

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 4741

Raspoznavanje kovanica sa slika

Andrija Miličević

Zagreb, srpanj 2016.

Zagreb, 17. ožujka 2016.

ZAVRŠNI ZADATAK br. 4741

Pristupnik: **Andrija Miličević (0108059515)**
Studij: Računarstvo
Modul: Računarska znanost

Zadatak: **Raspoznavanje kovanica sa slika**

Opis zadatka:

Raspoznavanje uzoraka čest je zadatak u današnjim modernim računalnim sustavima. Postoji mnoštvo prikladnih algoritama i primjena.

U okviru ovog završnog rada potrebno je razmotriti situaciju u kojoj su dostupne fotografije niza različitih kovanica na bijeloj podlozi. Potrebno je proučiti i implementirati odgovarajuće algoritme za detekciju kovanica na slici te njihovu klasifikaciju. U okviru rada potrebno je razmotriti jednostavan slučaj u kojem se kovanice ne preklapaju te istražiti u kojoj mjeri potencijalno preklapanje otežava zadatak klasifikacije. Za implementirane postupke potrebno je provesti vrednovanje. Radu je potrebno priložiti algoritme, izvorne kodove i rezultate uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Zadatak uručen pristupniku: 18. ožujka 2016.

Rok za predaju rada: 17. lipnja 2016.

Mentor:



Doc. dr. sc. Marko Čupić

Djelovođa:



Doc. dr. sc. Tomislav Hrkać

Predsjednik odbora za
završni rad modula:



Prof. dr. sc. Siniša Srbljić

SADRŽAJ

1. Uvod	1
2. Definicija problema	2
3. Detekcija kružnica	4
3.1. Obrada slike	4
3.1.1. Pretvorba u sliku s 256 nijansi sive	5
3.1.2. Primjena Gaussovog filtra	6
3.1.3. Filtrirana slika	7
3.1.4. Izrada matrice gradijenata	8
3.1.5. Izrada matrice rubnih točaka	9
3.2. Algoritmi za detekciju kružnica na slici	10
3.2.1. Houghova transformacija za kružnice	11
3.2.2. Pronalazak kružnica koristeći parove gradijentnih vektora	11
3.2.3. Umjetni imunološki algoritam za pronalazak kružnica	12
3.3. Imunološki algoritmi	13
3.3.1. Jednostavni imunološki algoritam	13
3.3.2. Algoritam klonske selekcije	14
3.3.3. Algoritam implementiran u radu	15
4. Skup podataka	19
4.1. Skup podataka za učenje	19
4.2. Skup podataka za ispitivanje	19
4.3. Skup podataka za rad klasifikatora	20
5. Klasifikacija	22
5.1. Izlučivanje značajki	22
5.2. Izrada novog skupa podataka	22
5.3. Značajke neovisne o rotaciji	22

5.4. Implementirani klasifikator	24
6. Rezultati	25
7. Zaključak	28
Literatura	29

1. Uvod

Raspoznavanje uzoraka temeljni je spoznajni postupak kojim mozak povezuje podražaje iz okoline s informacijama dohvaćenima iz memorije, što omogućuje interpretaciju istih. Ljudi pri rođenju nemaju gotovo nikakvu mogućnost raspoznavanja, no tijekom vremena uče razlikovati podražaje te prepoznavati uzorke među njima. Veliki napor uložen je u pokušaje prenošenja tog inteligentnog ponašanja čovjeka na računalu. Time je i nastalo jedno veliko područje u sklopu računarske znanosti.

U području umjetne inteligencije, raspoznavanje uzoraka je dio područja strojnog učenja koji je usmjeren na prepoznavanje uzoraka i pravilnosti u podacima na temelju nekog znanja ili naučenog postupka. Podaci mogu biti tekst, slike, zvuk te gotovo sve što se može digitalno zapisati. Prepoznavanje lica na Facebooku, pjesme na temelju snimke od par sekundi te rukopisa, samo su neki od primjera primjene.

Definicija problema te pristup rješavanju istog prikazan je u poglavlju 2. Veći dio ovog rada posvećen je samoj detekciji kovanica, tj. detekciji rubnih kružnica svih kovanica na slici što je opisano u poglavlju 3. Navedeno poglavlje opisuje potrebnu obradu slike, proučene radove za pronalazak kružnica na slici te implementirani vlastiti algoritam. U poglavlju 4 opisan je korišteni skup podataka koji se sastoji od niza slika kovanica na bijeloj podlozi. Osim toga, opisan je skup podataka pripremljen algoritmom iz prethodnog poglavlja za rad klasifikatora. U poglavlju 5 opisana je uloga klasifikatora, pronalazak značajki sa slika te vlastita implementacija jednostavnog klasifikatora. Rezultati rada opisani su u poglavlju 6, dok se sam zaključak rada nalazi u poglavlju 7.

2. Definicija problema

Ovaj rad bavi se problematikom raspoznavanja kovanica sa slika. Ideja rada je proučavanje postojećih rješenja te pokušaj implementacije jednostavnog rješenja za slike na kojima se nalaze kovanice na bijeloj podlozi. Kovanice na slikama se ne preklapaju. Osim toga, ovaj rad proučava u kojoj mjeri i na koji način preklapanje kovanica otežava cijeli proces raspoznavanja istih.

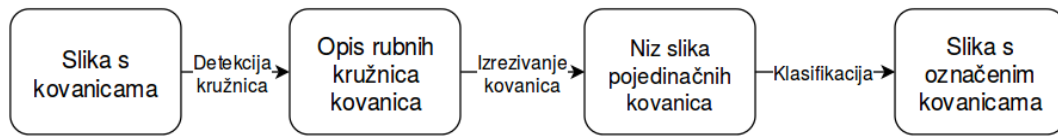
Cilj rada je implementacija programa koji bi iz prethodno opisanih slika označio kovanicu na slici te označio o kojoj se kovanici radi, u obliku vrijednost-valuta, što je prikazano na slici 2.1.



Slika 2.1: Ulazna i ciljna slika.

Sam postupak raspoznavanja kovanica podijeljen je u nezavisne dijelove. Detekcija kovanica na slici odvojena je od same klasifikacije istih. Stoga je moguće je iskoristiti različite klasifikatore nad istim podacima te ih međusobno usporediti. Neovisnost je postignuta tako da nakon detekcije kovanica na slici dolazi međukorak, koji iz slike od više kovanica stvara niz slika pojedinačnih kovanica ili iz slike izvlači bitne značajke pojedinačnih kovanica, koje služe za rad klasifikatora. Slika 2.2 prikazuje postupak

raspoznavanja.



Slika 2.2: Implementacija algoritma.

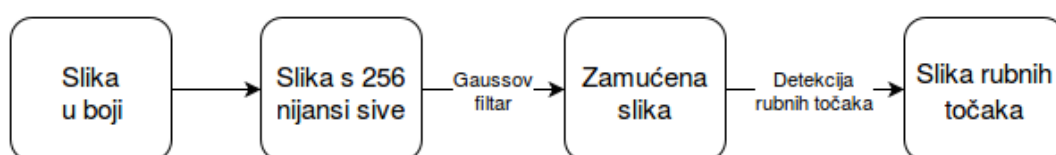
Problem preklapanja kovanica može se riješiti na dva načina. Jedan od načina je za sve kovanice koje nisu u cijelosti prikazane na slici, napraviti slike pojedinačnih kovanice koje prikazuju samo vidljivi dio kovanice, dok se ostatak jednostavno prebriše bijelom bojom. Drugi način je izrada prethodno navedenih značajki kovanica, koje bi klasifikator iskoristio za svoj rad.

