Uspoređivanje dinamičnih slika na osnovi karakteristika objekta

Lučić, Luka

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni diplomski studij Elektrotehnika

USPOREĐIVANJE DINAMIČKIH SLIKA NA OSNOVU KARAKTERISTIKA OBJEKATA

Diplomski rad

Luka Lučić

Osijek, 2024.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac D1: Obrazac za ocjenu diplomskog rada na sveučilišnom diplomskom studiju

Ocjena diplomskog rada na sveučilišnom diplomskom studiju

Ime i prezime pristupnika:	Luka Lučić
Studij, smjer:	Sveučilišni diplomski studij Elektrotehnika, Komunikacije i
Mat. br. pristupnika, god.	D-1485, 07.10.2022.
JMBAG:	0165077678
Mentor:	prof. dr. sc. Mario Vranješ
Sumentor:	
Sumentor iz tvrtke:	Zvonimir Kaprocki
Predsjednik Povjerenstva:	doc. dr. sc. Denis Vranješ
Član Povjerenstva 1:	prof. dr. sc. Mario Vranješ
Član Povjerenstva 2:	izv. prof. dr. sc. Ratko Grbić
Naslov diplomskog rada:	Uspoređivanje dinamičnih slika na osnovi karakteristika objekta
Znanstvena grana diplomskog rada:	**Obradba informacija (zn. polje računarstvo)**
Zadatak diplomskog rada:	Moderne aplikacije koriste animacije i efekte kako bi učinile svoj sadržaj zanimljivijim i ugodnijim oku korisnika. Testiranje funkcionalnosti takvih aplikacija vrši se uspoređivanjem slika preuzetih s testne verzije aplikacije s referentnim slikama ispravne aplikacije (black box testing, BBT). Zbog animacija i raznih efekata, sadržaj slike ovisi o trenutku u kojem je preuzeta, tj. slike preuzete iz aplikacije su dinamične slike. Dakle, testiranje funkcionalnosti svodi se na uspoređivanje dinamičnih slika. Postoji mnogo potencijalnih metoda koje bi mogle pomoći kod uspoređivanja
Datum ocjene pismenog dijela diplomskog rada od strane mentora:	16.09.2024.
Ocjena pismenog dijela diplomskog rada od strane mentora:	Izvrstan (5)
Datum obrane diplomskog rada:	01.10.2024.
Ocjena usmenog dijela diplomskog rada (obrane):	Izvrstan (5)
Ukupna ocjena diplomskog rada:	Izvrstan (5)
Datum potvrde mentora o predaji konačne verzije diplomskog rada čime je pristupnik završio sveučilišni diplomski studij:	01.10.2024.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O IZVORNOSTI RADA

Osijek, 01.10.2024.

Ime i prezime Pristupnika:	Luka Lučić
Studij:	Sveučilišni diplomski studij Elektrotehnika, Komunikacije i informatika
Mat. br. Pristupnika, godina upisa:	D-1485, 07.10.2022.
Turnitin podudaranje [%]:	9

Ovom izjavom izjavljujem da je rad pod nazivom**: Uspoređivanje dinamičnih slika na osnovi karakteristika objekta**

izrađen pod vodstvom mentora prof. dr. sc. Mario Vranješ

i sumentora

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis pristupnika:

SADRŽAJ

1. UVOD

Usporedba slika predstavlja ključan izazov u području računalnog vida i računalne grafike, s brojnim primjenama u industrijama poput digitalne forenzike, automatiziranog testiranja softvera, nadzornih sustava i analize video sadržaja. Posebno su zahtjevne dinamičke slike, koje se mijenjaju tijekom vremena, bilo kroz promjene u sadržaju (poput kretanja objekata) ili promjene u kontekstu (poput promjena u perspektivi ili uvjetima osvjetljenja). Ova dinamička priroda otežava preciznu usporedbu takvih slika, jer tradicionalne metode možda ne uzimaju u obzir ove vremenske ili kontekstualne promjene. DISM algoritam (engl. *Dynamic Image Similarity Matching*) jedno je od rješenja za usporedbu dinamičkih slika, koje se zasniva na analizi ključnih točaka (KT) na slikama. Ovaj pristup omogućuje prepoznavanje sličnosti između slika unatoč transformacijama objekata poput rotacije, skaliranja, translacije i okluzije, tj. dodavanja novih animiranih čestica u sadržaj koje vizualno predstavljaju smetnju na slici. Međutim, oslanjanje isključivo na analizu ključnih točaka može biti nedovoljno u složenijim situacijama. Tada je potrebna dodatna obrada pomoću dodatnih algoritmima kojima će se dobiti potrebni podaci koji će služiti za dodatnu odluku kod usporedbe dviju dinamičkih slika, kako bi se osigurala visoka točnost prepoznavanja.

Cilj ovog diplomskog rada bio je unaprijediti DISM algoritam dodavanjem novih metoda, uključujući analizu kontura i boje objekata, uz postojeću analizu ključnih točaka. Ovaj prošireni pristup ima za cilj povećati točnost i pouzdanost algoritma u različitim scenarijima, posebno onima u kojima ključne točke same po sebi nisu dovoljne za preciznu usporedbu dviju dinamičkih slika. Rad detaljno opisuje postupke razvoja i implementacije unaprijeđenog algoritma, kao i eksperimentalne postupke za njegovu verifikaciju na dvjema specifičnim bazama slika: CGIAD (engl. *Computer Generated Image Alteration Database*) i CG-STB-DI (engl. *Custom Generated STB Dynamic Images*). Dobiveni rezultati uspoređeni su s rezultatima originalnog DISM algoritma kako bi se evaluirala učinkovitost i prednosti unaprijeđenog pristupa. Ovaj rad ne samo da demonstrira učinkovitost unaprijeđenog algoritma, već također postavlja temelje za daljnje istraživanje i razvoj u području usporedbe dinamičkih slika, s posebnim naglaskom na integraciju različitih analiza i optimizaciju algoritama za primjenu u stvarnim uvjetima.

Sljedeća poglavlja rada bavit će se opisom metodologije, postojećih rješenja i unapređenja algoritma za usporedbu dinamičkih slika. U drugom poglavlju prikazane su ključne metode za detekciju i usporedbu objekata na slikama, s posebnim naglaskom na detekciju ključnih točaka te usporedbu kontura i boja. Treće poglavlje donosi pregled relevantnih radova i istraživanja u

području usporedbe dinamičkih slika te su analizirane prednosti i nedostaci pojedinih rješenja. U četvrtom poglavlju opisan je razvoj i nadogradnja DISM algoritma, uključujući uvođenje dodatnih metoda poput filtriranja ključnih točaka i usporedbe boja i kontura. Peto poglavlje predstavlja eksperimentalni postupak verifikacije i analizu rezultata nadograđenog algoritma te usporedbu dobivenih rezultata s rezultatima algoritma zasnovanog na sijamskoj neuronskoj mreži i originalnog DISM algoritma.

2. PREGLED METODA ZA DETEKCIJU I USPOREDBU OBJEKATA NA SLIKAMA

U ovom će se poglavlju razmotriti postojeće tradicionalne metode računalnog vida za detekciju i usporedbu objekata na slikama. Cilj je pružiti cjeloviti uvid u različite pristupe koji se koriste za obradu slika, naglašavajući prednosti i nedostatke određenih metoda. U suvremenom računalnom vidu, detekcija i usporedba objekata na slikama ima ključnu ulogu u brojnim aplikacijama, kao što su prepoznavanje lica, praćenje objekata, detekcija anomalija i analiza scene. Pomoću ovih metoda omogućuje se računalima da „vide“ i razumiju svijet na način sličan ljudskom, čime se omogućuje širok spektar naprednih funkcionalnosti.

2.1. Algoritmi za detekciju ključnih točaka

Detekcija ključnih točaka srž je mnogih zadataka u području računalnog vida, kao što su podudaranje/poravnavanje slike, identifikacija objekta i praćenje kretanja. Ključna točka tada se može definirati kao značajka slike na koju ne utječu neke operacije kao što su rotacija, skaliranje i promjene u osvjetljenju slike i stoga se može lako identificirati. Te se točke obično koriste kao sidrišne točke koje omogućuju usporedbu slika i identifikaciju promjena između referentne slike i slike koja sadrži te promjene.

2.1.1. SIFT (Scale-Invariant Feature Transform)

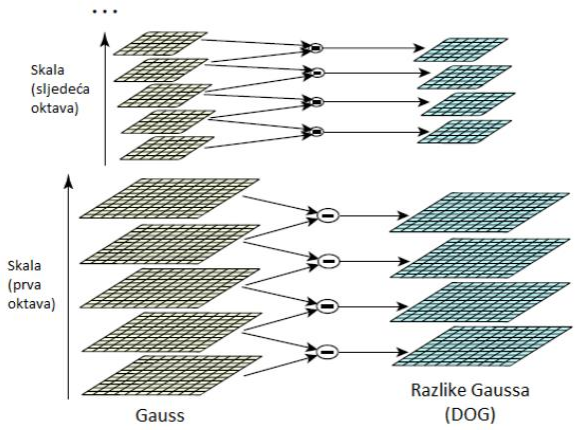
SIFT algoritam Davida G. Lowea iz 2004. smatra se jednim od najuspješnijih pristupa identificiranju ključnih točaka i generiranju deskriptora. Proces detekcije pomoću SIFT algoritma uključuje četiri glavne faze: detekciju ekstrema u prostornoj skali, lokalizaciju ključnih točaka, orijentaciju prostorne skale i generiranje deskriptora. Sve ove faze su neophodne kako bi detektirane ključne točke bile nepromjenjive u odnosu na mjerilo i rotaciju objekta.

Prvi korak SIFT algoritma je detekcija točaka interesa, odnosno ekstrema u prostornoj skali koristeći piramidu Gaussovih razlika, odnosno DOG piramidu (engl. *Difference of Gaussian pyramid*) prikazane na slici 2.1. DOG piramida se sastoji od više skala koje su grupirane u oktave, a svaka oktava sadrži slike s različitim intenzitetima Gaussovog zamućenja. Funkcija skale definirana je kao[1]:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \tag{2-1}$$

gdje je $G(x, y, \sigma)$ Gaussova funkcija, a $I(x, y)$ originalna slika. Razlika između dviju slika susjednih skala odvojenih konstantnim faktorom *k* u piramidi prikazana je formulom:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{2-2}$$



Slika 2.1. *Prikaz određivanja DOG piramide* [1]

Nakon detekcije ekstrema, potencijalnih ključnih točaka, neke od njih je potrebno izbaciti jer leže uz rub ili nemaju dovoljno kontrasta. Za to se koristi Taylorov red funkcije $D(x, y, \sigma)$ kako bi se odredila pozicija ekstrema[1]:

$$D(x) = D + \frac{\partial D^T}{\partial x^2} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \tag{2-3}$$

Ako je pomak $\hat{x}$ veći od 0,5 u bilo kojoj dimenziji, koji se određuje uzimanjem derivacije prethodne funkcije u odnosu na *x* i postavlja na nulu, ključna točka se odbacuje, a rubne točke se eliminiraju pomoću Hessainove matrice *H*[1]:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix} \tag{2-4}$$

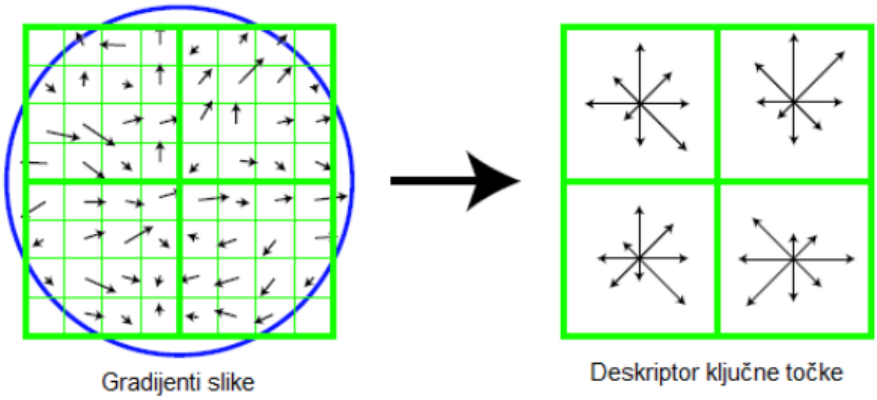
Dalje se određuje orijentacija svake ključne točke čije se provođenje zasniva na gradijentu piksela oko dane ključne točke kako bi se eliminirala ovisnost o orijentaciji. Koriste se sljedeće formule za računanje gradijenata magnitude i orijentacije[1]:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \tag{2-5}$$

$$\theta(x, y) = atan2(L(x, y+1) - L(x, y-1), L(x+1, y) - L(x-1, y)) \tag{2-6}$$

Konačno, kako bi se stvorio deskriptor, generira se histogram orijentacije gradijenata u području koje okružuje ključnu točku, stoga je ključna točka opisana na jednostavan i pouzdan način. Oblikovanje se zasniva na lokalnim gradijentima u okolici ključne točke. Područje oko ključne točke se dijeli na 16 pod-blokova od 4x4 piksela, prikazano na slici 2.2., a za svaki pod-

blok se stvara histogram s 8 binova. Deskriptor je vektor duljine 128 elemenata, koji se normalizira kako bi se osigurala robusnost na promjene osvjetljenja i perspektive.



