznos udaljenosti
$d(c_0, c_1)$	4.49
$d(c_0, c_2)$	6.75
$d(c_0, c_3)$	10.11
$d(c_0, c_4)$	13.62
$d(c_1, c_2)$	4.69
$d(c_1, c_3)$	8.65
$d(c_1, c_4)$	12.5
$d(c_2, c_3)$	5.63
$d(c_2, c_4)$	9.98
$d(c_3, c_4)$	6.27

Tablica 6.2: Euklidske udaljenosti među razredima

Iz priložene tablice moguće je vidjeti kako su središta razreda gustoća koji su bliski jedni drugima, npr. „restricted flow” i „dense flow”, odnosno c_2 i c_3 , bliža nego razredima koji su veoma različiti poput c_0 i c_4 . Manja udaljenost središta znači i povećana vjerojatnost krive klasifikacije. U idealnom slučaju bi razredi bili strogo odvojivi i svaki razredi bi sačinjavali vektori značajki koji formiraju nekakav oblik hiperkugle ili nekog srodnog oblika te ne bi postojali slučajevi gdje neke jedinice previše odskaču od središta svoga razreda i oblika toga razreda. Stvarni život nije savršen te postoji skoro uvijek neki primjer koji je poseban. Potrebno je naglasiti da su udaljenosti izračunate nad normaliziranim vektorima pa su radi toga razloga toliko male.

U te posebne slučajeve spadaju pogreške prilikom označavanja uzoraka i uzorci koje je moguće svrstati u više različitih razreda. Primjer je segmentacija izvorne slike u manje podslike gdje se stvara podslika koja cijepa gusto mnoštvo na dva dijela i sad je problematično jer klasifikator ne zna da li gleda samo gustoću mnoštva koje je prepolovljeno ili gustoću tog mnoštva naspram površine cijele podslike čime dolazi u konflikt jer se dvoumi između dva različita razreda gustoće. Isto tako, baš kao i klasifikator, čovjek se nekad može nedoumiti prilikom označavanja uzoraka odgovarajućim razredima gustoće. Iz istih razloga naša odluka jednom može biti u jednom smjeru, a

drugi put u drugom smjeru čime unosimo nekonzistentnost u skup podataka za učenje. Izbjegavati takve stvari nije lako te je zbog toga potrebno imati čim veći skup podataka kako bi se posebni slučajevi dovoljno malo puta pojavili i kako ne bi stvarali prevelikog utjecaja na ostatak skupa.

udaljenost između	iznos udaljenosti
$d(x_0, c_0)$	10.03
$d(x_0, c_1)$	10.91
$d(x_0, c_2)$	10.63
$d(x_0, c_3)$	12.74
$d(x_0, c_4)$	13.08

Tablica 6.3: Uzorak koji pripada razredu „no flow”

udaljenost između	iznos udaljenosti
$d(x_2, c_0)$	21.46
$d(x_2, c_1)$	20.33
$d(x_2, c_2)$	19.75
$d(x_2, c_3)$	17.86
$d(x_2, c_4)$	19.38

Tablica 6.5: Uzorak koji pripada razredu „restricted flow”

udaljenost između	iznos udaljenosti
$d(x_4, c_0)$	13.83
$d(x_4, c_1)$	13.35
$d(x_4, c_2)$	12.01
$d(x_4, c_3)$	9.37
$d(x_4, c_4)$	8.6

Tablica 6.7: Uzorak koji pripada razredu „jammed flow”

udaljenost između	iznos udaljenosti
$d(x_1, c_0)$	11.4
$d(x_1, c_1)$	10.93
$d(x_1, c_2)$	10.33
$d(x_1, c_3)$	11.75
$d(x_1, c_4)$	13.37

Tablica 6.4: Uzorak koji pripada razredu „free flow”

udaljenost između	iznos udaljenosti
$d(x_3, c_0)$	23.25
$d(x_3, c_1)$	22.55
$d(x_3, c_2)$	21.33
$d(x_3, c_3)$	18.26
$d(x_3, c_4)$	18.37