3. Detekcija kružnica

Prvi korak u raspoznavanju kovanica jest njihovo detektiranje na slici. Detektiranje kovanica svodi se na pronalazak parametara kružnica koje bi opisivale rubove kovanica kako bi se iste mogle pripremiti za rad klasifikatora. Takve kružnice nazivamo rubnim kružnicama kovanica. Postoji mnoštvo algoritama koji se bave detekcijom kružnica na slici. Svi imaju zajednički prvi pripremni dio, odnosno obradu slike kako bi se dobio oblik pogodan za rad samog algoritma.

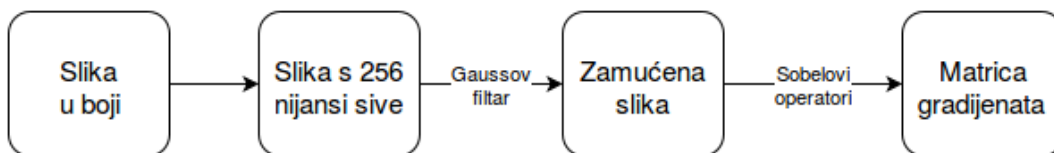
3.1. Obrada slike

Svi algoritmi podrazumijevaju određenu obradu slike prije samog pokretanja. Većina algoritama koristi identičan slijed postupaka pri obradi. Prvi korak je pretvorba slike u boji u sliku s 256 nijansi sive. Sljedeći korak je primjena filtra koji bi zamaglio sliku te ju time bolje pripremio za detekciju rubnih točaka objekata, tj. kovanica na slici. Konačno, zadnji korak jest pretvorba takve slike u sliku ili matricu rubnih točaka. Slika 3.1 prikazuje uobičajeni slijed obrade slike.



Slika 3.1: Obrada slike.

Primijenjena metoda obrade slika u ovom radu razlikuje se samo u posljednjem koraku ovoga slijeda. Umjesto primjene algoritma za detekciju rubnih točaka, koji stvara binarnu sliku, koristi se samo filter koji stvara matricu gradijenata u svakoj točki slike. Način obrade slike korišten u ovom radu prikazan je na slici 3.2.



Slika 3.2: Implementirani slijed obrade slike.

3.1.1. Pretvorba u sliku s 256 nijansi sive

Ideja ovog koraka je pretpostavka da se pretvorbom slike u boji u sliku s 256 nijansi sive neće izgubiti značajna količina informacija, a smanjit će se ukupna količina podataka koji opisuju sliku, te time pojednostavniti daljnja obrada slike te rad samog algoritma. Iako ovaj korak nije nužan, sastavni je dio gotovo svakog rada koji se bavi ovom problematikom.

Od mnoštva popularnijih načina pretvorbe (Helland, 2011), najjednostavniji način je zamjena tri vrijednosti pojedinog *pixela* srednjom vrijednošću crvene, zelene i plave komponente:

$$I = \frac{C + Z + P}{3}$$

gdje je I vrijednost sive boje, C vrijednost crvene komponente, Z vrijednost zelene komponente, a P vrijednost plave komponente u promatranoj točki.

U ovom radu korišten je izraz iz (Wikipedia, 2016) koji pri pretvorbi slike u boji u sliku s 256 nijansi sive zadržava istu razinu svjetline:

$$I = \frac{0.21 * C + 0.72 * Z + 0.07 * P}{3}.$$

Slika 3.3 prikazuje pretvorbu primjenom prethodno navedenog izraza.



Slika 3.3: Pretvorba u sliku s 256 nijansi sive.

3.1.2. Primjena Gaussovog filtra

Standardni dio obrade slike je primjena Gaussovog filtra kako bi se zamutila slika te olakšao pronalazak rubnih točaka. Zamućenje slike utječe tako da smanjuje utjecaj šumova i finih detalja na slici, dok ima mali utjecaj na rubne točke koje imaju veći gradijent prijelaza prema susjednim točkama. Time olakšava postupak detekcije "jačih" rubnih točaka.

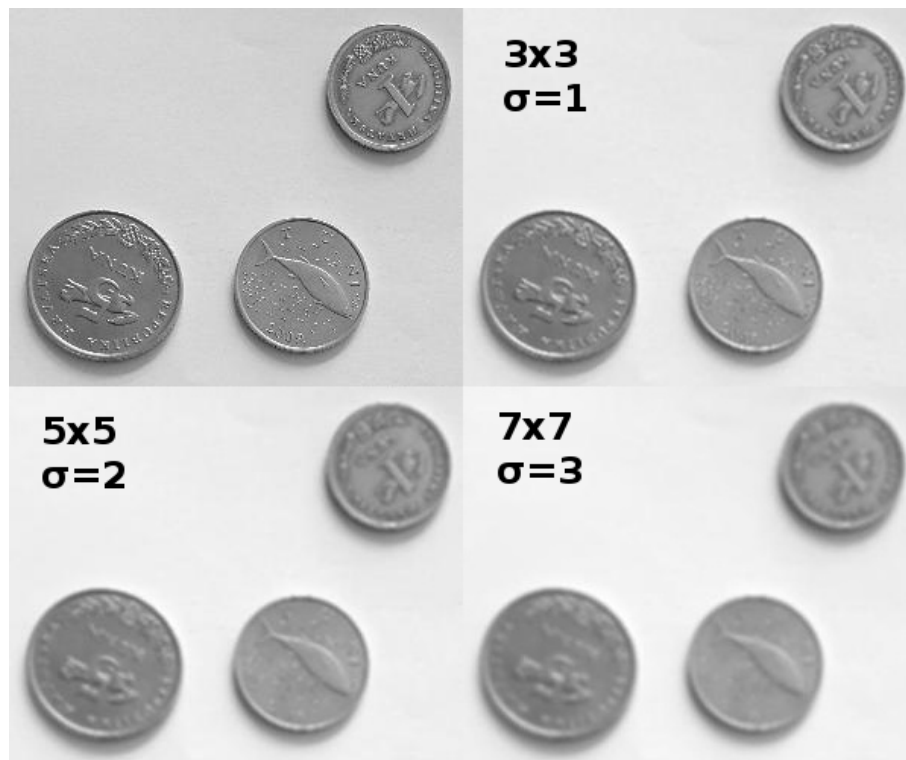
Izgradnja filtra započinje odabirom dimenzija matrica $(2n + 1) \times (2n + 1)$ kvadratnog oblika, pri čemu je n pozitivan cijeli broj. Matrica se gradi tako da se za svaku vrijednost i i j iz intervala cijelih brojeva $[-n, +n]$ primjeni Gaussova dvodimenzionalna formula:

$$Gauss(i, j) = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}}.$$

Matrica 3×3 , uz $\sigma = 1$, dana je u nastavku:

$$Gauss_{3 \times 3} = \begin{bmatrix} 0.075 & 0.124 & 0.075 \\ 0.124 & 0.204 & 0.124 \\ 0.075 & 0.124 & 0.075 \end{bmatrix}.$$

Svaki element matrice se dodatno podijeli ukupnom sumom svih elemenata kako bi



Slika 3.4: Primjena Gaussovog filtra na sliku s 256 nijansi sive prikazanu na slici 3.3.

se matrica normalizirala, tj. kako bi sada suma svih elemenata bila jednaka 1. Time se osigurava da se primjenom filtra ne gubi na ukupnom intenzitetu slike. Slika 3.4 prikazuje primjenu 3 različita Gaussova filtra. U radu je korištena 5×5 Gaussova matrica, uz $\sigma = 2$. Računanje filtrirane slike opisano je u nastavku.