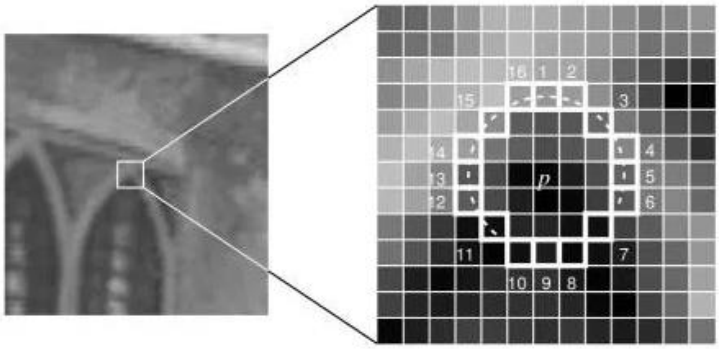
Slika 2.2. *Određivanje deskriptora iz gradijenata slike* [1]

SIFT algoritam je revolucionirao područje računalnog vida pružajući efikasan i pouzdan način za detekciju i opisivanje ključnih točaka u slikama. Njegova sposobnost da prepozna ključne točke neovisno o skali, rotaciji i osvjetljenju čini ga izuzetno korisnim u raznim aplikacijama koje zahtijevaju prepoznavanje i analizu vizualnih informacija. Ovaj algoritam i dalje predstavlja osnovu mnogih naprednih tehnika i istraživanja u području računalnog vida.

2.1.2. ORB (Oriented FAST and Rotated BRIEF)

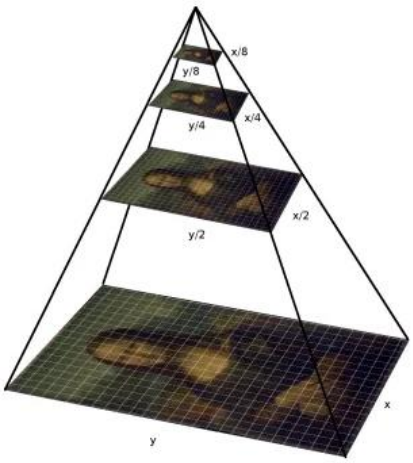
ORB, Oriented FAST (engl. *Features from Accelerated Segmentation Test*) and Rotated BRIEF (engl. *BRIEF - Binary Robust Independent Elementary Features)*, jedan je od najnaprednijih algoritama koji može otkriti ključne točke i proizvesti deskriptore. Razvijene su brže, učinkovitije alternative SIFT-u, a možda je jedno od najvećih otkrića ORB. ORB je posebno popularan u aplikacijama u stvarnom vremenu gdje je brzina kritična zbog svoje brzine i učinkovitosti.

ORB algoritam kombinira dva postojeća algoritma, FAST i BRIEF, definirani u nastavku teksta, kako bi uravnotežio brzinu i točnost. Prva komponenta ORB algoritma je oFAST, usmjereni FAST. FAST algoritam, koji je skraćenica za značajke iz ubrzanog testa segmenta, koristi se za brzo identificiranje ključnih točaka. Radi tako da analizira lokalne intenzitete oko piksela ***p*** s njegovih 16 susjeda koji su raspoređeni u malom krugu oko ***p***, kako je prikazano na slici 2.3. Ako je više od 8 susjednih piksela tamnije ili svjetile od ***p***, onda se ***p*** odabire kao ključna točka.



Slika 2.3. *Prikaz određivanja ključne točke FAST algoritmom* [2]

FAST značajke nemaju orijentacijsku komponentu i višeskalarne značajke. ORB koristi piramidu slike u više skala, sekvenca slika od kojih su sve verzije izvorne slike u različitim rezolucijama, što je vidljivo na slici 2.4. Svaka razina sadrži dvostruko manju sliku po visini i širini od slike iz prethodne razine. Nakon stvaranja piramide, tada se koristi FAST za određivanje ključnih točaka. Na ovaj način je ORB djelomično invarijantan na skalu.



Slika 2.4. *Prikaz piramide slika* [2]

Nakon što se otkriju ključne točke, one se naknadno obrađuju kako bi se dobila orijentacija, što omogućuje precizno i stabilno određivanje orijentacije ključne točke. Zasniva se na težištu intenziteta unutar definirane regije oko ključne točke. Prvo se definiraju momenti zakrpe (engl. *patch*)[2]:

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \tag{2-7}$$

gdje $I(x, y)$ predstavlja intenzitet piksela na (x,y) koordinatama, a p i q su eksponenti čiji rasponi i vrijednosti definiraju specifične informacije koje se dobivaju o zakrpi. Parametar p predstavlja eksponent za x-koordinate, a parametar q predstavlja eksponent za y-koordinate. Oni su mali cijeli brojevi, a njihove vrijednosti se najčešće uzimaju iz skupa $\{0, 1, 2\}$. Na osnovu momenata zakrpa izračunava se centroid, tj. „težište" zakrpe, koje se koristi za dobivanje orijentacije zakrpe[2]:

$$C_x = \frac{m_{10}}{m_{00}}, \quad C_y = \frac{m_{01}}{m_{00}} \tag{2-8}$$

Vektor od središta *O* do težišta *C* daje orijentaciju zakrpe[2]:

$$\theta = atan2(C_y - O_{y)}, C_x - O_x) \tag{2-9}$$

Druga komponenta ORB algoritma je rBRIEF (engl. *Rotated BRIEF*). Izvorni BRIEF deskriptor ima ograničenje koje se odnosi na otpor rotaciji slike. Kako bi se to prevladalo, rBRIEF proširuje osnovni BRIEF deskriptor i dodaje rotacijsku invarijantnost. Naime, originalni BRIEF deskriptor se rotira na temelju orijentacije ključne točke određene u oFAST koraku. Ovo proširenje omogućuje rBRIEF-u da bude robustan i otporan na rotaciju slike, što mu daje prednost u aplikacijama gdje se slike mogu rotirati ili gledati iz različitih perspektiva.

BRIEF generira binarne testove intenziteta između parova piksela u definiranoj regiji oko ključne točke za stvaranje kompaktnog i učinkovitog deskriptora. Deskriptor je niz binarnih vrijednosti. Svaki binarni test definiran je kao[2]:

$$\tau(p; x, y) = \begin{cases} 1 & : p(x) < p(y) \\ 0 & : p(x) \geq p(y) \end{cases} \tag{2-10}$$

gdje $p(x)$ predstavlja intenzitet promatrane regije p u točki x. Značajka je definirana kao vektor s n binarnih testova[2]:

$$f(n) = \sum_{1<i<n} 2^{i-1} \tau(p; x_i, y_i) \tag{2-11}$$

Rotacija BRIEF deskriptora prema orijentaciji ključne točke postiže se korištenjem rotacijske matrice R_θ[2]:

$$S_\theta = R_\theta S \tag{2-12}$$

gdje *S* matrica sadrži koordinate piksela korištenih za binarne testove. Tada ORB diskretizira kut na inkremente od 12° te time čini predefiniranu tablicu s BRIEF uzorcima za svaku rotaciju.

Kombinacija oFAST i BRIEF u ORB-u pruža cjelovito rješenje za određivanje ključnih točaka i izradu deskriptora koji nisu samo brzi, već i učinkoviti u svim uvjetima. ORB može otkriti

i opisati ključne točke na slici, dok je otporan na rotacije i promjene svjetline. Ovo svojstvo čini ga vrlo privlačnim u aplikacijama u stvarnom vremenu, kao što su robotska navigacija, proširena stvarnost i praćenje objekata.

ORB je značajno poboljšanje u odnosu na svoje prethodnike, omogućujući bržu i učinkovitiju obradu slike bez značajnog gubitka točnosti. ORB je brz i robustan, što ga čini popularnim alatom u mnogim modernim aplikacijama računalnog vida koje naglašavaju performanse i točnost u stvarnom vremenu.

Osim SIFT-a i ORB-a, često korišteni algoritmi za detekciju ključnih točaka uključuju DoG (engl. *Difference of Gaussian*), SURF (engl. *Speeded Up Robust Features*) i HoGs (engl. *Histogram of Oriented Gradients*). Svaki od ovih algoritama ima svoje specifične prednosti i primjene u različitim zadacima računalnog vida, poput prepoznavanja objekata, poravnanja slika i analize scena[19].

2.2. Algoritmi za usporedbu kontura objekata

Usporedba kontura objekata je metoda koja se koristi za prepoznavanje i usporedbu oblika objekata na slikama. Konture predstavljaju rubove objekata i mogu se koristiti za razlikovanje objekata jedni od drugih.

2.2.1. Metode za detekciju rubova

Jedna od najvažnijih tehnika u obradi slike je otkrivanje rubova, otkrivanje rubnih točaka objekata ili graničnih piksela. Rubovi su u biti točke na kojima postoji oštra promjena intenziteta piksela i oni prenose važne informacije vezane uz strukturu objekata kao i njihove granice. Koncept detekcije rubova bio je osnovni građevni blok u ovim domenama i intenzivno se koristi u aplikacijama kao što su prepoznavanje objekata, analiza scene, sustavi za navođenje robota i također u domeni medicinskih slika. Canny i Sobel algoritmi jedni su od najčešćih metoda za detekciju rubova, razlikuju se u pogledu složenosti i točnosti.

Canny algoritam jedan je od najuspješnijih algoritama za otkrivanje rubova jer ima proces u više faza. Filtriranje slike Gaussovim filtrom prije primjene značajke *Canny Edge Detection* izglađuje sliku, što smanjuje šum i pomaže u stabilnom procesu detekcije rubova. Gaussovo zamućenje se primjenjuje pomoću kernela, a matematički se može opisati kao[3]:

$$G(x,y) = \frac{1}{2\pi\sigma^2} exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \tag{2-13}$$

$G(x, y)$ je Gaussova funkcija koja se koristi za definiranje Gaussovog kernela, u njoj *x* i *y* predstavljaju koordinate piksela u dvodimenzionalnom prostoru kernela, a σ određuje količinu „zamućenja“ filtra. Filtrirana slika se zatim koristi za pronalaženje gradijenta slike, koji može otkriti promjene kontrasta koje se događaju u objektu koristeći bilo što (u ovom slučaju Sobel kernel). Gradijent magnituda *G* računa se kao[3]:

$$G = \sqrt{G_x^2 + G_y^2} \tag{2-14}$$

$G_x \; i \; G_y$ predstavljaju horizontalnu i vertikalnu komponentu gradijenta slike dobivene primjenom Sobelovog operatora u horizontalnom i vertikalnom smjeru. Sam izračun navedenih komponenti prikazan je formulom (2-15). Izračunava se gradijent, a zatim se primjenjuje tehnika koja se zove potiskivanje ne-maksimalnih vrijednosti (engl. *nonmaxima suppression*) koja stanji detektirane rubove na liniju široku jedan piksel. Ovo je jako važno jer na taj način se briše puno rubova koji su nasumično detektirani jer je fokus usmjeren na druge rubove. Posljednji korak Cannyjevog algoritma je primjena praga s histerezom, koji koristi dva praga za razlikovanje pravih rubova od lažnih rubova. Niži prag otkriva moguće rubove, dok se viši primjenjuje za pronalaženje (pravih) rubnih piksela, što omogućuje robusnu i pouzdanu detekciju.

Sobelov algoritam također čini otkrivanje rubova jednostavnijim. Ovaj algoritam koristi dva 3×3 kernela za simulaciju izvedenica slike u smjerovima x i y. Matematički se može prikazati kao[3]:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * I(x, y), \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * I(x, y) \tag{2-15}$$

To rezultira slikom koja ističe rubove jednostavno veličinom njihovog gradijenta. Sobelov algoritam je brži i lakši za implementaciju, ali manje precizan u usporedbi s Cannyjevim algoritmom. Sobelov algoritam se obično koristi za operacije gdje su brzina i jednostavnost iznad svega, gdje je sama točnost detekcije ruba nešto manje bitna.

Ovi algoritmi zasnivani na operacijama naširoko se koriste za detekciju kontura koja ima značajnu važnost u brojnim područjima računalnog vida i analize slike. Canny je visokokvalitetni algoritam i preporučuje se za zadatke u kojima su preciznost i pouzdanost glavni prioritet. Operator Sobel povećava brzinu obrade nauštrb točnosti detekcije ruba te ga se isplati koristiti u aplikacijama u stvarnom vremenu ili kada ne postoji zahtjev za visokom preciznošću detekcije rubova. Ova dva algoritma zajedno pokrivaju širok spektar slučajeva upotrebe i na taj način

olakšavaju učinkovito i prilagodljivo otkrivanje obrisa pod različitim uvjetima osvjetljenja ili primjene.

Osim Canny i Sobel algoritama, često se koriste i drugi detektori rubova kao što su Prewitt, Roberts-Cross i Laplace, svaki sa svojim specifičnim prednostima i područjima primjene[19].

2.2.2. Metode za detekciju kontura

Detekcija kontura je ključna komponenta u analizi slika jer omogućuje identifikaciju granica objekata unutar slike. Konture definiraju oblik objekata, što je od izuzetne važnosti za zadatke prepoznavanja i segmentacije objekata. Postoji nekoliko metoda za detekciju kontura, od kojih svaka ima svoje prednosti i specifične primjene.

Jedna od osnovnih metoda je Freemanov lančani kod (engl. *Freeman Chain Code*)[22], koja predstavlja konture kao niz povezanih linija, kodiranih smjerom kretanja između susjednih točaka na konturi. Ova metoda pruža sažet prikaz konture i koristi se za jednostavne oblike.

Houghova transformacija [24] je popularna metoda za detekciju geometrijskih oblika unutar slike, poput linija i krugova. Djeluje tako da svaki rubni piksel u slici mapira u prostor parametara, gdje se pronalaze skupovi piksela koji zadovoljavaju određene geometrijske uvjete, čime se prepoznaju konture.

Algoritam marširajućih kvadrata (engl. *Marching Squares Algorithm*) [23] koristi se za detekciju kontura u binarnim slikama. Algoritam prolazi kroz sliku, tražeći prijelaze između područja različitog intenziteta. *Marching Squares* je često primjenjivan u obradi medicinskih slika, geografskoj kartografiji i drugim područjima gdje je potrebna detaljna analiza oblika.