Tablica 6.6: Uzorak koji pripada razredu „dense flow”

ispravna oznaka	odluka klasifikatora
0	0
1	1
2	2
3	3
4	4

Tablica 6.8: Predočeni uzorci klasifikatoru i rezultati klasifikacije

Za svaki od razreda uzet je jedan još neviđen uzorak, izračunat je vektor značajki te je za taj vektor izračunata udaljenost do svakog od središta razreda (c_0, c_1, c_2, c_3, c_4). Uz to je pripremljen k -NN klasifikator s parametrima navedenim u tablici 6.1. Svaki od još neviđenih vektora predat je klasifikatoru kako bi vidjeli i njegov odluku. Svaki

x_i u tablicama iznad, gdje je $i = 0, 1, 2, 3, 4$, označava pripadnika jednog od razreda gustoće.

Iz tablice 6.8. moguće je vidjeti kako je klasifikator ispravno dodijelio oznaku svakome uzorku iako je njegova točnost 90.52%. Iščitavanjem tablica 6.3. - 6.7. zamjećuje se da se odluka klasifikatora razlikuje od odluke koja bi se donijela ako bi se promatrale najmanje udaljenosti do središta pojedinih razreda gustoće. Razlog za takvu odluku klasifikatora je jer radi na principu najbližih k susjeda, u ovom primjeru najbližeg jednog susjeda. Rezultati klasifikacije odgovaraju najbližim središtima za slučajeve razreda: 0, 3, 4 dok je kod ostalih središte razreda različito od dodijeljene odluke.

Primjer pokazuje kako je veoma važno uzeti prikladnu mjeru udaljenosti koju klasifikator koristi za usporedbu dva različita uzorka. Pritom nije svejedno uspoređuje li se nepoznati uzorak sa svim ostalim uzorcima ili samo onim umjetnim uzorcima koji bi trebali predstavljati središta svakog od razreda. Izbor vrste usporedbe pojedinih vektora značajki ovisi o samim podacima koje koristimo za učenje klasifikatora. Ako su nepravilne grupe vektora značajki u n -dimenzionalnom prostoru, možda je bolja ideja izabrati klasifikaciju prema bližim susjedima dok u slučaju da imamo donekle pravilne grupe (hiperkugle i sl.) je bolje izabrati središta pojedinih razreda kao vektore s kojima se svi dosad neviđeni uspoređuju jer i time u krajnjem slučaju štedimo vrijeme potrošeno na usporedbu.

Eksperimentom se došlo do zaključka kako k -NN klasifikator daje dobre rezultate za manje vrijednosti broja k pri čemu je on neparan i to za vrijednosti $k = 1$ ili $k = 3$. Većim vrijednostima se unosi određena greška jer raspodjela vektora značajki nije uvelike pravilna. Veličina koraka okna od 64 i 32 pokazala se kao najbolja mjera obzirom na veličinu okna od 64x64. Manji koraci ne donose nikakvu novu informaciju nego samo povećavaju vrijeme potrebno za obradu pojedine podslike, a koraci veći od 64 nemaju smisla obzirom da je veličina okna zadržana stalnom.

Općenito je uvođenje većeg broja različitih vrijednosti kutova davalo bolje rezultate vezano uz točnost klasifikacije međutim pogledom u tablicu rezultata u dodatku B moguće je vidjeti kako postoje modeli klasifikatora koji rade samo s kutem od 0 stupnjeva i postižu bolje rezultate od nekih drugih koji imaju npr. 4 različita kuta. Klasifikatori koji su koristili slike sivih razina generalno imaju neznatno veću točnost klasifikacije u usporedbi s klasifikatorima koji rade sa slikama sivih razina nad kojima je primijenjen

operator gradijenta, pritom su obje vrste klasifikatora imala iste parametre osim vrste slike koju koriste.