3.1.3. Filtrirana slika

Neka je slika predstavljena matricom elemenata $[g_{x,y}]$, a filter dimenzija $(2n + 1) \times (2n + 1)$ matricom $[m_{i,j}]$ gdje elemente u posljednjoj matrici numeriramo oznakama od $-n$ do $+n$, pri čemu centralni element ima oznaku $m_{0,0}$. Filtriranu sliku, predstavljenu matricom elemenata $[h_{x,y}]$, računamo slikovni element po slikovni element sljedećim izrazom:

$$h_{x,y} = \sum_{i=-n}^n \sum_{j=-n}^n g_{x+i,y+j} \cdot m_{i,j}$$

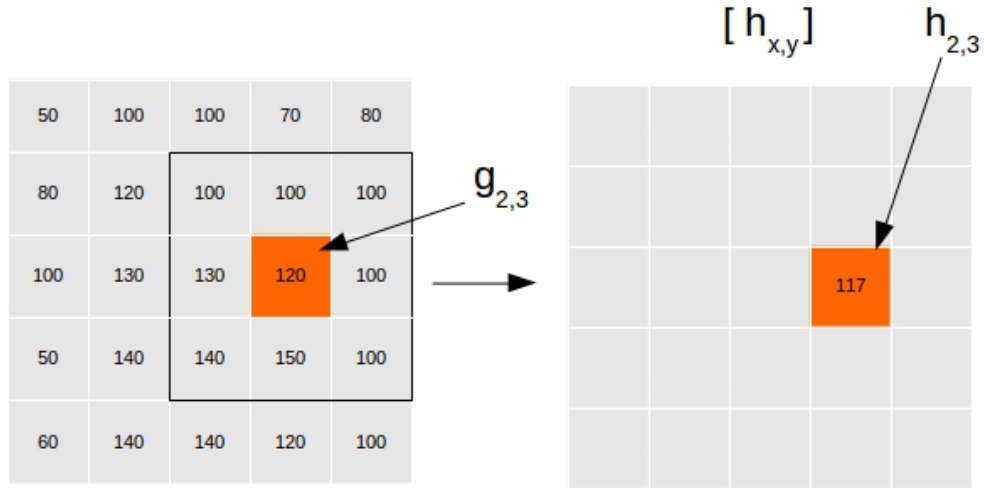
pri čemu se za nepostojeće elemente $g_{x+i,y+j}$, tj. ukoliko je jedan od indeksa $x + i$ ili $y + j$ manji od 0 ili veći od dimenzija matrice elemenata slike, uzima vrijednost $g_{x,y}$. Primjer matrica $[g_{x,y}]$ i $[m_{i,j}]$ prikazan je na slici 3.5, pri čemu je $[g_{x,y}]$ matrica dimenzija 5×5 , a $[m_{i,j}]$ Gaussova matrica dimenzija 3×3 , uz $\sigma = 1$.

[$g_{x,y}$]					[$m_{i,j}$]		
50	100	100	70	80	0.075	0.124	0.075
80	120	100	100	100	0.124	0.204	0.124
100	130	130	120	100	0.075	0.124	0.075
50	140	140	150	100			
60	140	140	120	100			

Slika 3.5: Matrice $[g_{x,y}]$ i $[m_{i,j}]$.

Slika 3.6 prikazuje primjenu filtra $[m_{i,j}]$ na $[g_{x,y}]$ te izračun elementa $h_{2,3}$ filtrirane matrice $[h_{x,y}]$. Izračun elementa $h_{2,3}$ prethodno navedenim izrazom prikazan je u nastavku.

$$\begin{aligned}
h_{2,3} &= 100 * 0.075 + 100 * 0.124 + 100 * 0.075 \\
&\quad + 130 * 0.124 + 120 * 0.204 + 100 * 0.124 \\
&\quad + 140 * 0.075 + 150 * 0.124 + 100 * 0.075 \\
&= 117.
\end{aligned}$$



Slika 3.6: Izračun elementa $h_{2,3}$ primjenom filtra $[m_{i,j}]$ na $[g_{x,y}]$ sa slike 3.5.

3.1.4. Izrada matrice gradijenata

Umjesto posljednjeg koraka uobičajene obrade slike, u ovom radu korištena je obrada pri kojoj se dobiva matrica približnih gradijenata koristeći dobro poznate Sobelove operatore. Sobelovi operatori su dvije matrice, jedna za svaku os:

$$Sobel_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad Sobel_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix}$$

gdje je G_x matrica, tj. filter koji računa približni gradijent u horizontalnom smjeru, dok je G_y matrica, tj. filter koji računa približni gradijent u vertikalnom smjeru. Kako su u ovom radu korištene slike velikih dimenzija, umjesto klasičnih 3×3 matrica korištena

je prikladnija 5×5 matrica:

$$Sobel_x = \begin{bmatrix} -2 & -1 & 0 & +1 & +2 \\ -3 & -2 & 0 & +2 & +3 \\ -4 & -3 & 0 & +3 & +4 \\ -3 & -2 & 0 & +2 & +3 \\ -2 & -1 & 0 & +1 & +2 \end{bmatrix} \quad Sobel_y = \begin{bmatrix} -2 & -3 & -4 & -3 & -2 \\ -1 & -2 & -3 & -2 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ +1 & +2 & +3 & +2 & +1 \\ +2 & +3 & +4 & +3 & +2 \end{bmatrix}$$

Ideja Sobelovih operatora je računanje razlike između nasuprotnih bliskih točaka za pojedine točke te izražavanje približnog gradijenta u toj točki. Primjenom Sobelovih operatora na sliku 3.4 dobivaju se dvije matrice gradijenata. Za potrebe prikaza, napravljene su dvije nove matrice koristeći skalirane apsolutne vrijednosti prethodno dobivenih matrica gradijenata. Vrijednosti su skalirane na cijele brojeve iz intervala $[0, 255]$ te prikazane na slici 3.7 kao nijanse sive boje.



Slika 3.7: Primjena $Sobel_x$ i $Sobel_y$ filtra na sliku 3.4.

Kako operatori djeluju u samo dvije dimenzije, u ovom radu koristi se aproksimacija ukupnog gradijenta u zadanoj točki tako da se zbroje apsolutne vrijednosti dobivene primjenom $Sobel_x$ i $Sobel_y$ filtra. Ova aproksimacija nije uobičajena, no pokazala se dovoljno dobrom. Konačna matrica gradijenata skalirana je na vrijednosti iz intervala $[0, 255]$ te je kao slika s 256 nijansi sive prikazana na slici 3.8.

3.1.5. Izrada matrice rubnih točaka

Posljednji korak u standardnoj obradi slike je detekcija rubnih točaka te pretvorba slike s 256 nijansi sive u binarnu sliku, tj. binarnu matricu, gdje su rubne točke prikazane jednom vrijednošću (npr. 1), a sve ostale suprotnom vrijednošću (npr. 0).



Slika 3.8: Konačna slika gradijenata.

Većini proučenih algoritama upravo je dovoljna sama informacija o tome je li pojedina točka rubna točka ili nije. Najjednostavniji način pretvorbe slike s 256 nijansi sive u binarnu matricu je izrada matrice gradijenata postupkom navedenim u prethodnom pododjeljku te pretvorba svih elemenata matrice iznad nekog praga u vrijednost 1, a ostale u vrijednost 0. Primjena navedenog postupka uz prag od 15% maksimalne vrijednosti svih elemenata dobivene matrice gradijenata prikazana je na slici 3.9.