U ovom radu, detekcija kontura ostvarena je korištenjem metode implementirane putem OpenCV funkcije `cv2.findContours`. Ova metoda je široko primjenjiva i koristi se za identifikaciju i praćenje kontura u različitim aplikacijama računalnog vida. Nakon pretprocesiranja, `cv2.findContours` detektira konture prepoznajući zatvorene rubove unutar binarne slike. Konture se predstavljaju kao nizovi točaka koje definiraju rubove objekata. Ovaj pristup je učinkovit jer omogućuje prilagodbu različitim uvjetima osvjetljenja i složenosti scena, što osigurava pouzdanu detekciju kontura.

Primjena ovih metoda omogućuje preciznu analizu oblika i strukture unutar slike, olakšavajući zadatke prepoznavanja i klasifikacije objekata. Usporedbom različitih metoda i prilagodbom specifičnim potrebama zadatka postiže se visoka preciznost i učinkovitost u detekciji kontura.

2.2.3. Metode za usporedbu kontura

Nakon detekcije kontura potrebno ih je usporediti kako bi se analizirali i prepoznali oblici objekata. Postoje različite metode za usporedbu kontura, od kojih svaka ima svoje prednosti i specifične primjene. Dvije značajne metode koje se često koriste su Hausdorffova udaljenost i Freemanov kod.

Hausdorffova udaljenost vrlo je snažan alat za mjerenje sličnosti između dviju kontura. Maksimalna udaljenost između točaka dviju kontura može se izračunati ovom metodom. Hausdorffova udaljenost odnosi se na najnepovoljniju slučajnu udaljenost između točaka, što je čini vrlo robusnom prema malim pomacima i deformacijama. Ovo svojstvo čini Hausdorffovu udaljenost posebno korisnom u slučajevima kada se konture mogu deformirati zbog perspektive, savijanja objekata ili drugih razloga. Matematički se može definirati kao[17]:

$$H(A,B) = max\left\{\max_{a\in A}\min_{b\in B} d(a,b), \max_{b\in B}\min_{a\in A} d(b,a)\right\} \quad (2\text{-}16)$$

gdje su A i B skupovi točaka na konturama, a *d(a,b)* je Euklidska udaljenost između točaka a i b. Ovaj način je koristan za prepoznavanje oblika gdje su konture djelomično pomaknute ili izmjenjene.

Prednosti korištenja Hausdorffove udaljenosti su dobro dokumentirane, prvenstveno naglašavajući njegovu sposobnost čak i u otkrivanju sitnih modifikacija preko dviju kontura, svojstvo koje je vrlo traženo kada se radi o zadacima koji zahtijevaju točnost. Ali budući da je ova metoda osjetljiva na ekstremne vrijednosti, može biti manje učinkovita u slučajevima kada su konture izložene značajnom šumu ili nepravilnostima.

Freemanovi kodovi, s druge strane, kodiraju konture u drugom obliku pokazujući niz smjerova kretanja duž rubova objekta. Dakle, ovaj umjesto toga kodira informacije o konturama kao slijed smjerova koji idu duž rubova, tako da se mogu usporediti parovi na temelju tih smjerova. Budući da se sve geometrijske značajke navedene u njegovom kodu mogu lako izračunati ili usporediti za konture, Freemanov kod je posebno prikladan za prepoznavanje oblika objekta.

Ova tehnika posebno dobro funkcionira u slučajevima kada unutar slike postoje neki predmeti koje je potrebno prepoznati. Jednostavnost i učinkovitost konturnog kodiranja znači da je Freemanov kod posebno prikladan za aplikacije analize slike u stvarnom vremenu. Međutim, budući da se zasniva na smjerovima kretanja, Freemanov algoritam može biti osjetljiv na rotacije i skaliranje objekta, što može onemogućiti njegovu upotrebu kao ljudskog detektora opće namjene u mnogim slučajevima.

Hausdorffova udaljenost u kombinaciji s Freemanovim kodom može oblikovati opsežnu metodu usklađivanja kontura koja dovodi do kompromisa između točnosti i brzine. Iako Hausdorffova udaljenost osigurava stabilnost prilikom rotacije i malih pomaka, ovu metodu ne mogu svi dobro prepoznati, pa je Freemanov kod jednostavan način opisa korištenjem informacija o smjeru za prepoznavanje oblika.

2.3. Algoritmi za usporedbu boje objekata

Boja je jedan od ključnih znakova na koje se ljudi oslanjaju prilikom prepoznavanja i klasificiranja stvari u svakodnevnom životu. Analiza i usporedba boja objekta kritični su u prepoznavanju objekata, klasifikaciji slika i zadacima razumijevanja scene, posebno za aplikacije računalnog vida. U nastavku će se detaljno opisati prostori boja, metode ekstrakcije boja iz slika i neki algoritmi koji omogućuju usporedbu između objekata na slikama na osnovu boje.

2.3.1. Prostori boja

U kontekstu digitalnih slika, prostor boja je specifična organizacija boja (ili kroz matematički model) međusobno različitih s obzirom na kromatske karakteristike. Svaki prostor boja pruža različite sheme definiranja boja i dopušta analizu i obradu boje prema zahtjevima za određenu primjenu. Postoji niz modela prostora boja, uključujući RGB, HSV, HLS i Lab, koji su među najčešće korištenima u računalnom vidu.

RGB prostor boja (crvena, zelena i plava) je jedan od najosnovnijih i najčešće korištenih prostora boja u digitalnoj obradi slike. Ovo je prostor boja koji koristi aditivni model, što znači da se boje formiraju na zasniva miješanja razina crvenog, zelenog i plavog svjetla. R, G i B označavaju intenzitet crvene, zelene i plave boje. Komponente su normalizirane i njihove vrijednosti obično leže između 0 i 255. S ovim elementima u različitim omjerima generira se širok raspon boja. Na primjer, bijela je napravljena tako da svaka od triju komponenti ima najveću moguću vrijednost (255, 255, 255), a crna kada su sve komponente na minimumu (0,0,0).

Koristi se za prikaz boja na monitorima, kamerama i stvarima koje emitiraju svjetlost. Njegova jednostavnost i odnos prema fizičkoj slici svjetla čine ga moćnim alatom za mnoge namjene, ali nije nužno optimalan u svim slučajevima kada se analizira percepcija boja.

HSV (engl. *Hue, Saturation and Value*) prostor boja izvorno nije korišten za određivanje boja koje dolaze iz RGB slika. Ovaj prostor boja odvaja nijansu od zasićenosti i vrijednosti, čime se omogućuje intuitivnija kontrola boja.

- Ton boje (H) – vrsta boje, izražena u stupnjevima od 0° do 179°. Boja koju oko vidi određena je nijansom, kao što su crvena, žuta, zelena, plava i sve nijanse između.
- Zasićenje (S) – opisuje koliko je boja intenzivna ili čista, 0% su razine sive (bez boja), a 100% najčišći oblik te boje.
- Vrijednost (V) – svjetlina boje, gdje 0% znači potpuno crnu dok 100% predstavlja maksimalnu svjetlinu boje.

U grafičkom dizajnu, obradi slika i računalnoj grafici općenito se koristi HSV prostor boja jer omogućuje rad s bojama na način koji je bliži ljudskoj percepciji. Sveukupno, ovo omogućuje izmjenu nijanse slike bez utjecaja na svjetlinu/zasićenost, a to je svojstvo dobro za mnoge svrhe.

HLS (engl. *Hue, Lightness, Saturation*) prostor boja sličan je HSV-u, ali umjesto vrijednosti je svjetlina. Svjetlina je, s druge strane, količina boje koju ima HLS prostor boja i u tome prostoru to predstavlja prosječnu svjetlinu između 0 do n elemenata od najcrnjih dijelova do najbjeljeg elementa, što daje drugačiji pogled od HSV-a.

- Ton boje (H) – Kao u HSV prostoru, to je osnovna boja.
- Svjetlina (L) – percipira se od 0-100 ljestvice intenziteta, ali akromatika se proteže cijelom njegovom dužinom. Krajnji krajevi su crno-bijeli, a na 50 je neutralna svjetlina.
- Zasićenje (S) – mjera koliko je čista ili izblijedjela boja naklonjena bijeloj.

HLS se često koristi u softveru kada je potrebno zasebno upravljati svjetlinom slike ili intenzitetom boje, što je posebno važno u aplikacijama koje zahtijevaju preciznu kontrolu boja.

Lab prostor boja (CIE Lab) osmišljen je tako da jednaki koraci u vrijednosti L predstavljaju otprilike jednako vidljive promjene u svjetlini, odnosno da su promjene u boji u skladu s ljudskom percepcijom razlika u boji. Lab prostor boja se sastoji od 3 dijela:

- L (svjetlina) – svjetlina boje, od 0 (crna) do 100 (bijela).
- a – Mjerenje boje u zeleno-crvenoj osi. Ako je vrijednost negativna, to znači da će proizvesti zelenu boju, a crvenu ako je pozitivna.
- b – Mjeri boju od plavkaste na jednom kraju do žućkaste na drugom. Kao rezultat toga dobivene su niže vrijednosti plave i više vrijednosti žute boje.

Lab prostor boja koristi se u aplikacijama koje se odnose na računarstvo, koje zahtijevaju mnogo točniji rad s bojama, kao što je prepoznavanje objekata ili napredni uređivač slika, na primjer, ali također i mjerila i ispis. Budući da su sve boje definirane jednolično, to znači da se podaci mogu analizirati s jednakom preciznošću za razliku od drugih prostora boja.

2.3.2. Ekstrakcija boje

Nakon što su definirani prostori boja, analiza slike može se izvršiti izdvajanjem relevantnih informacija o boji. Dominantnost boja na slici lako se rastavlja i sažima tako da se čak mogu usporediti dvije slike s obzirom na boje. Neke od najčešćih metoda za ekstrakciju boja su histogram boja u različitim prostorima boja i grupiranje boja (engl. *color clustering*).

Histogram boja je stupičasti graf koji predstavlja distribuciju boja unutar slike mjerenjem frekvencije pojavljivanja različitih boja u određenom prostoru boja. Histogramom je moguće prikazati sažeti prikaz bokja u slici te omogućuje usporedbu boje između različitih slika. Histogram boja u RGB prostoru se može matematički prikazati kao[4]:

$$H(c) = \sum_{x=1}^{M} \sum_{y=1}^{N} I(x, y, c) \tag{2-17}$$

gdje je $H(c)$ histogram za kanal *c* (R, G ili B), *I(x,y,c)* intenzitet boje u kanalu *c* na *(x,y)* koordinatama, a *M* i *N* su dimenzije slike.

Iako je RGB prostor boja najčešće korišten, koriste se i drugi prostori poput HSV, HLS i Lab. Za razliku od RGB prostora boja, u HSV i HLS prostoru boja moguće je posebno prikazati i preciznije analizirati nijanse boja. Histogram boja u HSV prostoru se definira[4]:

$$H(h, s, v) = \sum_{x=1}^{M} \sum_{y=1}^{N} \delta\big(h - H(x, y)\big) * \delta\big(s - S(x, y)\big) * \delta(v - V(x, y)) \tag{2-18}$$

gdje $H(x, y)$, $S(x, y)$ i $V(x, y)$ predstavljaju vrijednosti nijanse boje, zasićenja i vrijednost za (x, y) piksel. δ je Diracova delta funkcija s kojom je moguće brojati piksele s istim vrijednostima.

Grupiranje boja (engl. *Color clustering*) pomoću K-srednjih vrijednosti (engl. *K-means*) algoritma je naprednija metoda ekstrakcije boja koja grupira piksele u klastere na temelju njihovih

boja. Ovom metodom je moguće identificirati *K* dominantnih boja u slici, što je vrlo korisno za prepoznavanje objekata i analizu scene.

K-means algoritam smanjuje razlike unutar svakog klastera, raspoređujući piksele u *K* klastera, gdje je *K* unaprijed definirani broj klastera. Svaki piksel se pridružuje klasteru čija je središnja točka (centroid) najbliža boji tog piksela, mjereno Euklidskom udaljenošću. Na taj način, slične boje se grupiraju zajedno, čineći identifikaciju dominantnih boja u slici jasnijom i preciznijom. Matematički se može prikazati[5]:

$$\min_{C} \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2 \tag{2-19}$$

gdje je C_i skup piksela unutar *i*-tog klastera, *x* je vektor boje piksela, a μ_i je centar *i*-tog klastera. Ovim se postupkom sažima informacija o boji slike na nekoliko dominantnih boja, čime je analiza i usporedba boje olakšana.

Osim K-means postoji i **Mean shift** algoritam [25], koji za razliku od K-means algoritma ne mora imati unaprijed određen broj klastera, jer sam određuje broj klastera, te se lakše bori s klasterima nepravilnog oblika ili različitih veličina. Ne donosi nikakve pretpostavke o veličini ili obliku klastera jer se oslanja na informacije o gustoći podataka te ga to svojstvo čini fleksibilnijim i sposobnijim za identificiranje klastera. Odličan je izbor za skupove podataka koji nemaju jasan broj klastera no sporiji je od K-means algoritma.

2.3.3. Usporedba boje

Idući korak je usporedba ekstrahiranih boja iz slike kako bi se odredila sličnost između objekata zasnovana na boji, što je ključno kod prepoznavanja, klasifikacije i analize vizualnih podataka. Postoji nekoliko metoda usporedbe boja, uključujući Euklidsku udaljenost i korelaciju histograma.

Euclidska udaljenost je metoda koja računa udaljenost između boja u prostoru boja, kao što su HSV, HLS ili RGB, koristeći standardnu Euclidsku metriku. Ova metoda pruža jednostavan način usporedbe boja te omogućava brzo izračunavanje sličnosti između slika. Međutim, može biti osjetljiva na promjene u svjetlini. Može se prikazati matematički između dviju boja (R_1, G_1, B_1)i (R_2, G_2, B_2) u RGB prostoru boja kao[6]:

$$d = \sqrt{(R_2 - R_1)^2 + (G_2 - G_1)^2 + (B_2 - B_1)^2} \tag{2-20}$$

Omogućava brzu i jednostavnu usporedbu boja različitih objekata na slikama te je vrlo korisna metoda za usporedbu boja, posebno u kombinaciji s drugim metodama koje pružaju dodatne informacije.