Izbor Haralickovih značajki koje će model klasifikatora koristiti za izračunavanje nad nekom podslikom je jedna od najvažnijih ako ne i najvažnija stavka. Važno je napomenuti kako se pojedine značajke mogu izuzetno brzo izračunati jer je npr. potrebna samo suma GLCM pomnožena s nekim brojem dok druge značajke izuzetno vremenski optereće postupak učenja pa je stoga izbor značajki ograničeni i tom ogradom. U ovom radu nikad nije uzeto više od 7 različitih Haralickovih značajki zbog ograničenih računalnih i vremenskih resursa. Smisleno je koristiti skup parametara koji nije prezahtjevan jer preveliko čekanje na klasifikaciju slike nije poželjno u situacijama koje zahtijevaju možda brzu reakciju. Druga opcije je koristiti grafičke procesore i paralelizirati koliko je to moguće sam postupak izračunavanja čime se automatski diže kompleksnost samog sustava.

U radu ¹ odabrane su Haralickove značajke: energija, kontrast, homogenost i entropija. Te konkretne značajke veoma se brzo daju izračunati, a daju vrlo dobre rezultate kako je vidljivo iz tablice u dodatku B. Izbor tih značajki uvelike poboljšava performanse samog postupka naspram nekih drugih izbora parametara s više različitih značajki, a ne narušava značajno točnost klasifikacije. Postoje bolji odabiri značajki, ali kako je već navedeno, računski su zahtjevniji i time njihovo korištenje baš nema smisla.

ODABRATI NEKE KLASIFIKAORE I PROVJERITI TOČNOST

¹?

7. Prikladnost korištenog postupka na drugome skupu podataka

Naći neki skup podataka na internetu s velikom brojem ljudi.

Ocijeniti prikladnost postupka na novome skupu podataka.

8. Zaključak

Zaključak.

LITERATURA

- O. Barkan, J. Weill, L. Wolf, i H. Aronowitz. Fast high dimensional vector multiplication face recognition. U *2013 IEEE International Conference on Computer Vision*, stranice 1960–1967, 2013.
- Dong-chen He i Li Wang. Texture unit, texture spectrum, and texture analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 28(4):509–512, 1990.
- R. M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, 1979.
- Intel. Opencv.org. <https://opencv.org/>, 2000. Posjećeno 12.5.2020.
- A. N. Marana, S. A. Velastin, L. F. Costa, i R. A. Lotufo. Estimation of crowd density using image processing. U *IEE Colloquium on Image Processing for Security Applications (Digest No: 1997/074)*, stranice 11/1–11/8, 1997.
- T. Ojala, M. Pietikainen, i D. Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. U *Proceedings of 12th International Conference on Pattern Recognition*, svezak 1, stranice 582–585 vol.1, 1994.
- Abishai Polus, Joseph L. Schofer, i Ariela Ushpiz. Pedestrian flow and level of service. *Journal of Transportation Engineering*, 109(1):46–56, 1983. doi: 10.1061/(ASCE)0733-947X(1983)109:1(46).

POPIS SLIKA

2.1. Prikaz poretka jedinica tekstone u susjedstvu srednjeg piksela.	3
2.2. Susjedstvo $V = \{40, 62, 20, 41, 30, 10, 40, 130, 79\}$	3
2.3. Jedinica tekstone $TU = \{2, 0, 2, 0, 0, 1, 2, 2\}$	3
2.4. Susjedstvo nakon usporedbe s centralnim pikselom	3
2.5. Susjedstvo $V = \{55, 71, 171, 190, 78, 78, 12, 24, 5\}$	3
2.6. Primjer dobivenih LBP slika iz izvorne slike uz različite radijuse oko centralnog piksela	4
2.7. Primjer crno bijele slike	5
2.8. Prikaz GLCM prije i nakon normalizacije	6
2.9. Prikaz dva razreda s jednim nestandardnim uzorkom	7
2.10. Prikaz uzoraka dva razreda i novog uzorka za klasifikaciju	8
2.11. Prikaz dviju mogućih hiperravnina između dva razreda	9
2.12. Prikaz potencijalnih hiperravnina koje dijele najbliže točke razreda . .	10
2.13. Linearno nerazdvojivi razredi	11
2.14. Linearno nerazdvojivi razredi u većoj dimenziji postaju linearno odvojivi	11
2.15. Prikaz smjera gradijenta plavim strelicama, tamnija boja označava veće vrijednosti.	13
2.16. Izvorna slika	14
2.17. Prijenjen Sobelov operator na izvornoj slici	14
2.18. Prijenjen Sobelov operator u smjeru y na izvornoj slici	14
2.19. Prijenjen Sobelov operator u smjeru x na izvornoj slici	14
3.1. Prikaz pojedinih gustoća mnoštva	16
3.2. Prikaz arhitekture sustava	19
4.1. Prikaz početnog prozora nakon pokretanja aplikacije	20
4.2. Prikaz glavnog prozora	21
4.3. Prikaz stranice za obradu izvornih slika	22