Slika 3.9: Binarna slika rubnih točaka.

3.2. Algoritmi za detekciju kružnica na slici

U okviru ovog rada proučeno je nekoliko različitih postojećih radova i algoritama za detekciju kružnica na slici i implementano vlastito rješenje. Svi proučeni radovi koriste standardnu, gore navedenu obradu slike, pri kojoj se na kraju dobiva matrica rubnih točaka ili matrica gradijenata.

3.2.1. Houghova transformacija za kružnice

Houghova transformacija za kružnice svakako je najpoznatiji algoritam za detekciju kružnica na slici. Nakon obrade slike pri kojoj se dobiva matrica rubnih točaka, sljedeći korak je pretvorba te matrice u parametarski prostor, gdje kružnica ima tri parametra: koordinate centra kružnice X i Y te radijus R . Time je određeno preslikavanje u trodimenzionalni parametarski prostor. Svaka rubna točka potencijalni je dio neke kružnice. Prolazeći kroz sve rubne točke, algoritam povećava akumulacijsku vrijednost za svaku kombinaciju parametara X , Y i R na čijoj se kružnici nalazi upravo ta točka. Kombinacije parametara koje akumuliraju najviše glasova, najbolji su kandidati za stvarne kružnice obzirom da sadrže najviše pronađenih rubnih točaka. Stoga je posljednji korak ovog algoritma pronalazak lokalnih maksimuma te izdvajanje onih koji su viši od određenog praga. Upravo su ti maksimumi kombinacije parametara koji predstavljaju detektirane kružnice na slici.

Houghova transformacija za kružnice može u isto vrijeme detektirati više kružnica. Jedna od velikih prednosti je robusnost, tj. detekcija kružnica koje su nepotpune ili su pod utjecajem šuma. No, najveći nedostatak Houghove transformacije za kružnice je jako velika složenost i memorijsko zauzeće. Postoji mnoštvo razvijenih algoritama kojima je ideja nadogradnja i poboljšanje ovdje predstavljenog algoritma.

3.2.2. Pronalazak kružnica koristeći parove gradijentnih vektora

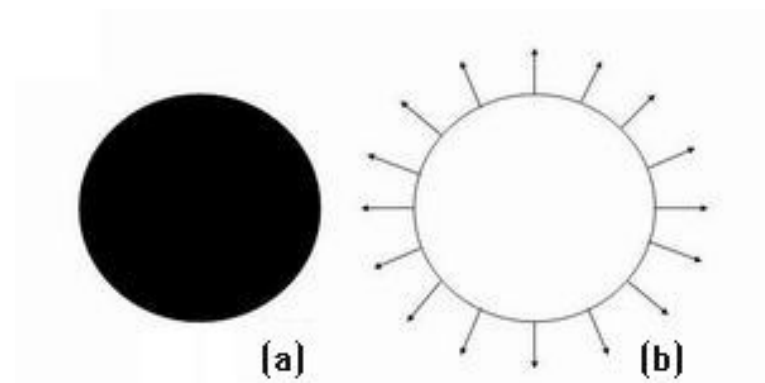
Ovaj algoritam, opisan u radu (Rad et al., 2003), neusporedivo je brži od generalizirane Houghove transformacije za kružnice, no ograničen je na pronalazak kružnih objekata koji su u potpunosti tamniji ili svjetliji od pozadine.

Prvi korak algoritma sastoji se od određivanja gradijentnih vektore koristeći Sobelove operatore. Algoritam određuje duljinu i smjer gradijentnog vektora. Uz pretpostavku da je kružni objekt u potpunosti tamniji od pozadine, vektori rubnih točaka su orijentirani izvan rubne kružnice, tj. iz centra kružnice prema van, što je prikazano na slici 3.10.

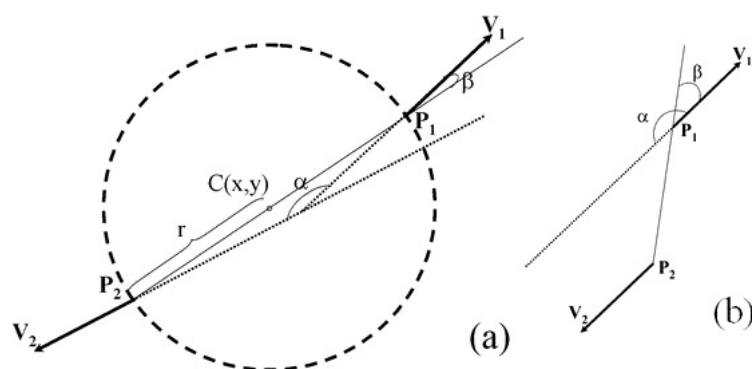
Za svaki vektor V_1 na kružnici postoji njegov par V_2 , koji je suprotne orijentacije te se nalazi na suprotnoj točki rubne kružnice. Taj vektor zadovoljava dva svojstva.

1. Kut α između tih vektora trebao bi biti približno 180° .
2. Kut β između vektora P_2-P_1 te vektora V_1 trebao bi biti približno 0° .

Slika 3.11 prikazuje ta svojstva.



Slika 3.10: (a) Crni krug na bijeloj podlozi, (b) Vektori gradijenata od (a). Preuzeto iz Rad et al. (2003).



Slika 3.11: (a) Par vektora i njihova kružnica (b) vektori odstranjeni drugim pravilom. Preuzeto iz Rad et al. (2003).

Drugi korak algoritma pronalazi sve parove vektora koji zadovoljavaju prethodno navedena svojstva. U trećem koraku svaki par vektora daje glas za kružnicu čiji se centar nalazi na polovici sjecišta dužine koja spaja P_1 i P_2 , a čiji je radijus jednak polovici te dužine. Posljednji, četvrti korak može se izvesti identično prethodno navedenoj Houghovoj transformaciji za kružnice, gdje se pronalaze lokalni maksimumi te oni maksimumi koji su iznad određenog praga predstavljaju kružnice na slici.

Iako brži nego prethodno navedeni algoritam, nedostatak je pretpostavka velikog kontrasta između kružnog objekta i pozadine. Također, pokazuje manju otpornost na nepotpune rubne kružnice nastale šumovima, kao i na preklapanje kružnih objekata.

3.2.3. Umjetni imunološki algoritam za pronalazak kružnica

Postoje brojni radovi koji primjenjuju metaheurističke optimizacijske algoritme za detekciju kružnica na slici. Rad (Cuevas et al., 2012) primjer je rada koji koristi umjetni imunološki algoritam. Ono što ga razlikuje od većine drugih radova jest da omogućuje

detekciju više kružnica u jednom klasičnom pokretanju algoritma, dok većina drugih radova pretpostavlja jedno pokretanje algoritma za pronalazak jedne, potencijalno najbolje kružnice.

Ideja opisana u radu (Cuevas et al., 2012) jest da algoritam tijekom izvođenja ipak na neki način pamti i druga, ne nužno najbolja rješenja, tj. kružnice. Algoritam koristi matricu rubnih točaka te birajući tri točke gradi rješenje izračunavajući parametre kružnice koja prolazi kroz te tri točke. Algoritam ima zadaću pronaći tri točke koje najbolje opisuju kružnicu tj. rješenje s najvećom dobrotom. Dobrota rješenja vrednuje se na sličan način kao i u prethodno navedenim algoritmima, gdje je bolja kružnica ona koja ima više rubnih točaka u matrici rubnih točaka.