Korelacija histograma uspoređuje distribuciju boja između dviju slika pomoću histograma boja i mjeri stupanj sličnosti. Robusna je na male promjene u boji i svjetlini te je moguće identificirati slike sa sličnim bojama snimljenim u različitim uvjetima. Korelacija se izračunava[6]:

$$R(H_1, H_2) = \frac{\sum_i (H_1(i) - \overline{H_1})(H_2(i) - \overline{H_2})}{\sqrt{\sum_i (H_1(i) - \overline{H_1})^2 (H_2(i) - \overline{H_2})^2}} \tag{2-21}$$

gdje su $H_1(i)$ i $H_2(i)$ vrijednosti histograma i-tog bina za dvije slike, a $\overline{H_1}$ i $\overline{H_2}$ su srednje vrijednosti histograma. Visoke vrijednosti izračunate korelacije ukazuju na veliku sličnost između histograma dviju slika, odnosno da su slike slične u smislu boje.

2.4. Ostale metode uspoređivanja dinamičnih slika

Osim metoda zasnovanih na ključnim točkama, konturama i bojama, postoje sofisticiranije tehnike koje se koriste za uspoređivanje dinamičnih scena. Tri značajna pristupa u ovoj domeni su optički tok, struktura iz pokreta (SfM) i pristupi zasnovani na dubokom učenju.

Optički tok je tehnika koja se koristi za praćenje kretanja objekata između uzastopnih slika analizirajući promjene u intenzitetu piksela. Ova metoda omogućava dobivanje informacija o smjeru i brzini kretanja objekata unutar scene. Optički tok može se izračunati pomoću različitih algoritama, kao što su Lucas-Kanade metoda [26] ili Horn-Schunck metoda [27]. Praćenje kretanja objekata putem optičkog toka je izuzetno korisno u aplikacijama kao što su autonomna vozila, video nadzor i analiza sportskih događaja, gdje je važno razumjeti dinamiku scena. Jedna od glavnih prednosti optičkog toka je njegova sposobnost da pruži detaljne informacije o lokalnim kretanjima unutar slike, omogućujući precizno praćenje objekata. Međutim, optički tok može biti osjetljiv na brze promjene i kompleksne scene s velikim brojem pokretnih objekata. Unatoč tome, njegova primjena je široka i pruža temelj za mnoge napredne sustave praćenja.

Struktura iz pokreta (SfM) je metoda koja koristi višestruke slike iste scene za rekonstrukciju 3D strukture. Korištenjem informacije iz različitih perspektiva, SfM metode omogućuju preciznu rekonstrukciju scene i usporedbu slika na osnovu trodimenzionalnih informacija. Ova metoda je posebno korisna u područjima kao što su 3D modeliranje, virtualna

stvarnost i arheološka istraživanja, gdje je važno dobiti točnu trodimenzionalnu reprezentaciju objekata i prostora. SfM koristi ključne točke i njihove podudarnosti između slika kako bi rekonstruirao 3D strukturu. Ova metoda može pružiti izuzetno detaljne i precizne modele, ali zahtijeva visoku računalnu snagu i kvalitetne ulazne podatke. Njena primjena je široka, od inženjerskih projekata do filmske industrije, gdje točne 3D rekonstrukcije omogućavaju realistične i detaljne prikaze.

Duboko učenje (engl. *Deep learning*) pristupi, posebno oni zasnovani na dubokim neuronskim mrežama, predstavljaju najmodernije metode za uspoređivanje slika. Neuronske mreže, kao što su konvolucijske neuronske mreže (CNN), mogu automatski naučiti složene reprezentacije iz velikih skupova podataka slika. CNN arhitekture su posebno učinkovite u ekstrakciji značajki i uspoređivanju slika, omogućujući visoku točnost i robustnost u različitim zadacima. Duboko učenje omogućava sustavima da prepoznaju obrasce i značajke koje bi bile teško ručno programirati, pružajući fleksibilnost i preciznost u analizi slika. Ovi sustavi mogu biti trenirani na velikim skupovima podataka kako bi naučili razne vizualne karakteristike, što ih čini izuzetno moćnima u aplikacijama poput prepoznavanja lica, klasifikacije slika i analize medicinskih snimaka. Iako zahtijevaju značajne resurse za treniranje i implementaciju, duboko učenje pruža nevjerojatne mogućnosti za napredne analize i usporedbe slika.

3. POSTOJEĆA RJEŠENJA ZA USPOREDBU DINAMIČKIH SLIKA

3.1. Pregled relevantnih radova i istraživanja u području usporedbe dinamičkih slika

Jedan od ključnih radova u ovom području je onaj Matije Pula u kojem je razvio DISM algoritam (engl. *Dynamic Image Similarity Measure*) [10],[11]. Ovaj algoritam posebno je osmišljen za precizno mjerenje sličnosti između dviju dinamičkih slika, koristeći se tehnikom detekcije ključnih točaka unutar slike. Ključne točke koje su identificirane unutar referentne slike uspoređuju se s onima u testnoj slici, a geometrijski odnosi među tim točkama koriste se za kvantifikaciju sličnosti između slika. DISM algoritam nije usmjeren na praćenje objekata kroz sekvence slika, već na usporedbu dviju statičkih slika, što omogućuje preciznu analizu sličnosti, čak i kada dolazi do promjena u perspektivi ili položaju objekata unutar slike. Ova metoda pokazala se korisnom u kontekstu testiranja aplikacija s dinamičkim korisničkim sučeljima, gdje je bitno osigurati točnu usporedbu vizualnih elemenata kroz različite prikaze. Iako je DISM algoritam vrlo precizan, suočava se s izazovom visoke računalne zahtjevnosti, što znači da njegova primjena u sustavima koji zahtijevaju brzo procesiranje podataka, poput sustava koji rade u stvarnom vremenu, može biti ograničena. Unatoč tome, DISM algoritam ostaje jedno od ključnih rješenja u usporedbi dinamičkih slika, pružajući čvrstu osnovu za daljnja istraživanja i unapređenja u ovom području.

Sličan doprinos, ali s drugačijim fokusom, nalazi se u radu Ahmeda Ibrahima i Craiga Vallija iz 2015. godine [16], koji su uveli metodu dinamičkog vremenskog rastezanja (engl. *Dynamic Time Warping* - DTW) za usporedbu slika zasnovanih na fraktalnim značajkama. DTW je izvorno razvijen za prepoznavanje sličnosti u vremenskim serijama, poput govora, ali Ibrahim i Valli uspješno su ga primijenili na slike. Ova metoda omogućuje usporedbu slika koje su prošle kroz manje transformacije, poput promjene formata ili boje, te pruža robusnu usporedbu iako slike nikada nisu potpuno identične. DTW se pokazao izuzetno korisnim u digitalnoj forenzici, gdje je često potrebno identificirati slike koje su prošle kroz minimalne promjene, a tradicionalne metode usporedbe mogu biti neadekvatne. Glavna prednost DTW metode je njezina sposobnost prepoznavanja sličnosti unatoč promjenama u slici, dok je glavni nedostatak visoka računalna složenost koja može ograničiti njezinu upotrebu u aplikacijama koje zahtijevaju veliku brzinu. Iako pruža značajan napredak u digitalnoj forenzici, njezina primjena u sustavima koji rade u stvarnom vremenu može biti otežana zbog zahtjeva za računalnim resursima.

Još jedan značajan doprinos dolazi iz rada Gagana Kanojije i Shanmuganathana Ramana iz 2020. godine [15], koji se bavi problemom simultanog prepoznavanja i uklanjanja dinamičkih objekata iz slika s višestrukim pogledom (engl. *multi-view*). Algoritam razvijen u ovom radu koristi set slika snimljenih iz različitih kutova iste scene kako bi identificirao i uklonio pokretne objekte, ostavljajući iza sebe samo statične dijelove scene. Ovaj pristup omogućuje automatsko uklanjanje neželjenih objekata bez potrebe za korisničkom intervencijom, što je izuzetno važno u kontekstu automatizirane obrade slika, primjerice u nadzornim sustavima. Algoritam je zasnovan na analizi podudarnosti između slika iz različitih kutova, čime se omogućuje precizna detekcija dinamičkih objekata. Ovaj pristup pokazao se vrlo učinkovitim u stvarnim aplikacijama, posebno u dinamičnim okruženjima kao što su prometne situacije ili javni prostori s velikim brojem pokretnih objekata. Međutim, unatoč visokoj razini automatizacije i točnosti, glavni izazov ovog rješenja leži u točnom usklađivanju slika iz različitih kutova, što može dovesti do manjih netočnosti, osobito u situacijama gdje postoje velike razlike u perspektivi između slika.

Doprinos rada Jianshenga Weija i njegovih suradnika iz 2022. godine [12] usmjeren je na razvoj metode koja koristi kombinaciju detekcije objekata i geometrijskih ograničenja za filtriranje dinamičkih objekata iz sekvenci slika. Ovaj pristup koristi napredne tehnike dubokog učenja za detekciju objekata, dok se geometrijska ograničenja primjenjuju kako bi se osiguralo točno filtriranje pokretnih objekata, ostavljajući statičke dijelove slike netaknutima. Ova metoda se pokazala vrlo učinkovitom u složenim scenama s visokim razinama šuma, poput prometnih gužvi ili situacija s mnogo pokretnih osoba. Glavna prednost ovog pristupa je njegova sposobnost preciznog prepoznavanja i uklanjanja dinamičkih objekata, što poboljšava kvalitetu analize slike i smanjuje broj lažnih pozitivnih rezultata. Međutim, složenost implementacije ove metode i potreba za naprednim računalnim resursima predstavljaju izazov, osobito za primjenu u manje opremljenim sustavima ili u kontekstima gdje su resursi ograničeni.

Domagoj Steiner je u svom radu iz 2022. godine [13],[18] dao značajan doprinos u području usporedbe dinamičkih slika, primjenjujući napredne statističke modele, uključujući korištenje sijamskih neuronskih mreža. Njegov rad usredotočuje se na preciznu analizu i usporedbu dinamičkih slika, koristeći neuronske mreže za dublju analizu i prepoznavanje obrazaca unutar slika. Sijamske neuronske mreže analizirane su usporedno s DISM algoritmom, a rezultati pokazuju da, iako ove mreže nude zanimljive mogućnosti za usporedbu dinamičkih slika, one još uvijek ne nadmašuju točnost i preciznost koju pruža DISM algoritam. Ova metoda zahtijeva dodatno treniranje modela, što može predstavljati dodatno opterećenje u kontekstima gdje je potrebna brza obrada podataka. Ipak, prednosti koje nude neuronske mreže u smislu

prilagodljivosti i mogućnosti dublje analize čine ih perspektivnim alatom za daljnje istraživanje i razvoj.

Na kraju, rad koji su proveli Syed Muslim Jameel, Manzoor Ahmed Hashmani, Mobashar Rehman i Arif Budiman 2020. godine [14] istražuje prednosti multi-senzorskog pristupa za simultano praćenje i detekciju dinamičkih objekata. Ovaj rad naglašava važnost integracije podataka iz različitih senzora kako bi se postigla sveobuhvatna analiza scena koje uključuju dinamičke objekte. Kombiniranjem različitih izvora podataka, ovaj pristup omogućuje povećanu točnost i robustnost u složenim okruženjima, kao što su industrijska postrojenja ili prometni sustavi. Prednost ovog pristupa je sposobnost smanjenja lažnih alarma i poboljšanja točnosti u detekciji objekata, no složenost upravljanja multi-senzorskim podacima i potrebna infrastruktura mogu predstavljati izazov u uvjetima ograničenih resursa. Ipak, multisenzorski pristup predstavlja značajan napredak u poboljšanju performansi sustava za praćenje i analizu dinamičkih objekata, posebno u zahtjevnim okruženjima gdje je potrebna visoka razina preciznosti.

Kombinacija ovih radova pruža sveobuhvatan pregled trenutnog stanja tehnologije u području usporedbe dinamičkih slika, pri čemu se identificiraju ključne prednosti i izazovi svakog pristupa. Ovi nalazi pružaju solidnu osnovu za daljnje istraživanje i razvoj naprednijih algoritama u budućnosti, s ciljem poboljšanja učinkovitosti, točnosti i primjenjivosti u različitim kontekstima.

3.2. Analiza prednosti i nedostataka postojećih rješenja

Analiza postojećih rješenja za usporedbu dinamičkih slika otkriva širok spektar metoda koje, iako učinkovite u specifičnim kontekstima, dolaze s određenim prednostima i nedostacima koji su ključni za razumijevanje njihove primjene u različitim situacijama. Ova analiza omogućuje dublje sagledavanje tehnoloških ograničenja i potencijala, pružajući osnovu za daljnji razvoj i optimizaciju postojećih algoritama.

DISM algoritam, koji je razvio Matija Pul, jedno je od najpreciznijih rješenja za usporedbu dinamičkih slika, posebno kada je riječ o usporedbama dviju statičkih slika. Njegova sposobnost prepoznavanja sličnosti čak i u slučajevima značajnih promjena u perspektivi ili položaju objekata čini ga vrlo robusnim rješenjem za usporedbu dinamičkih slika. Međutim, glavni nedostatak DISM algoritma je njegova visoka računalna zahtjevnost, što može ograničiti njegovu primjenu u aplikacijama koje zahtijevaju brzinu, kao što su sustavi koji rade u stvarnom vremenu.