4.4. Prikaz izgleda podslike s nešenih dimenzijama	23
4.5. Prikaz podijeljene izvorne slika na manje podslike	24
4.6. Prikaz prozora za označavanje podslika	24
4.7. Početni prozor za stvaranje vektora značajki	26
4.8. Izgled prozora za stvaranje konfiguracije	26
4.9. Prikaz napretka pojedinih konfiguracija	29
4.10. Prikaz stranice za odabir jednog modela	30
4.11. Prikaz stranice za odabir dva modela	31
4.12. Prikaz stranice za klasifikaciju slika	32
4.13. Prikaz klasificirane slike i procjene broja ljudi na temelju gustoće mnoštva	32
5.1. Prikaz prozora tehnike klizećeg okna	34

POPIS TABLICA

2.1. Prikaz nekih popularnih jezgara	12
4.1. Opis parametara konfiguracije	28
6.1. Parametri primjera	38
6.2. Euklidske udaljenosti među razredima	39
6.3. Uzorak koji pripada razredu „no flow”	40
6.4. Uzorak koji pripada razredu „free flow”	40
6.5. Uzorak koji pripada razredu „restricted flow”	40
6.6. Uzorak koji pripada razredu „dense flow”	40
6.7. Uzorak koji pripada razredu „jammed flow”	40
6.8. Predočeni uzorci klasifikatoru i rezultati klasifikacije	40
B.1. Prikaz rezultata različitih modela klasifikatora	54

Procjena gustoće mnoštva ljudi utemeljena na matrici pojavnosti lokalnih binarnih uzoraka

Sažetak

Procjena gustoće mnoštva od velike je važnosti u video nadzoru i upravljanju mnoštvom. Predložene su različite metode za rješavanje ovog problema, međutim, većina ih procjenjuje gustoću mnoštva cijele slike pritom ignorirajući lokalna mnoštva. U ovom radu primjenjuje se operator lokalnih binarnih značajki nad sivim slikama i sivim slikama nad kojima je primjenjen operator gradijenta. Iz takvih slika se pomoću matrice pojavnosti lokalnih binarnih značajki, postupkom klizećeg okna, izračunavaju Haralickove značajke koje tvore vektore značajki. Koriste se dvije vrste klasifikatora: k -NN i SVM koji se na kraju spajaju u jedan klasifikator i utjecaj svakog klasifikatora uvjetovan je njegovom točnošću klasifikacije.

Ključne riječi: Lokalna binarna značajka, Matrica pojavnosti lokalnih binarnih uzoraka, Vektor značajki, k -NN klasifikator, SVM klasifikator, gustoća mnoštva

Crowd density estimation based on local binary pattern co-occurrence matrix

Abstract

Crowd density estimation is of great importance in video surveillance and in crowd management. Most of the existing methods estimate crowd density of a whole image while ignoring local density areas. In this work local binary pattern is used on gray and gradient images. These pictures are used to extract Haralick features from local binary pattern co-occurrence matrix, using sliding window technique, which later form feature vectors. Two type of classifiers are used: k -NN and SVM. They are later fused together and the influence of each one is dependant on the classifier accuracy of training set of images.