U okviru ovog završnog rada osmišljen je i implementiran imunološki algoritam za detekciju kružnica koji je djelomično motiviran algoritmom opisanim u radu (Cuevas et al., 2012). No, za razliku od tamo opisanog pristupa, algoritam razvijen u okviru ovog završnog rada ne koristi matricu rubnih točaka već se temelji na uporabi matrice gradijenata. Budući da se razvijeni algoritam zasniva na imunološkom algoritmu, u sljedećem podpoglavlju opisani su najpoznatiji predstavnici imunoloških algoritama, nakon čega je opisan i algoritam predložen u ovom završnom radu.

3.3. Imunološki algoritmi

Imunološki algoritmi su porodica metaheurističkih algoritama koji spadaju u skupinu prirodom inspiriranih optimizacijskih algoritama (Čupić, 2013). Ova porodica algoritama inspiraciju pronalazi u imunološkom sustavu čovjeka.

Postoji mnogo podvrsta imunoloških algoritama od kojih su najpoznatiji jednostavni imunološki algoritam te algoritam klonske selekcije.

3.3.1. Jednostavni imunološki algoritam

Jednostavni imunološki algoritam je optimizacijski algoritam koji radi s populacijom rješenja koja se nazivaju antitijela. Naziv antitijelo proizlazi iz imunološkog sustava. Sam problem koji algoritam rješava nazivamo antigen. Najčešće je to pronalazak optimuma funkcije. Dobrotu pojedinog rješenja nazivamo njegovim afinitetom te ono u imunološkom smislu predstavlja razinu prilagođenosti, tj. kompatibilnosti antitijela prema antigenu. Pseudokod je prikazan algoritmom 1.

Nakon inicijalizacije i evaluacije početnih rješenja započinje slijed prilagodbe tj. pronalaska antitijela s najvećim afinitetom za zadani antigen. U svakom koraku dolazi

Algoritam 1 Jednostavni imunološki algoritam

```
P := stvoriPočetnuPopulaciju()
evaluiraj(P)
while uvjet do
    Pclo = kloniraj(P, k)
    Phyp = hipermutiraj(P)
    evaluiraj(Phyp)
    P := odaberi(Phyp + P, d)
end while
```

do kloniranja svakoga rješenja k puta te se sva rješenja hipermutiraju kako bi se dobila nova potencijalno bolja rješenja. Evaluacijom se trenutno postojećim rješenjima pridjeljuje dobrota. Nakon evaluacije, od svih mogućih rješenja tog koraka, uzima se d najboljih koja se prenose u sljedeću generaciju, tj. sljedeći korak algoritma.

Kao uvjet zaustavljanja najčešće se koristi određen broj iteracija odnosno generacija algoritma ili razina dobrote koju najbolje rješenje mora postići.

3.3.2. Algoritam klonske selekcije

Algoritam klonske selekcije nasljednik je jednostavnog imunološkog algoritma koji uvodi dodatne operatore i sitne izmjene kako bi se poboljšao rad algoritma. Algoritam 2 prikazuje pseudokod algoritma klonske selekcije.

Algoritam 2 Algoritam klonske selekcije

```
P := stvoriPočetnuPopulaciju()
evaluiraj(P)
while uvjet do
    Pclo = kloniraj(P,  $\beta$ )
    Phyp = hipermutiraj(P)
    evaluiraj(Phyp)
    P := odaberi(Phyp, n)
    Pnovi := generirajNove()
    zamjeniNajlošije(P, Pnovi)
    evaluiraj(P)
end while
```

Nakon početne inicijalizacije i evaluacije rješenja, algoritam ulazi u petlju koja

predstavlja jednu generaciju, tj. iteraciju izvođenja algoritma. Operator kloniranja djeluje malo drugačije nego prethodno navedeni. Umjesto kloniranja svih rješenja u jednakoj mjeri, ovaj operator daje prednost trenutno boljim rješenjima te njih klonira više puta u odnosu na lošija rješenja. Najbolje rješenje klonira se $\lfloor \beta \cdot n \rfloor$ puta, dok se najlošije klonira svega $\lfloor \beta \rfloor$ puta, pri čemu je β pozitivan decimalni broj.

Operator hipermutacije također djeluje u ovisnosti o kvaliteti rješenja, na način da su bolja rješenja manje mijenjana u odnosu na lošija. Ideja je zadržati se u okolini najboljih rješenja te pokušati finijim izmjenama naći još bolja rješenja, a većom izmjenom lošijih rješenja potencijalno naći druga područja s kvalitetnim rješenjima, pa možda i boljim od trenutno najboljih. Od tako izmijenjenih rješenja uzima se n najboljih. Dodatno se d najlošijih rješenja zamjenjuju ponovno stvorenim rješenjima kako bi se na neki način izbjeglo zapinjanje algoritma u lokalnim minimumima.

3.3.3. Algoritam implementiran u radu

Algoritam implementiran u ovome radu bazira se na radu o umjetnom imunološkom algoritmu za detekciju kružnica (Cuevas et al., 2012). No, za razliku od njega koristi matricu vrijednosti gradijenata a ne matricu rubnih točaka. Samim time rješenje nije predstavljeno trima točkama koje opisuju kružnicu, već s X i Y koordinatama centra kružnice te radijusom R . Algoritam 3 prikazuje pseudokod implementiranog algoritma.

Ideja za izmjenu tog bitnog dijela jest da se dobrotu rješenja povećava što je trenutna kružnica sličnija stvarno postojećoj kružnici, tj. rubnoj kružnici kovanice. Naime, nakon obrade slike te dobivanjem matrice gradijenata može se vidjeti da su gradijenti jako maleni na područjima gdje se nalazi podloga, dok su puno veći na područjima gdje se nalaze kovanice. Najveći gradijent nalazi se na samom prijelazu iz kovanice prema podlozi. Time se rješenja grupiraju u području kovanica, a izbjegavaju područje podloge.

Sam algoritam pretpostavlja da na slici postoji barem jedna kovanica te da najbolje rješenje zasigurno predstavlja rubnu kružnicu kovanice. Najbolje trenutno rješenje određuje dozvoljene veličine rubnih kružnica ostalih rješenja te njihovu minimalnu dobrotu.

Algoritam sadrži arhivu svih kružnica detektiranih tijekom rada te jedno pokretanje algoritma može detektirati sve kružnice. Da bi rješenje bilo proglašeno rubnom kružnicom kovanice, ono mora biti najbolje rješenje iz trenutne populacije kroz određen broj generacija algoritma. Ukoliko je ono bolje od svih detektiranih kružnica iz

Algoritam 3 Implementirani imunološki algoritam

```
P := stvoriPočetnuPopulaciju()
evaluiraj(P)
while uvjet do
    Pclo = kloniraj(P, beta)
    Phyp = hipermutiraj(P)
    evaluiraj(Phyp)
    if najbolje rješenje nepromijenjeno kroz X iteracija then
        pokušaj dodati u arhivu
    else
        dodaj u Phyp
    end if
    P := odaberi(Phyp, n)
    Pnovi := generirajNove()
    zamijeniNajlošije(P, Pnovi)
    evaluiraj(P)
    zamijeniNedozvoljeneNovim(P)
    evaluiraj(P)
end while
```

archive ili u skladu s najboljom dosad pronađenom kružnicom iz archive kroz određen broja generacija, algoritam dodaje to rješenje u arhivu detektiranih kružnica. Kružnica je u skladu s najboljom kružnicom iz archive ukoliko je razlika njihovih radijusa manja od 55% radijusa najbolje kružnice iz archive te ako je razlika njihovih dobrota manja od 50% dobrote najbolje kružnice iz archive. Pri kraju svake generacije izbacuju se sva rješenja iz populacije koja su dovoljno slična već detektiranim kružnicama iz archive te se zamjenjuju novim rješenjima. Ukoliko postoje dvije kružnice u arhivi koje su dovoljno slične, iz archive se izbacuje ona s lošijom dobrotom. Dvije kružnice su dovoljno slične ako im je razlika radijusa i udaljenost između njihovih centara manja od 65% radijusa veće kružnice. Time se izbjegava zapinjanje algoritma u već riješene lokalne optimume te proširuje prostor pretrage.