Dinamičko vremensko rastezanje (DTW), koje su uveli Ahmed Ibrahim i Craig Valli, nudi učinkovit pristup za usporedbu slika koje su prošle kroz manje transformacije, poput promjena

formata ili boje. Prednost DTW metode leži u njenoj sposobnosti prepoznavanja sličnosti unatoč promjenama, što je korisno u digitalnoj forenzici. Ipak, kao i DISM, DTW se suočava s problemom visoke računalne složenosti, što može biti prepreka u aplikacijama koje zahtijevaju brzinu .

Gagan Kanojia i Shanmuganathan Raman razvili su algoritam za simultano prepoznavanje i uklanjanje dinamičkih objekata iz slika s višestrukim pogledom. Ovaj pristup omogućuje visok stupanj automatizacije u obradi slika i automatsko uklanjanje pokretnih objekata bez potrebe za korisničkom intervencijom. To je posebno korisno u nadzornim sustavima i analizi. Međutim, točno usklađivanje slika snimljenih iz različitih kutova predstavlja značajan izazov, što može rezultirati netočnostima, osobito u situacijama s velikim varijacijama perspektive. Unatoč ovim izazovima, algoritam pokazuje visok potencijal za primjenu u stvarnim scenarijima, a daljnja istraživanja mogla bi se usredotočiti na poboljšanje točnosti u situacijama s velikim varijacijama perspektive.

Metoda za filtriranje dinamičkih objekata koju su razvili Jiansheng Wei i njegovi suradnici kombinira napredne tehnike dubokog učenja s geometrijskim ograničenjima, što omogućuje visoku točnost u detekciji i eliminaciji pokretnih objekata iz slika. Ova metoda se pokazala vrlo učinkovitom u složenim scenama s visokim razinama šuma, ali je složenost implementacije i potreba za naprednim računalnim resursima značajan izazov za primjenu u manje opremljenim sustavima. Daljnje istraživanje moglo bi se usmjeriti na optimizaciju ove metode kako bi se smanjili zahtjevi za resursima.

Domagoj Steiner u svom radu iz 2022. godine primjenjuje napredne statističke modele, uključujući korištenje sijamskih neuronskih mreža, za preciznu analizu i usporedbu dinamičkih slika. Njegov rad uspoređuje neuronske mreže s DISM algoritmom, pri čemu rezultati pokazuju da, iako neuronske mreže nude dublju analizu i prilagodljivost, još uvijek ne nadmašuju točnost koju pruža DISM algoritam. Ova metoda zahtijeva dodatno treniranje modela, što može biti izazovno u aplikacijama koje zahtijevaju brzu obradu podataka. Ipak, prednosti koje nude neuronske mreže u smislu prilagodljivosti čine ih perspektivnim alatom za daljnje istraživanje i razvoj.

Konačno, rad autora Syed Muslim Jameel, Manzoor Ahmed Hashmani, Mobashar Rehman i Arif Budiman istražuje multi-senzorski pristup koji se ističe sposobnošću integracije podataka iz više senzora, što omogućuje poboljšanu točnost i robusnost u detekciji dinamičkih objekata. Ovaj pristup je posebno koristan u složenim okruženjima poput industrijskih postrojenja ili prometnih sustava, gdje može smanjiti broj lažnih alarma i poboljšati ukupnu točnost sustava. Ipak, složenost

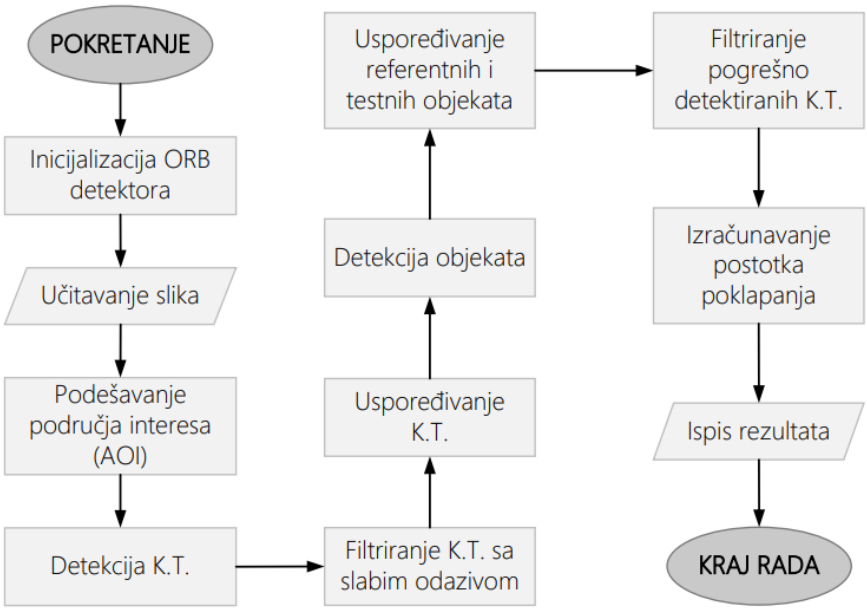
upravljanja multi-senzorskim podacima i potreba za specijaliziranom infrastrukturom predstavljaju izazov, posebno u uvjetima ograničenih resursa.

Kroz ovu analizu jasno je da svako od postojećih rješenja za usporedbu dinamičkih slika ima svoje specifične prednosti i izazove. Dok neki algoritmi nude visoku točnost i robusnost, drugi se suočavaju s ograničenjima vezanim uz računalne zahtjeve i složenost implementacije. Ovi nalazi pružaju temelj za daljnje istraživanje i razvoj, s ciljem optimizacije postojećih rješenja i izrade novih metoda koje će ponuditi bolji balans između točnosti, brzine i računalne učinkovitosti.

4. RAZVOJ I NADOGRADNJA DISM ALGORITMA ZA USPOREDBU DINAMIČNIH SLIKA

4.1. DISM algoritma

DISM algoritam razvijen je kao rješenje za usporedbu dinamičkih slika koje se mijenjaju tijekom vremena. Takav je algoritam učinkovit u prepoznavanju i usporedbi slika čak i kada su objekti unutar njih prošli transformacije poput rotacije, skaliranja, translacije i okluzije. Princip rada DISM algoritma zasnovan je na detekciji ključnih točaka, koje su stabilne karakteristike na slici, i usporedbi tih točaka s točkama na drugim slikama. Proces rada DISM algoritma uključuje inicijalizaciju, gdje se definiraju svi potrebni parametri i učitavaju slike za usporedbu. Prvi ključni korak koji se izvodi u algoritmu je otkrivanje ključnih točaka pomoću ORB algoritma. Ovaj detektor identificira značajne točke unutar slike koje su otporne na transformacije poput rotacije i skaliranja, osiguravajući da ključne točke ostanu stabilne i dosljedno prepoznatljive unatoč razlikama u izgledu istog objekta na različitim slikama. Slika 4.1, gdje je prikazan dijagram toka DISM algoritma, jasno prikazuje kako se ključne točke (KT) koriste kao središnji element u procesu usporedbe dinamičkih slika, omogućujući algoritmu da otkrije i prati promjene u dvjema slikama.

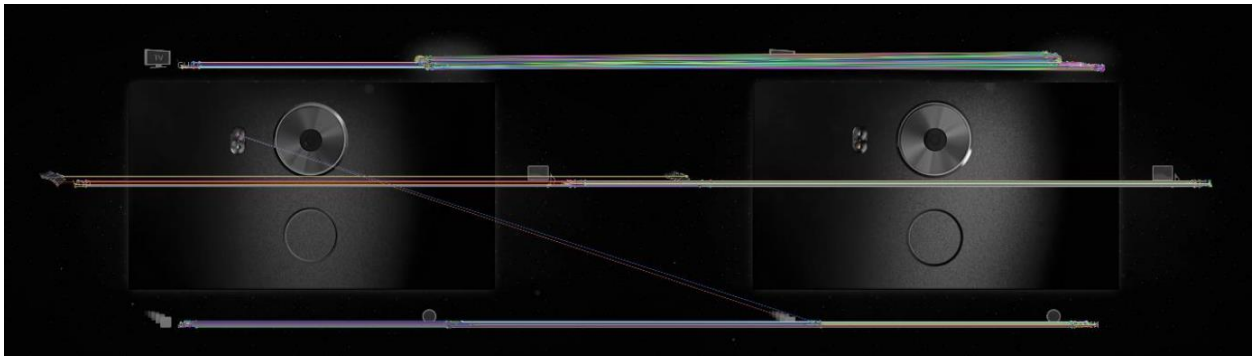


Slika 4.1. *Dijagram toka DISM algoritma* [10]

Algoritam zatim provodi postupak filtriranja KT kako bi uklonio KT koje nisu dovoljno značajne za usporedbu ili koje bi mogle dovesti do netočnih rezultata. Filtriranje se zasniva na mjerenju odziva KT, pri čemu se slabiji odzivi eliminiraju kako bi se osigurala stabilnost preostalih

KT. Ovaj korak smanjuje lažno pozitivne detekcije i osigurava da samo najrelevantnije točke ostanu za daljnju analizu. Nakon ovog koraka, algoritam nastavlja uspoređivati KT referentne i testne slike. U ovoj fazi algoritam identificira i grupira KT koje pripadaju istom objektu na slici. Grupiranje dopušta algoritmu povezivanje KT koje su prostorno bliske i imaju slične deskriptore. Podudarnosti između KT uspostavljaju se na osnovu usporedbe njihovih deskriptora, tražeći najsličnije parove točaka između dviju slika.

Nakon uspješne detekcije i grupiranja KT, algoritam prelazi na usporedbu objekata referentne i testne slike. Algoritam identificira sličnosti i razlike između objekata na osnovu njihove geometrije i rasporeda KT. Algoritam koristi matricu udaljenosti za mjerenje sličnosti između objekata, povezujući svaki objekt s referentne slike s najbližim objektom na testnoj slici. Ako su svi objekti s referentne slike uspješno pronađeni na testnoj slici i njihove KT su ispravno usklađene, slike se smatraju identičnima u smislu sadržaja. Ovaj postupak provjere podudaranja ključan je za određivanje stupnja sličnosti između dviju slika, omogućujući algoritmu da detektira čak i najmanje promjene u rasporedu i obliku objekata. Konačni rezultat usporedbe referentne i testne slike prikazan je na slici 4.2., gdje se može vidjeti kako su svi objekti na testnoj slici pravilno detektirani i povezani s odgovarajućim objektima na referentnoj slici. Linije koje povezuju KT između slika pokazuju točnu podudarnost, pokazujući snagu DISM algoritma u prepoznavanju sličnosti unatoč različitim transformacijama slike.



Slika 4.2. *Prikaz poklapanja ključnih točaka između dviju slika* [10]

DISM algoritam također uključuje fazu analize rezultata, koja mjeri stupanj sličnosti između dviju slika. Ovaj korak omogućuje korisnicima da detaljno kvantificiraju razlike između slika, dajući uvid u to koliko su slike različite ili slične. Rezultati ove analize mogu se primijeniti u različitim aplikacijama, od testiranja grafičkih sučelja do analize promjena u video sekvencama. DISM algoritam moćan je alat za usporedbu dinamičkih slika, koji omogućuje točno praćenje promjena i otkrivanje sličnosti između slika koje se neprestano mijenjaju. Njegova primjena u područjima kao što su računalni vid, analiza slike i automatizirano testiranje softvera omogućena

je njegovom sposobnošću izdvajanja i usporedbe KT čak i kada slike prolaze kroz promjene, poput rotacije, skaliranja, translacije, promjene u osvjetljenju ili okluzije.

4.2. Nadogradnje DISM algoritma

Osnovna verzija algoritma DISM pokazala je izvanredan kapacitet za usporedbu dinamičkih slika, uspješno se noseći s izazovima kao što su šum, rotacija i i skaliranje. Međutim, daljnjom primjenom i analizom utvrđeno je da su potrebna poboljšanja algoritma kako bi se povećala njegova učinkovitost i preciznost u određenim zadacima. Identificirani su određeni slučajevi u kojima se rezultati mogu poboljšati, osobito u situacijama koje zahtijevaju detaljniju analizu slike ili preciznije filtriranje podataka. Motivacija za ove nadogradnje proizlazi iz potrebe za optimiziranjem algoritma za određene scenarije primjene. Na primjer, filtriranje KT moglo bi se učiniti preciznijim kako bi se smanjila šansa obrade viška informacija, što je posebno važno pri radu s velikim skupovima podataka. Osim toga, postupak izdvajanja relevantnih dijelova slike omogućuje algoritmu da se fokusira isključivo na ključna područja, povećavajući brzinu i učinkovitost obrade. Daljnja poboljšanja uključuju dodavanje metoda za usporedbu kontura i boja objekata, omogućujući precizniju analizu oblika i boja na složenim slikama. Ove nadogradnje ne samo da poboljšavaju točnost algoritma, već i povećavaju njegovu fleksibilnost, omogućujući prilagodbu specifičnim potrebama korisnika.

Sljedeće sekcije detaljno opisuju svako od ključnih poboljšanja primijenjenih na izvorni DISM algoritam, uključujući filtriranje KT, izdvajanje važnih dijelova slike, usporedbu kontura objekta i usporedbu boja objekta. Svaka od ovih nadogradnji predstavlja značajan korak prema postizanju učinkovitijeg i prilagodljivijeg algoritma u različitim okruženjima.