Keywords: Keywords.

Dodatak A

Teksturne značajke

Oznake korištene dalje u tekstu:

N_g - broj sivih razina $p(i, j)$ - (i, j) -ti element normalizirane matrice pojavnosti sivih razina

$p_x(i)$ - i -ti element u matrici marginalnih vjerojatnosti koja je dobivena zbrajanjem redaka $= \sum_{j=1}^{N_g} p(i, j)$

$p_y(j)$ - j -ti element u matrici marginalnih vjerojatnosti koja je dobivena zbrajanjem stupaca $= \sum_{i=1}^{N_g} p(i, j)$

$$p_{x+y}(k) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \quad k = 2, 3, \dots, 2N_g, \quad i + j = k$$

$$p_{x-y}(k) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \quad k = 0, 1, \dots, N_g - 1, \quad |i - j| = k$$

$$\sum_i \text{ predstavlja pokratu } \sum_{i=1}^{N_g}, \text{ a } \sum_j \text{ predstavlja pokratu } \sum_{j=1}^{N_g}$$

1) Angular second moment:

$$f_1 = \sum_i \sum_j p(i, j)^2$$

2) Contrast:

$$f_2 = \sum_{n=0}^{N_g-1} n^2 \left(\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j) \right), |i - j| = n$$

3) Correlation:

$$f_3 = \frac{\sum_i \sum_j (ij) p(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y}$$

4) Sum of squares: variance:

$$f_4 = \sum_i \sum_j (i - \mu)^2 p(i, j)$$

5) Inverse difference moment:

$$\sum_i \sum_j \frac{1}{1 + (i - j)^2} p(i, j)$$

6) Sum of average:

$$f_6 = \sum_{i=2}^{2N_g} i p_{x+y}(i)$$

7) Sum variance:

$$f_7 = \sum_{i=2}^{2N_g} (i - f_8)^2 p_{x+y}(i)$$

8) Sum entropy:

$$f_8 = - \sum_{i=2}^{2N_g} p_{x+y}(i) \log(p_{x+y}(i))$$

9) Entropy:

$$f_9 = - \sum_i \sum_j p(i, j) \log(p(i, j))$$

10) Difference variance

$$f_{10} = \text{variance of } p_{x-y}$$

11) Difference entropy:

$$f_{11} = - \sum_{i=0}^{N_g-1} p_{x-y}(i) \log(p_{x-y}(i))$$

12), 13) Information measures of correlation

$$f_{12} = \frac{HXY - HXY1}{\max\{HX, HY\}}$$

$$f_{13} = \sqrt{(1 - \exp - 2(HXY2 - HXY))}$$

$$HX = - \sum_i \sum_j p(i, j) \log(p(i, j))$$

$$HXY1 = - \sum_i \sum_j p(i, j) \log(p_x(i) p_y(y))$$

$$HXY2 = - \sum_i \sum_j p_x(i) p_y(y) \log(p_x(i) p_y(y))$$

14) Maximal correlation coefficient:

$$f_{14} = \sqrt{\text{Second largest eigenvalue of } Q}$$

$$Q(i, j) = \sum_k \frac{p(i, k)p(j, k)}{p_x(i)p_y(k)}$$

Dodatak B

Rezultati eksperimenta

redni broj	vrsta klasifikatora	vrsta podsluke	radius	GLCM udaljenosti	korak okna	veličina okna	kutovi	broj susjeda	značajke	greška
1	<i>k</i> -NN	grad	1	1	32	64x64	$0, \pi$	1	f1,f2,f5,f9	9.9%
2	SVM	gray	1	1	32	64x64	$0, \pi$	-	f1,f2,f5,f9	8.86%
3	SVM	grad	1	1	32	64x64	$0, \pi$	-	f1,f2,f5,f9	7.92%
4	<i>k</i> -NN	gray	1	1	32	64x64	$0, \pi$	1	f1,f2,f5,f9	9.52%
5	SVM	gray	1	1	32	64x64	$0, \pi$	-	f1,f2,f3,f4,f5,f9	8.39%
6	SVM	grad	1	1	32	64x64	$0, \pi$	-	f1,f2,f3,f4,f5,f9	8.11%
7	<i>k</i> -NN	grad	1	1	32	64x64	$0, \pi$	3	f1,f2,f3,f4,f5,f9	8.58%
8	<i>k</i> -NN	gray	1	1	32	64x64	$0, \pi$	3	f1,f2,f3,f4,f5,f9	7.45%
9	SVM	grad	1	1	32	64x64	$0, \pi$	-	f8,f9,f12,f13	9.99%
10	<i>k</i> -NN	grad	1	1	32	64x64	$0, \pi$	3	f8,f9,f12,f13	11.31%
11	<i>k</i> -NN	gray	1	1	32	64x64	$0, \pi$	3	f8,f9,f12,f13	9.24%
12	SVM	gray	1	1	32	64x64	$0, \pi$	-	f8,f9,f12,f13	8.95%
13	SVM	gray	1	1	32	64x64	$0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$	-	f4,f5,f9,f11	7.35%
14	<i>k</i> -NN	gray	1	1	32	64x64	$0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$	3	f4,f5,f9,f11	6.79%
15	<i>k</i> -NN	grad	1	1	32	64x64	$0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$	3	f4,f5,f9,f11	7.45%
16	SVM	grad	1	1	32	64x64	$0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$	-	f4,f5,f9,f11	8.58%
17	<i>k</i> -NN	grad	1	1	64	64x64	$0, \pi$	3	f9,f10,f11,f12,f13	11.12%
18	SVM	gray	1	1	64	64x64	$0, \pi$	-	f9,f10,f11,f12,f13	10.08%
19	SVM	grad	1	1	64	64x64	$0, \pi$	-	f9,f10,f11,f12,f13	9.9%

20	<i>k</i> -NN	gray	1	1	64	64x64	$0, \pi$	1	f9,f10,f11,f12,f13	10.65%
21	SVM	grad	1	1	32	64x64	0	-	f1,f2,f4,f7,f10,f12,f13	8.95%
22	SVM	gray	1	1	32	64x64	0	-	f1,f2,f4,f7,f10,f12,f13	7.82%
23	SVM	gray	1	1	32	64x64	$0, \pi$	-	f1,f2,f4,f7,f10,f12,f13	7.82%
24	SVM	grad	1	1	32	64x64	$0, \pi$	-	f1,f2,f4,f7,f10,f12,f13	8.95%
25	SVM	grad	1	1	32	64x64	$0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$	-	f1,f2,f3,f4,f5,f9	8.77%
26	SVM	gray	1	1	32	64x64	$0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$	-	f1,f2,f3,f4,f5,f9	8.29%
27	SVM	gray	1	1	32	64x64	$0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$	-	f1,f2,f3,f4,f5,f9,f12,f13	8.2%
28	SVM	grad	1	1	32	64x64	$0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}$	-	f1,f2,f3,f4,f5,f9,f12,f13	8.58%
29	SVM	grad	1	1	64	64x64	0	-	f2,f6,f7,f8,f9	8.58%
30	SVM	gray	1	1	64	64x64	0	-	f2,f6,f7,f8,f9	8.58%
31	SVM	gray	1	1	32	64x64	0	-	f2,f6,f7,f8,f9	7.73%
32	SVM	grad	1	1	32	64x64	0	-	f2,f6,f7,f8,f9	8.95%
32	SVM	grad	1	1	32	64x64	0	-	f2,f6,f7,f8,f9	8.95%
32	SVM	grad	1	1	32	64x64	0	-	f2,f6,f7,f8,f9	8.95%
32	SVM	grad	1	1	32	64x64	0	-	f2,f6,f7,f8,f9	8.95%

Tablica B.1: Prikaz rezultata različitih modela klasifikatora