U nastavku je prikazan detaljniji opis korištenih operatora te implementacija evalutora.

Operator generiranja populacije

Operator generiranja populacije stvara novu populaciju veličine n koja sadrži rješenja sa slučajno dodijeljenim vrijednostima X , Y i R , koja su u nekom dozvoljenom intervalu.

Operator selekcije

Operator selekcije prima populaciju rješenja te parametar n . Kao rezultat vraća populaciju od N najboljih rješenja. Ukoliko primljena populacija sadrži manje od n rješenja, vraća sva rješenja.

Operator kloniranja

Operator kloniranja prima populaciju rješenja te u ovisnosti o parametru β stvara novu populaciju kloniranih rješenja. Rješenja s većom dobrotom kloniraju se više puta u odnosu na ona s manjom dobrotom. Preciznije, broj klonova svakog rješenja određen je svojim rangom kvalitete rješenja u trenutnoj populaciji. Broj klonova računa se prema sljedećem izrazu:

$$BrojKlonova = \lfloor (\beta \cdot n) / rang \rfloor$$

gdje je β parametar operatora kloniranja, n veličina predane populacije, a $rang$ rang trenutno promatranog rješenja.

Operator hipermutacije

Operator hipermutacije mutira, tj. mijenja X , Y i R vrijednosti rješenja ovisno o rang. Najbolje rješenje bit će najmanje mutirano, a najlošije rješenje bit će najviše mutirano. Time se omogućuje da algoritam brže i s većom vjerojatnošću pronađe lokalne optimume i time najboljeg kandidata za rubnu kružnicu kovanice.

Za svaku vrijednost X , Y i R zadaje se minimalna i maksimalna vrijednost koju X , Y i R mogu poprimiti. Mutacijom se mijenjaju vrijednosti na način da se najlošije rješenje izmjeni za najviše 30% maksimalnog intervala, a najbolje rješenje izmjeni za najviše 1% maksimalnog intervala jednolikom razdiobom. Maksimalni intervali ovise o dimenzijama slika i pretpostavljenom intervalu radijusa kovanica.

Operator zamjene najlošijih rješenja

Operator zamjene najlošijih rješenja zamjenjuje određen broj najlošijih rješenja u predanoj populaciji nekom drugom populacijom.

Evaluator

Evaluator dodjeljuje rješenju njegovu dobrotu. Za zadane parametre rješenja, evaluator stvara koordinate točaka koje se nalazi na kružnici Bresenhamovim algoritmom za kružnice. Za svaku dobivenu točku, ukupnoj dobroti se dodaje vrijednost gradijenta u toj točki. Ukupna suma gradijenata dijeli se s korijenom radijusa kružnice te se dobiva konačna dobrota rješenja.

Ukoliko se suma gradijenata ne podijeli s korijenom radijusa kružnica, algoritam povremeno detektira jako velike kružnice koje ne odgovaraju stvarnim rubnim kružnicama, a prolaze kroz nekoliko kovanica te imaju veliku ukupnu sumu. Dijeljenjem sume gradijenata s radijusom kružnice, kao mjera se dobiva prosječna vrijednost gradijenta po obodu kružnice. Međutim, tada algoritam povremeno detektira i manje kružnice unutar samih kovanica, što nije željeno ponašanje. Slika 3.12 prikazuje takve neispravno detektirane kružnice. Kako bi izbjegli takvo ponašanje kao mjeru koristimo ukupnu sumu gradijenata podijeljenu s korijenom radijusa kružnice.



Slika 3.12: Neispravno detektirane kružnice.

4. Skup podataka

Skup podataka napravljen je u sklopu izrade ovog rada. Korištene su hrvatske kovanice i to pet različitih vrsta. To su kovanice od 5 kuna, 2 kune, 1 kuna, 50 lipa te 20 lipa. Kako svaka kovanica sadrži dvije strane, time dobivamo 10 različitih slika kovanica.

Pri izradi slika kovanica korišten je skener radi dobivanja visoko kvalitetnih slika. Izrađene slike su slike s 256 nijansi sive uz visoku razlučivosti od 600 DPI. Iako implementirani algoritam radi i sa slikama u boji, korišteni skener stvara kvalitetnije slike s 256 nijansi sive te su one odabrane za izradu konačnog skupa podataka. Time je izbjegnuta potreba za ručnom pretvorbom slike u boji u sliku s 256 nijansi sive. U slučaju da bi se koristile slike u boji, taj bi korak trebalo provesti. Skenirane slike su formata A6 tj. dimenzija 2480×3496 . Ideja je po potrebi skalirati slike za rad algoritma te pronaći idealnu skaliranu dimenziju za rad istog.

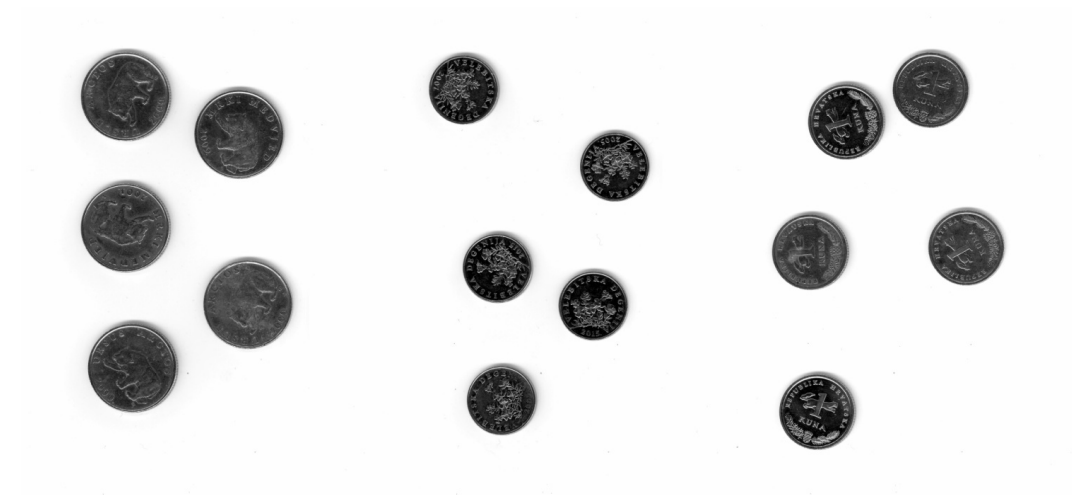
Na ovaj način su izrađena dva skupa podataka: skup podataka za učenje te skup podataka za ispitivanje.

4.1. Skup podataka za učenje

Skup podataka za učenje sastoji se od 30 slika. Na svakoj slici nalazi se 5 istovrsnih kovanica skeniranih s iste strane. Razlog tome je olakšano ispitivanje rada implementiranog algoritma za pronalazak rubnih kružnica kovanica. Svake vrste kovanica ima 15 različitih primjeraka. Ukupno se na slikama nalazi 75 različitih kovanica. Svaka kovanica slikana je s obje strane jednom te se stoga na slikama nalazi točno 150 različitih strana kovanica. Na slikama nema preklapanja kovanica. Tri primjera slika iz skupa podataka za učenje prikazana su na slici 4.1.