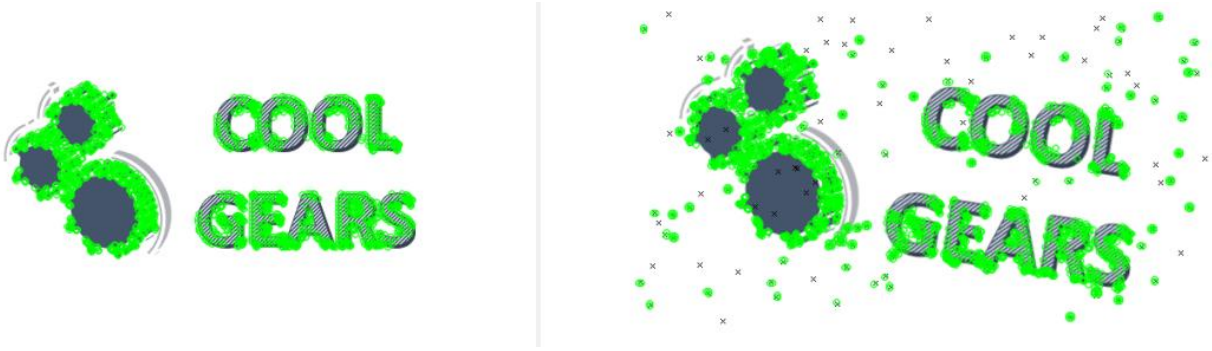
4.2.1. Filtriranje ključnih točaka

Jedan od najvažnijih koraka za poboljšanje DISM algoritma je filtriranje KT. U osnovnoj verziji algoritam je obrađivao sve detektirane KT, što je ponekad dovodilo do korištenja točaka koje nisu bile dovoljno pouzdane ili relevantne za točnu usporedbu slika. To bi moglo smanjiti točnost algoritma, posebno u složenim slikovnim scenama gdje su otkrivanje KT i analiza presudni za cijeli proces. Kako bi se riješili ti problemi, algoritam RANSAC (engl. *Random Sample Consensus*) integriran je u nadograđenu verziju kao robusna metoda za procjenu modela iz podataka koji sadrže visok postotak odstupanja ili šuma. RANSAC filtrira KT ignorirajući one koje nisu korisne za preciznu analizu i zadržavajući samo točke koje su najbolje usklađene s modelom transformacije između slika.

RANSAC radi tako da nasumično odabire mali broj KT i koristi ih za izračunavanje modela transformacije, kao što je homografija. Homografija, matematička transformacija, omogućuje mapiranje KT s jedne slike na drugu, što je ključno za preciznu usporedbu slika u slučajevima transformacija poput rotacije i skaliranja. Nakon što se model izračuna, preostale KT testiraju se u odnosu na njega. Točke koje se dobro uklapaju identificiraju se kao *inlieri,* dok se one koje značajno odstupaju od modela odbacuju. Ovaj se postupak ponavlja više puta kako bi se pronašao model s najviše *inliera*, što rezultira konačnim skupom KT pogodnih za daljnju analizu. Integracija algoritma RANSAC u DISM značajno je povećala točnost uparivanja parova KT, omogućujući eliminaciju KT koje proizlaze iz šuma ili lažnih detekcija. Ovo smanjuje količinu lažnih KT i fokusira se na ključne točke s najvećim utjecajem na usporedbu slika. Na slici 4.3. prikazane su referentna i testna slika koje će se koristiti zbog prikaza rada algoritma. Na testnoj slici objekt je rotiran 9° u smjeru kazaljke na satu, uvećan 14%, okludiran 2% i translatiran 7 piksela ulijevo i 2 piksela prema dolje. Slika 4.4. prikazuje referentnu i testnu sliku s detektiranim ključnim točkama. Na slici 4.5. prikazano je uparivanje ključnih točaka prije i nakon primjene RANSAC-a. Slika jasno pokazuje da RANSAC učinkovito filtrira nepouzdane točke, zadržavajući samo one najrelevantnije. *Python* kod za filtriranje KT korištenjem RANSAC algoritma prikazan je u prilogu P.4.1.

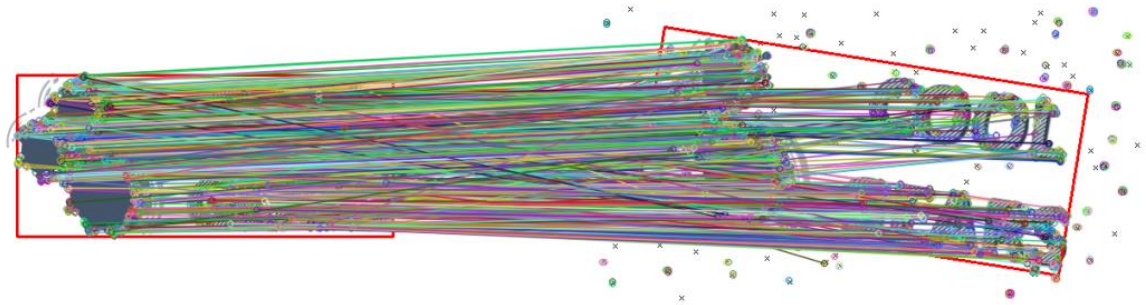
(a) (b)

Slika 4.3. *Prikaz referentne i testne slike (a) referentna slika (b) testna slika*

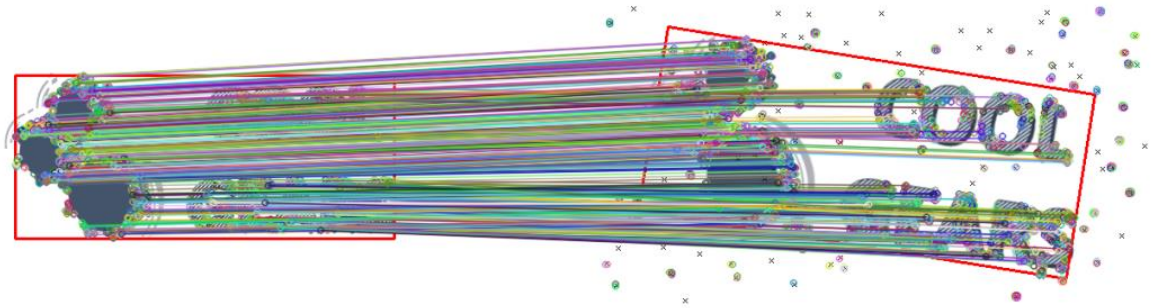


(a) (b)

Slika 4.4. *Prikaz detektiranih ključnih točaka (a) referentnoj slici (b) testnoj slici*



(a)



(b)

Slika 4.5. *Prikaz uparenih ključnih točaka za referentnu i testnu sliku sa slike 4.4. (a) bez RANSAC-a (b) s RANSAC-om*

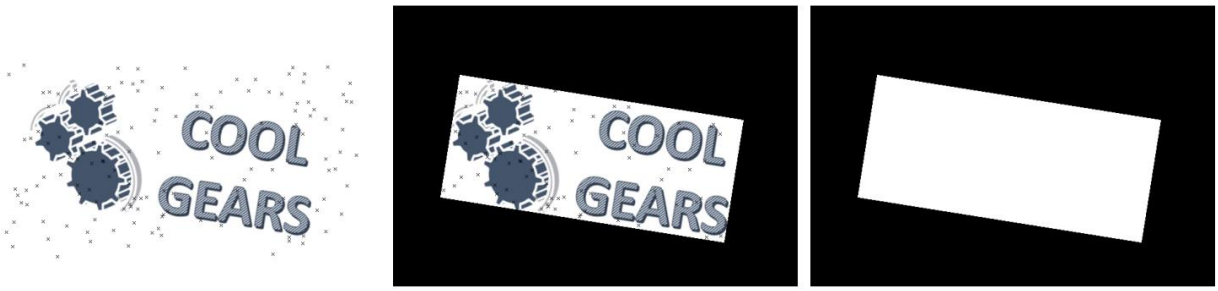
Zbog toga je RANSAC algoritam posebno koristan za slike s puno šuma ili koje su podvrgnute transformacijama kao što su rotacija ili promjene osvjetljenja. Korištenjem RANSAC-a, algoritam DISM postaje mnogo otporniji na ove izazove, osiguravajući da se za daljnju analizu koriste samo pouzdane ključne točke. Ova nadogradnja ne samo da poboljšava točnost usporedbe slika, već također smanjuje potrebu za ručnim prilagođavanjem parametara ili intervencijama, omogućujući algoritmu da automatski prepoznaje i filtrira nepouzdane KT. Korištenje RANSAC-

a unutar DISM algoritma pokazalo se kao učinkovita metoda za filtriranje KT, omogućujući algoritmu da se nosi s različitim izazovima u obradi dinamičkih slika.

4.2.2. Izdvajanje relevantnih dijelova slike

Sljedeći korak poboljšanja DISM algoritma odnosi se na izdvajanje relevantnih dijelova slika, s ciljem povećanja učinkovitosti i točnosti procesa analize slike. U osnovnoj verziji algoritma analizirana je cijela slika, što je često dovodilo do nepotrebnog korištenja resursa na dijelovima slike koji nisu bili relevantni za zadatak usporedbe. To bi moglo učiniti algoritam neučinkovitim, posebno sa složenim slikama gdje su samo određeni dijelovi slike ključni za zadatak. Kako bi se to riješilo, postupak maskiranja slike uveden je u nadograđeni algoritam kako bi obrada bila djelotvornija i učinkovitija. Maskiranje omogućuje algoritmu da se fokusira samo na dijelove slike relevantne za zadatak usporedbe, isključujući druge dijelove. Time se smanjuje količina podataka koje treba analizirati, povećavajući brzinu i preciznost obrade.

Maska se generira na osnovu specifičnih kriterija za izdvajanje dijelova slike. Ovi kriteriji obično uključuju karakteristike slike, kao što su boja, intenzitet ili drugi parametri ovisno o prirodi analize. Definirani kriteriji usmjeravaju primjenu maske na sliku, izolirajući područja od interesa. Primjena maske omogućuje algoritmu da zanemari nevažne dijelove slike, skraćujući vrijeme obrade i poboljšavajući preciznost konačnih rezultata. Na slici 4.6. prikazan je primjer nanošenja maske na sliku, gdje su uklonjeni nebitni dijelovi, a zadržani samo oni dijelovi korisni za daljnju analizu. Vizualizacija jasno pokazuje kako maskiranje omogućuje algoritmu da se fokusira na ključne dijelove slike, što rezultira preciznijim i pouzdanijim rezultatima. U prilogu P.4.2. nalazi se *Python* kod za izdvajanje relevantnih dijelova slike.



(a) (b) (c)

Slika 4.6. *Prikaz maskiranja testne slike nakon pronalaženja traženog objekta (a) testna slika (b) maskirana testna slika (c) maska*

Ovaj postupak značajno poboljšava sposobnost algoritma da učinkovito rukuje velikim i složenim slikama, s većim fokusom na objekte relevantne za zadatak usporedbe. Ova tehnika je učinila DISM fleksibilnijim i učinkovitijim algoritmom jer omogućuje fokusiranje analize na specifična područja slike od interesa koja su relevantna za usporedbu i prilagođava se različitim uvjetima osvjetljenja i složenosti scena. To omogućuje optimalnu obradu slika bez gubitka kvalitete, jer smanjuje vrijeme obrade i resurse potrebne za analizu.

4.2.3. Usporedba kontura objekata

Unaprijeđenje DISM algoritma u segmentu usporedbe kontura objekata zasniva se na detaljnoj analizi oblika unutar slika. Konture objekata pružaju ključne informacije koje omogućuju algoritmu razlikovanje i prepoznavanje različitih oblika te procjenu njihove sličnosti na dvjema slikama. Cijeli proces započinje pretprocesiranjem slike, gdje se prvo primjenjuje pretvorba slike u sliku u skali sive boje pomoću funkcije `cv2.cvtColor`, a zatim se koristi Gaussovo zamućenje (`cv2.GaussianBlur`) s ciljem smanjenja šuma u slici. Ovi koraci su ključni za osiguravanje precizne detekcije kontura. Nakon ovog početnog koraka, slika se podvrgava postupku binarizacije (`cv2.threshold`), što znači da se slika u skali sive boje pretvara u binarnu, pri čemu se pikseli dijele na one iznad i ispod određenog praga. Primjer originalne slike, koja je pretvorena u sliku u skali sive boje, a zatim u binarnu sliku primjenom postupka binarizacije, prikazan je na slici 4.7.

(a)

(b)

(c)

Slika 4.7. *Prikaz koraka pretprocesiranja testne slike (a) testna slika (b) zamućena testna slika u skali sive boje (c) binarna testna slika*

Kombinacija Otsuove metode i adaptivnog postupka binarizacije u ovom koraku omogućava automatsko određivanje optimalnog praga na osnovu histograma slike, dok adaptivni postupak binarizacije omogućuje prilagodbu praga na različitim dijelovima slike. Ova kombinacija osigurava jasnu i preciznu detekciju kontura, čak i u uvjetima neravnomjerne osvjetljenosti ili složenih pozadinskih uzoraka. Kada je slika pretvorena u binarni format, konture objekata unutar slike detektiraju se pomoću funkcije `cv2.findContours`, koja identificira sve konture u slici i prikazuje ih kao niz točaka koje definiraju rubove objekata, što je vidljivo na slici 4.8., a *Python* kod za detekciju kontura je prikazan u prilogu P.4.3.

Slika 4.8. *Prikaz detektiranih kontura na binarnoj testnoj slici*

Nakon detekcije kontura slijedi važan korak filtriranja kontura kako bi se zadržale samo one konture koje su relevantne za daljnju analizu, na način da se koristi površina svake konture kao glavni kriteriji za filtriranje. Izračunava se površina svake konture, a zatim se postavlja prag zasnovan na postotku površine najveće konture pronađene u slici (0,3% površine najveće konture). Prag je određen empirijski iz razloga da izbaci jako male konture koje uglavnom predstavljaju nekakav šum koji se nije uspio izbaciti zamućivanjem slike. Konture koje su manje od postavljenog praga uklanjaju se iz daljnje analize, čime se osigurava da se analiziraju samo najvažnije i najrelevantnije konture. Rezultat ovog filtriranja prikazan je na slici 4.9. Ovo filtriranje

je važno jer smanjuje utjecaj šuma i nebitnih detalja, zato što u suprotnom za rezultat može dati dodatne konture koje bi mogle narušiti preciznost usporedbe, a ne bi pokazivale nikakve bitne elemente.