4.2. Skup podataka za ispitivanje

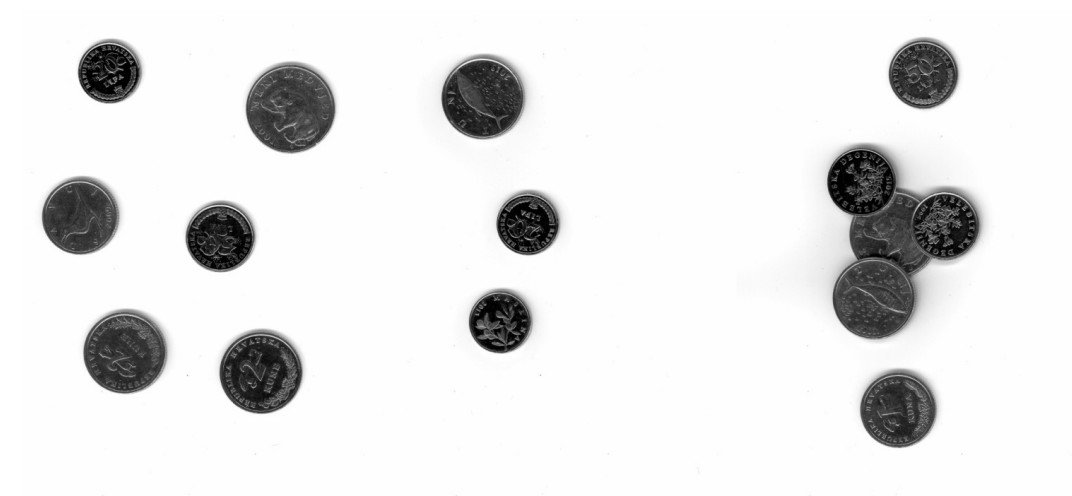
U skupu podataka za ispitivanje nalazi se ukupno 18 slika. Svake vrste kovanica ima 5 različitih primjeraka. Na slikama se nalazi ukupno 25 različitih kovanica te 50 različi-



Slika 4.1: Tri primjera slika iz skupa podataka za učenje.

tih strana kovanica.

10 slika napravljeno je tako da se kovanice ne preklapaju, dok 8 slika sadrži kovanice u kojima se barem neke preklapaju. Na svakoj slici nalazi se barem 1 kovanica, a najviše njih 6. Tri primjera slika iz skupa podataka za ispitivanje prikazana su na slici 4.2.



Slika 4.2: Tri primjera slika iz skupa podataka za ispitivanje.

4.3. Skup podataka za rad klasifikatora

Implementirani algoritam za detekciju rubnih kružnica kovanica pokrenut je na svim prethodno navedenim slikama. Detekcijom kružnica te izrezivanjem kružnica sa svih slika na kojima nema preklapanja napravljen je skup podataka za rad klasifikatora.

Time je nastalo 150 različitih slika pojedinačnih kovanica kao skup podataka za učenje klasifikatora te 40 različitih pojedinačnih slika kovanica za ispitivanje ispravnosti rada klasifikatora. Neke od njih prikazane su na slici 4.3.



Slika 4.3: Tri primjera slika iz skupa podataka za rad klasifikatora.

5. Klasifikacija

Posljednji korak ovog rada sastoji se od klasifikacije kovanica. Za klasifikaciju se koristi skup podataka za rad klasifikatora koji je stvoren prethodno implementiranim modificiranim imunološkim algoritmom.

5.1. Izlučivanje značajki

Značajke su svojstva objekata koja ih razlikuju od ostalih objekata te na temelju njih klasifikator može raspoznavati objekte.

Problem kod klasifikacije kovanica jest mogućnost rotacije kovanica na slici. Ljudi raspoznaju kovanice neovisno o rotaciji automatskom detekcijom njihovih bitnih značajki: bilo prepoznavanjem brojeva koji označavaju vrijednost kovanice, bilo prepoznavanjem likova koji se nalaze na jednoj od strana kovanice.

5.2. Izrada novog skupa podataka

Jedno od mogućih rješenja ovoga problema jest izrada dodatnog skupa podataka koji bi od svake pojedinačne slike kovanica napravio nekoliko slika zarotiranih kovanica. Npr. od 1 slike napraviti 72 nove slike, pri čemu bi svaka slika bila rotirana u odnosu na prethodnu za 5° . Takav skup podataka mogao bi biti korišten za učenje klasifikatora. Najveći problem ovdje je veliko povećanje skupa podataka za rad klasifikatora te bi sam klasifikator morao naučiti raditi sa 72 puta više, njemu različitih, slika.

5.3. Značajke neovisne o rotaciji

Pronalaženje značajki neovisnih o rotaciji djeluje kao prihvatljivije rješenje od prethodno navedenog. U ovom radu pokušana je podjela slika kovanica na segmente iz kojih bi se izvukle određene značajke koje bi barem u nekoj mjeri bile otporne na

rotaciju.

Prva ideja bila je podjela slike u segmente koji izgledaju kao tortni grafikoni. Odmah se pojavio problem pronalaska broja segmenata iz kojih bi se trebale izvući značajke kako bi barem u nekoj mjeri postojala otpornost na rotaciju. Pored toga, iz dobivenih značajki trebalo bi ponovno pronaći značajke neovisne o rotaciji. Prikaz podjele na prethodno navedene segmente prikazan je na slici 5.1.



Slika 5.1: Podjela na segmente poput tortnog grafikona.

Iako prethodna ideja nije praktična, poslužila je kao inspiracija za smisleniju ideju. Podjela slika kovanica na segmente u obliku diska. Ideja je da u kojem god smjeru se kovanica zarotira, iste točke kovanice trebale bi se uvijek nalaziti u istom disku. Potrebno je pronaći kvalitetne značajke koje bi opisivale svaki od diskova te se time dobiva niz značajki neovisnih o rotaciji. Prikaz podjele na segmente u obliku diska prikazan je na slici 5.2.



Slika 5.2: Podjela na diskove.

U ovom radu pokušano je svakom takvom disku dodijeliti vrijednosti intenziteta njegovih točaka sa slike. Također je pokušano i s dodjelom gradijentnih vrijednosti. Oba implementirana načina su primijenjena te ispitana u radu s umjetnom neuronskom mrežom kao klasifikatorom. Značajke nisu pokazala dovoljnu specifičnost vezanu uz vrstu kovanice. Sve vrijednosti bile su vrlo slične te ih klasifikator nije mogao iskoristiti za klasifikaciju. Tablica 5.1 prikazuje matricu zabune tako implementiranog klasifikatora primijenjenog na skup podataka za učenje klasifikatora. Prikazani klasifikator koristi dodjelu gradijentnih vrijednosti svakom disku. Uspješno je klasificirano samo 52 od ukupno 150 kovanica.

Tablica 5.1: Matrica zabune umjetne neuronske mreže na skupu podataka za učenje klasifikatora.

prepoznato stvarno	20 lp	50 lp	1 kn	2 kn	5 kn
20 lp	21	5	0	4	0
50 lp	15	7	0	7	1
1 kn	2	6	0	15	7
2 kn	2	4	0	13	11
5 kn	0	1	0	18	11

5.4. Implementirani klasifikator

Zbog poteškoća u pronalasku kvalitetnih značajki, implementiran je jednostavan klasifikator koji iz slika kovanica izvlači njihov radijus. Radijus svih kovanica iste vrste se uprosječuje te se proglašava radijusom te kovanice. Klasifikator na temelju tih vrijednosti raspoznaje vrstu kovanice.