Slika 4.9. *Prikaz filtriranih kontura na binarnoj testnoj slici*

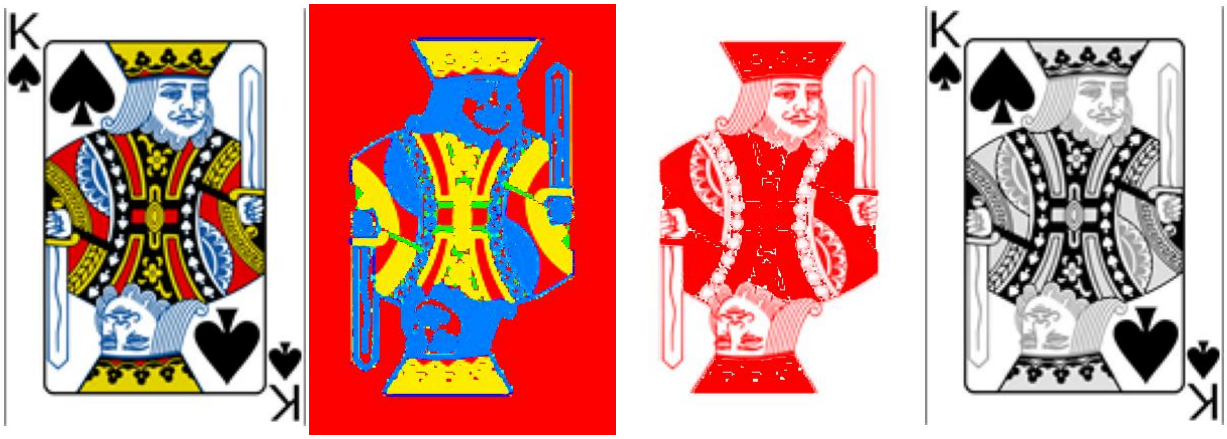
Filtrirane konture se tada koriste u idućem koraku, gdje se uspoređuju konture referentne i testne slike. Ova usporedba je osnova za procjenu sličnosti oblika objekata na različitim slikama. U ovom koraku koristi se funkcija `cv2.matchShapes`, koja uspoređuje dvije konture i vraća vrijednost koja označava razinu njihove sličnosti. *Python* kod kojim se određuje razina sličnosti kontura na referentnoj i testnoj slici prikazan je u prilogu P.4.4. Navedena vrijednost pokazuje koliko se razlikuju dvije konture. Na slici 4.10. su prikazane sve konačne konture koje su detektirane na referentnoj i testnoj slici. Usporedba kontura referentne i testne slike odvija se na način da se traže najbolji parovi za svaku konturu iz referentne slike koristeći funkciju `cv2.matchShapes`. Funkcija za rezultat daje decimalni broj od 0 do 1 (ili se može prikazatu u postotku množeći rezultat sa 100) koji predstavlja razinu sličnosti između kontura. Veći broj (1 ili 100%) ukazuje da se konture u potpunosti razlikuju, odnosno mali broj (0 ili 0%) ukazuje da su konture u potpunosti slične. Nakon što se dobije par dviju kontura, kontura iz testne slike se izbacuje iz liste kontura testne slike za traženje idućih parova. Ako za neku konturu iz liste kontura referentne slike ne postoji niti jedna odgovarajuća kontura u listi kontura testne slike sa sličnosti većom ili jednakom 85%, tada se smatra da ta referentna kontura nema odgovarajući par. Sličnost se računa na način da se od 100% oduzme postotak rezultata funkcije `cv2.matchShapes`, odnosno da se od 1 oduzme rezultat funkcije (te će sličnost biti decimalni broj od 0 do 1, množeći sa 100 dobije se iznos u postotcima). Prag za sličnost od 85% odabran je empirijski da se izbjegne uparivanje kontura objekta s lažnim konturama detektiranih zbog šuma (koje nisu uspjele biti filtrirane). Iako se te konture razlikuju u veličini, funkcija uspoređuje samo njihov oblik. Za ukupnu sličnost kontura između dviju slika računa se aritmetička sredina postotaka uparenih kontura.

(a) (b)

Slika 4.10. *Prikaz detektiranih i filtriranih kontura (a) na referentnoj slici (b) na testnoj slici*

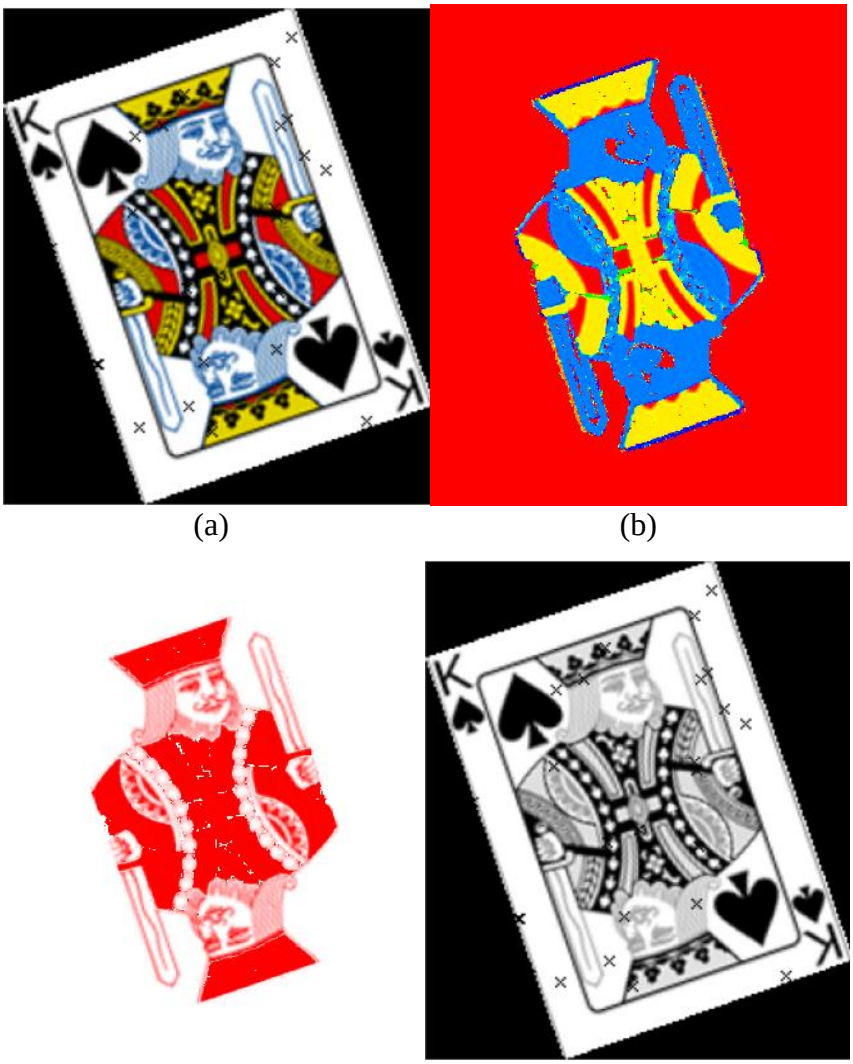
4.2.4. Usporedba boja objekata

Boja objekta često igra važnu ulogu u prepoznavanju i analizi slika, omogućavajući algoritmu da razlikuje objekte koji mogu imati sličan oblik, ali različite boje. Prvi korak u procesu usporedbe boja je ekstrakcija boje objekata iz slika. Postupak započinje pretvorbom slike u odgovarajući prostor boja, najčešće u HSV ili RGB prostor boja, ovisno o zahtjevima analize. Za unaprjeđenje algoritma se koristi HSV prostor boja koji omogućuje algoritmu da preciznije obradi nijanse boje, zasićenost i osvjetljenje, što je posebno korisno kada je osvjetljenje na slikama različito . Ovaj proces je prikazan na slici 4.11. i slici 4.12., gdje je prikazana pretvorba originalne slike u sliku u HSV prostoru boja, pri čemu su prikazane H, S i V komponente zasebno. Na slikama je bijela pozadina u RGB slici (R=255, G=255, B=255) prikazana u crvenoj boji jer nakon konverzije u HSV sliku (0°, 0%, 100%) H komponenta ima vrijednost 0°, što je vrijednost koja odgovara crvenoj boji u HSV prostoru boja. To vrijedi također za bilo koju nijansu sive, uključujući i crnu, jer su R, G i B vrijednosti jednake, odnosno niti jedna boja nije dominantna.



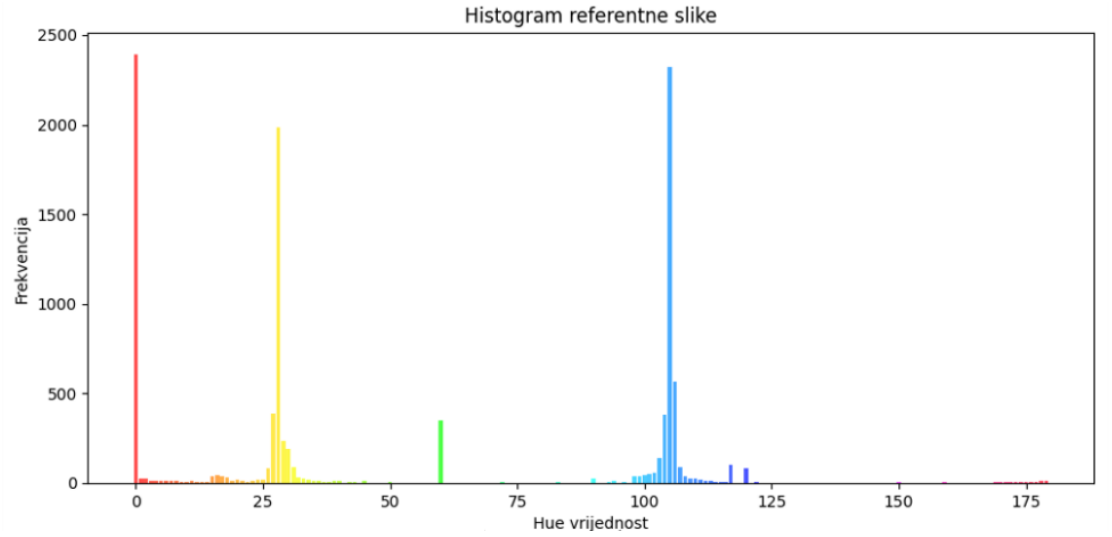
(a) (b) (c) (d)

Slika 4.11. *Prikaz referentne slike u RGB prostoru boja i pojedinačnih H, S, i V kanala u HSV prostoru boja (a) originalna slika u RGB prostoru boja (b) H kanal (c) S kanal (d) V kanal*

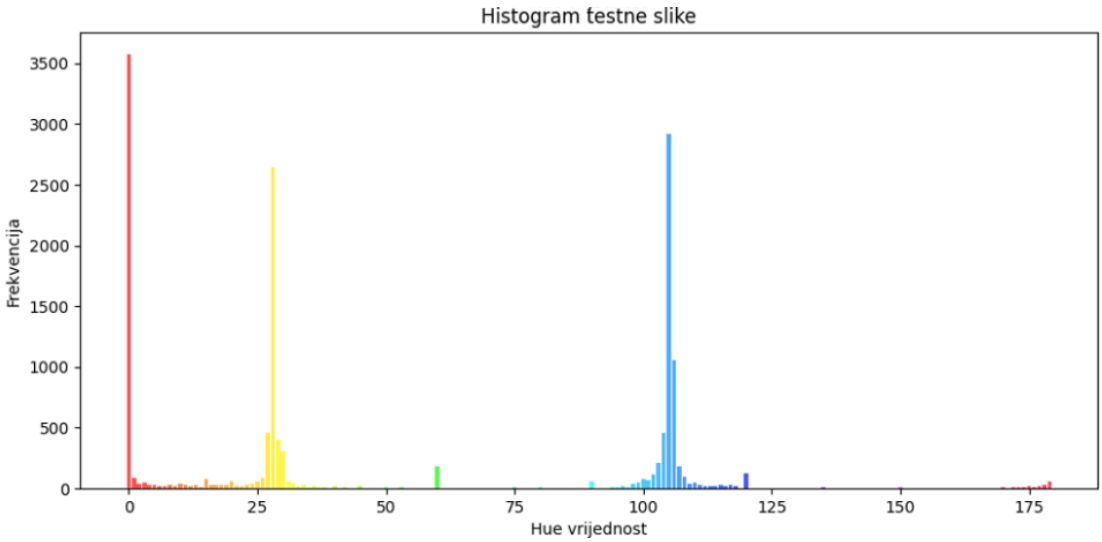


Slika 4.12. *Prikaz testne slike u RGB prostoru boja i pojedinačnih H, S, i V kanala u HSV prostoru boja (a) originalna testna slika u RGB prostoru boja (b) H kanal (c) S kanal (d) V kanal*

Nakon pretvorbe slike, algoritam izdvaja informacije o bojama objekta koje će se koristiti u procesu usporedbe. Boje se analiziraju u određenom dijelu slike, a zatim se konstruira histogram boja, koji prikazuje distribuciju različitih nijansi unutar odabranog područja. Histogram boja predstavlja ključni alat u analizi boja unutar slike. U ovom algoritmu, histograme boja konstruira se pomoću funkcije `cv2.calcHist`, koja analizira sliku i generira histogram na osnovu distribucije piksela u prostoru boja. Histogrami se najčešće računaju u HSV ili RGB prostoru boja, ovisno o zahtjevima aplikacije. Na slici 4.13. prikazan je primjer histograma boja konstruiranog iz H komponente slike u HSV prostoru boja, koji prikazuje frekvenciju pojavljivanja različitih nijansi unutar slike.