Ovakav klasifikator ponaša se dovoljno dobro za ovakve skupove podataka gdje se svaka vrsta kovanica dovoljno razlikuje svojim radijusom. Klasifikator ovisi o samom radu algoritma za detekciju kovanica koji zbog šumova na slici ili same sjene kovanice može pokazivati malo odstupanje. Klasifikator očekuje da je svaka slika skenirana istom rezolucijom, tj. da su veličine istih vrsta kovanica jednake na svim korištenim slikama. Ovakav klasifikator najčešće nije primjenjiv na druge skupove s drugim vrstama kovanica ili slike skenirane drugim rezolucijama.

6. Rezultati

U ovom poglavlju opisani su rezultati dobiveni radom detektora i klasifikatora kovanica. Detektor kovanica ispravno je detektirao svih 150 kovanica iz skupa podataka za učenje te iz njih uspješno napravio novi skup podataka koji sadrži pojedinačne slike. Takve slike mogu se neovisno iskoristiti. Na skupu podataka za ispitivanje nisu detektirane 2 kovanice od ukupno 80. Te 2 kovanice nalaze se ispod drugih kovanica te nemaju dovoljno dug vidljivi opseg rubnih kružnica. Preciznost i osjetljivost detektora prikazana je u tablici 6.1. Slika 6.1 prikazuje ispravno detektirane te izrezane kovanice.

Tablica 6.1: Preciznost i osjetljivost detektora.

Skup podataka	Preciznost	Osjetljivost
za učenje	100%	100%
za ispitivanje	100%	97.5%



Slika 6.1: Primjer ispravno detektiranih kovanica.

Kod nekoliko kovanica, iako ispravno detektiranih, postoji malo odstupanje pri čemu stvarna rubna kružnica kovanica odstupa od detektiranih. Pri izradi novog skupa podataka dobivamo slike koje izgledaju kao da im je odsječen rub. Nekoliko takvih primjera prikazano je na slici 6.2. To odstupanje toliko je maleno da utjecaj na rad kvalitetnijeg klasifikatora ne bi trebao postojati. Slika 6.3 prikazuje ispravno detektirane i klasificirane kovanice na ciljnim slikama.



Slika 6.2: Primjer detektiranih kovanica s odstupanjem.



Slika 6.3: Primjer uspješnog raspoznavanja kovanica.

Rad implementiranog klasifikatora ipak je ograničen kvalitetom detektora kovanica. Zbog prejake sjene postoji mogućnost detektiranja prevelike kružnice pa je radijus takve kružnice bliži radijusu neke druge kovanice.

Sve kovanice iz skupa podataka za učenje ispravno su klasificirane. Od 78 kovanice iz skupa podataka za ispitivanje, njih 5 nije ispravno klasificirano. 4 neispravno klasificiranih kovanica nalaze se na slikama s preklapanjem kovanica. Tablice 6.2 i 6.3 prikazuju matrice zabune implementiranog klasifikatora. Primjer neispravno detektiranih te pogrešno klasificiranih kovanica prikazan je na slici 6.4.

Tablica 6.2: Matrica zabune klasifikatora za uniju skupova podataka.

prepoznato stvarno	20 lp	50 lp	1 kn	2 kn	5 kn
20 lp	46	0	0	0	0
50 lp	3	42	0	0	0
1 kn	0	0	45	0	0
2 kn	0	0	0	44	1
5 kn	0	0	0	0	47

Tablica 6.3: Matrica zabune klasifikatora za skup podataka za ispitivanje rada klasifikatora.

prepoznato stvarno	20 lp	50 lp	1 kn	2 kn	5 kn
20 lp	16	0	0	0	0
50 lp	3	12	0	0	0
1 kn	0	0	15	0	0
2 kn	0	0	0	14	1
5 kn	0	0	0	0	17



Slika 6.4: Primjer neuspješnog raspoznavanja kovanica.

7. Zaključak

Raspoznavanje uzoraka složen je postupak. U radu je implementirano rješenje za raspoznavanje kovanica sa slika. Implementirani algoritmi nadahnuti su proučenim radovima, no samostalno osmišljeni.

Detektor kovanica radi ispravno za sve slike na kojima nema preklapanja. Veliko preklapanje može uzrokovati neuspjeh u detektiranju pojedinih kovanica. Ovaj problem mogao bi se riješiti izradom evaluatora otpornijeg na preklapanja, pri čemu bi dobrota rješenja ovisila i o duljini vidljivog ruba kovanica. Detektor pokazuje odlične rezultate te se može koristiti neovisno o samom klasifikatoru.

Implementirani klasifikator u velikoj mjeri ovisan je o radu detektora. Njegov rad u potpunosti ovisi o ispravnosti i preciznosti detektiranja kovanica. Ovaj klasifikator nije primjenjiv na širi skup kovanica. Za svako ozbiljnije rješenje trebalo bi implementirati kvalitetniji klasifikator.

Izrađeni skup podataka za rad klasifikatora može se koristiti neovisno o ostalim dijelovima ovog rada.

LITERATURA

Erik Cuevas, Valentín Osuna-Enciso, Fernando Wario, Daniel Zaldívar, i Marco Pérez-Cisneros. Automatic multiple circle detection based on artificial immune systems. *Expert Systems with Applications*, 39(1):713–722, 2012.

Tanner Helland. Seven grayscale conversion algorithms (with pseudocode and vb6 source code), 2011. URL <http://www.tannerhelland.com/3643/grayscale-image-algorithm-vb6/>.

Ali Ajdari Rad, Karim Faez, i Navid Qaragozlou. Fast circle detection using gradient pair vectors. U *DICTA*, stranice 879–888, 2003.

Wikipedia. Colorimetric (luminance-preserving) conversion to grayscale, 2016. URL https://en.wikipedia.org/wiki/Grayscale#Converting_color_to_grayscale.

Marko Čupić. Prirodom inspirirani optimizacijski algoritmi. metaheuristike., 2013.

Raspoznavanje kovanica sa slika

Sažetak

Problem raspoznavanja kovanica sa slika sastoji se od dva manja problema: detekcije kovanica te njihove klasifikacije. Veći dio ovog rada posvećen je detekciji rubnih kružnica kovanica. Houghova transformacija za kružnice svakako je najpoznatiji algoritam za pronalazak kružnica na slici. No, najveći nedostatak Houghove transformacije za kružnice je jako velika složenost i memorijsko zauzeće. Ovaj rad proučava primjenu modificiranog umjetnog imunološkog algoritma koji ima mogućnost detekcije više kružnica pri jednom pokretanju. U sklopu rada izrađen je skup podataka od ukupno 200 kovanica s 5 različitih vrijednosti. U radu je predstavljan primjer jednostavnog klasifikatora koji kao jedinu značajku koristi radijus kovanica.

Ključne riječi: raspoznavanje uzoraka, računalni vid, detekcija kružnica, umjetni imunološki algoritam, algoritam klonske selekcije

Coin Recognition from Images

Abstract

The problem of coin recognition from images consists of two smaller problems: coin detection and coin classification. The main focus of this thesis is coin detection. Hough circle transform has been the most common method for detecting circles on images. The biggest drawbacks are huge computational load and large memory usage. This thesis examines usage of a modified artificial immune system to detect multiple circles at once. A dataset of 200 coins with 5 different values was made. A simple classifier was presented. The classifier uses only the radius as a single feature.

Keywords: pattern recognition, computer vision, circle detection, artificial immune system, clonal selection algorithm