(a)



(b)

Slika 4.13. *Prikaz histograma H komponente (a) za referentnu sliku 4.11.b) (b) za testnu sliku 4.12.b)*

Konstrukcija histograma omogućuje algoritmu da kvantificira distribuciju boja unutar slike, pružajući detaljan uvid u to kako su boje raspoređene. Ova informacija je ključna za daljnju usporedbu boja referentne i testne slike. Nakon što su histogrami boja konstruirani, slijedi njihova usporedba. Usporedba histograma boja provodi se pomoću funkcije `cv2.compareHist`, koja izračunava sličnost dvaju histograma koristeći različite metrike, kao što su korelacija (`CV_COMP_CORREL`), metoda *intersect* (`CV_COMP_INTERSECT`) ili metoda *chi-square* (`CV_COMP_CHISQR`). Ove metrike omogućuju algoritmu da kvantificira sličnost između boja dvaju objekata na slikama. Za usporedbu je odabrana korelacija kao metrika za izračunavanje sličnosti, a računa se formulom (2-21) navedenom ranije. U prilogu P.4.5. prikazan je *Python* kod za usporedbu boja referentne i testne slike.

Uvođenje nadogradnji u DISM algoritam, poput filtriranja KT, izdvajanja relevantnih dijelova slike, te usporedbe kontura i boja, značajno je unaprijedilo učinkovitost i točnost algoritma u usporedbi dinamičkih slika. Ove nadogradnje omogućuju bolju prilagodbu specifičnim primjenama, poboljšavajući time robusnost i fleksibilnost algoritma u različitim scenarijima. U sljedećem poglavlju provest će se detaljna verifikacija učinkovitosti ovih poboljšanja koristeći dvije baze slika, CGIAD i CG-STB-DI, čime će se procijeniti ne samo točnost već i vrijeme obrade algoritma prije i nakon implementacije dodatnih funkcionalnosti. Ova procjena bit će ključna za razumijevanje ukupnog učinka nadogradnji na performanse algoritma u stvarnim aplikacijama.

5. VERIFIKACIJA I ANALIZA REZULTATA UNAPRIJEĐENOG ALGORITMA ZA USPOREDBU DINAMIČNIH SLIKA

5.1. Opis baza podataka za verifikaciju rada nadograđenog DISM algoritma

Kako bi se provjerila učinkovitost unaprijeđenog algoritma za usporedbu dinamičkih slika, korištene su dvije specifične baze podataka: CGIAD [13] i CG-STB-DI [13] baza slika. Ove baze pružaju različite izazove i scenarije za testiranje algoritma, omogućujući sveobuhvatnu evaluaciju njegovih performansi.

5.1.1. CGIAD baza slika

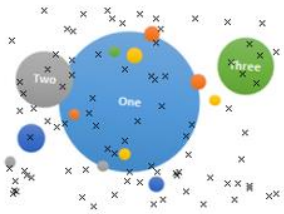
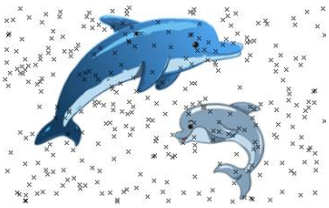
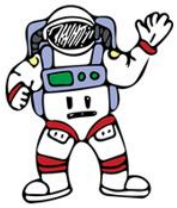
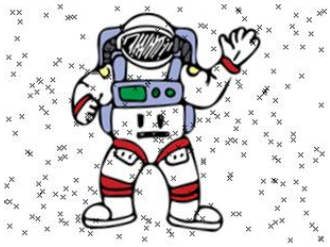
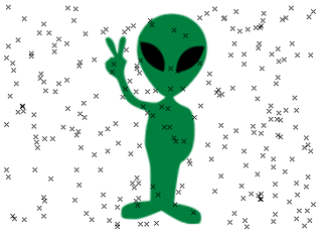
CGIAD baza slika sastoji se od 10 testnih setova, pri čemu svaki set sadrži dvije slike koje dijele vrlo sličan kontekst, ali se razlikuju u sadržaju. Ove slike podvrgnute su različitim transformacijama, uključujući rotaciju, skaliranje, translaciju i prekrivanje objekata na slici. Na taj način simuliraju se uvjeti u kojima bi algoritam trebao prepoznati sličnosti slika, čak i kada su one podvrgnute značajnim promjenama.

Svaki testni set u CGIAD bazi sastoji se od 1000 slika. U svakom setu nalazi se jedna referentna slika, 500 pozitivnih slika (na kojima su objekti isti kao na referentnoj slici, ali su prošli kroz transformacije) i 500 negativnih slika (na kojima su objekti slični, ali se razlikuju u ključnim aspektima). Primijenjene transformacije uključuju rotaciju objekta do 40° u oba smjera, skaliranje za ±20% u odnosu na originalnu veličinu, translaciju do 30 piksela horizontalno i vertikalno, te prekrivanje do 10% površine objekta malim sivim „x-ićima“.

Tablica 5.1. prikazuje primjere slika iz CGIAD baze, uključujući referentne, pozitivne i negativne slike za jedan od testnih setova. Ovi primjeri ilustriraju kako se objekti na slikama mijenjaju kroz različite transformacije, stvarajući izazove za algoritam koji se testira.

Tablica 5.1. *Prikaz primjera slika iz CGIAD baze slika* [13]

Naziv grupe	Referentna slika	Pozitivna slika	Negativna slika
Testna grupa 1			

Testna grupa 2			
Testna grupa 3			
Testna grupa 4			
Testna grupa 5			
Testna grupa 6			
Testna grupa 7			
Testna grupa 8			

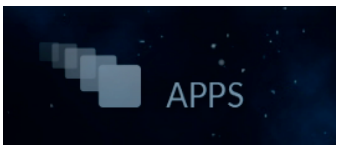
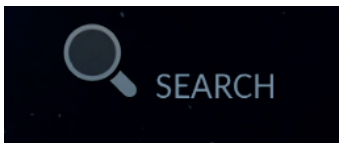
Testna grupa 9			
Testna grupa 10			

5.1.2. CG-STB-DI baza slika

CG-STB-DI baza slika (engl. *Custom Generated Set-Up Box Dynamic Images*) stvorena je posebno za potrebe treniranja i testiranja sijamske neuronske mreže predložene u [13]. Slike korištene u ovoj bazi odabrane su iz STB (engl. *Set-Up Box*) baze slika [11] grafičkog sučelja, a prikazuju različite izbornike u stanju kada su izabrani i kada nisu izabrani. Cilj ove baze je omogućiti procjenu sposobnosti algoritma da razlikuje između izabranih i neizabranih izbornika, te istražiti može li sijamska neuronska mreža poslužiti kao univerzalno rješenje za usporedbu dinamičnih slika. Za izradu CG-STB-DI baze odabrano je 12 slika koje prikazuju različite izbornike u oba stanja – kada je izbornik izabran i kada nije izabran. Ove slike su zatim obrađene pomoću algoritma za generiranje dinamičkih slika, pri čemu je za svaku originalnu sliku stvoreno 170 transformiranih slika. Ukupno je generirano 2040 slika, od kojih polovica prikazuje izbornike u izabranom stanju, a druga polovica u neizabranom. Transformacije koje su primijenjene na ove slike uključuju skaliranje, rotaciju, translaciju po x i y osi te okluziju, čime se simuliraju razni izazovi koje algoritam mora savladati.

Transformacije su generirane unutar specifičnih granica za svaku od njih. Skaliranje varira od 80% do 120% veličine originalne slike, rotacija se kreće između -40° i +40°, translacija po x i y osi varira od -40 do +40 piksela od središta slike, dok okluzija pokriva do 10% ukupne površine slike malim točkicama. Ove transformacije osiguravaju da algoritam mora prepoznati sličnosti unatoč značajnim promjenama u slikama. Primjeri transformiranih slika iz CG-STB-DI baze prikazani su u tablici 5.2. Ove slike ilustriraju kako originalne slike iz STB baze prolaze kroz različite transformacije, pružajući uvid u izazove s kojima se algoritam susreće tijekom analize.

Tablica 5.2. *Prikaz primjera slika CG-STB-DI baze slika* [13]

Naziv grupe	Referentna slika	Testna slika
apps_selected		
apps_not_selected		
media_selected		
media_not_selected		
search_selected		
search_not_selected		
settings_selected		
settings_not_selected		

tv_guide_selected		
tv_guide_not_selected		
vod_selected		
vod_not_selected		

Detaljnije o generiranju navedene baze može se pronaći u znanstvenom radu [13].

5.2. Eksperimentalni postupak za verifikaciju rada nadograđenog DISM algoritma

Eksperimentalni postupak za verifikaciju unaprijeđenog DSIM algoritma proveden je korištenjem dviju različitih baza slika: CGIAD i CG-STB-DI. Glavni cilj ovog postupka bio je testirati sposobnost algoritma da prepozna sličnosti slika koje su prošle kroz različite transformacije.

Unaprijeđeni DSIM algoritam prolazio je kroz nekoliko koraka kako bi identificirao sličnosti slika. Prvi korak uključuje detekciju ključnih točaka (KT) na slici, nakon čega se pokušava generirati granični okvir (engl. *bounding box* - BB) oko objekta. Da bi algoritam mogao nastaviti s provjerom kontura i boja, potrebno je da postotak podudaranja KT s referentnom slikom bude veći od 5%. Taj prag je odabran smisleno jer je BB moguće kreirati sa samo 4 točke koje se mogu upariti čak i kada su objekti potpuno različiti. Ako je taj postotak manji od 5%, algoritam

ne može generirati BB niti dalje obrađivati sliku, što rezultira zabilježenim negativnim ishodom. Ukupni rezultat za svaku sliku izračunava se prema sljedećoj formuli:

$$Rezultat\_sličnosti = 0.4 * (BB\_rezultat) + 0.3 * (Rezultat\_boja) + 0.3 * (rezultat\_kontura) \quad (5\text{-}1)$$

„BB_rezultat“ ovisi o postotku podudaranja (engl. *matches_percentage*) KT s referentnom slikom. „Rezultat_boja“ dobiva se usporedbom histograma boja funkcijom `cv2.compareHist` koristeći metodu korelacije te daje vrijednosti između -1 i 1. Konačni „Rezultat_boja“ se skalira da bude između 0 i 100. „Rezultat_kontura“ izračunava se kao prosječna vrijednost sličnosti parova kontura referentne i testne slike. Ove 3 komponente se mogu prikazati matematički:

$$BB\_rezultat = \begin{cases} 0, & matches\_percentage < 5 \\ 50 + \frac{(matches\_percentage - 5) * (70 - 50)}{5}, & 5 \leq matches\_percentage < 10 \\ 70 + \frac{(matches\_percentage - 10) * (100 - 70)}{10}, & 10 \leq matches\_percentage < 20 \\ 100, & 20 \leq matches\_percentage \end{cases} \quad (5\text{-}2)$$

$$Rezultat\_boje = (rezultat\_usporedbe\_histograma + 1) * 50 \quad (5\text{-}3)$$

$$Rezultat\_kontura = \frac{\sum(1 - rezultat\_usporedbe\_kontura) * 100}{broj\_referentnih\_kontura} \quad (5\text{-}4)$$

Sve tri varijable su u rasponu od 0 do 100, prikazuju se postotcima kako bi bilo lakše dobiti konačan rezultat. Ako sve komponente zadovoljavaju postavljeni prag točnosti od minimalno 50%, rezultat se klasificira kao točno prepoznati pozitivan ishod (TP). U suprotnom slučaju, rezultat se bilježi kao negativan (FN).

5.2.1. Verifikacija na CGIAD bazi slika

Verifikacija na CGIAD bazi slika bila je usmjerena na procjenu sposobnosti unaprijeđenog DSIM algoritma da prepozna sličnosti slika koje su prošle kroz različite transformacije, poput rotacije, skaliranja, translacije i okluzije.

Rezultati verifikacije prikazani su u Tablici 5.3. U svim testnim grupama algoritam je uspješno prepoznao sličnosti slika, bez ijednog lažno pozitivnog (FP) ili lažno negativnog (FN) rezultata. Svi zabilježeni rezultati bili su iznad praga od 50%, što potvrđuje visoku preciznost algoritma. Srednje vrijednosti pozitivnih rezultata kretale su se između 80,07% i 95,61%, što

ukazuje na različite stupnjeve složenosti među testnim grupama, ali dosljedno pokazuje pozitivne ishode. Sve negativne slike su imale rezultat od 0%, bile su previše različite te nisu imale dovoljno uparenih KT da algoritam nastavi s daljnim uspoređivanjem, što znači da negativne slike nisu zadovoljavale postavljene uvjete i nisu imale ni približnu šansu proći verifikaciju.

Tablica 5.3. *Prikaz rezultata unaprijeđenog DISM algoritma na CGIAD bazi*

Naziv grupe slike	TP	FP	TN	FN	Srednja vrijednost pozitivno detektiranih rezultata
Testna grupa 1	500	0	500	0	92,40
Testna grupa 2	500	0	500	0	87,05
Testna grupa 3	500	0	500	0	80,07
Testna grupa 4	500	0	500	0	85,03
Testna grupa 5	500	0	500	0	94,24
Testna grupa 6	500	0	500	0	94,22
Testna grupa 7	500	0	500	0	93,96
Testna grupa 8	500	0	500	0	89,03
Testna grupa 9	500	0	500	0	80,19
Testna grupa 10	500	0	500	0	95,61

Osim točnosti prepoznavanja, dodatno je analizirano i potrebno vrijeme obrade za svaku testnu grupu, kao što je prikazano u tablici 5.4. Rezultati pokazuju da unaprijeđeni DISM algoritam općenito zahtijeva neznatno više vremena za obradu u usporedbi s originalnim algoritmom. Ovo je vidljivo u svim testnim grupama, gdje su vremena obrade za cijelu testnu grupu za unaprijeđeni algoritam nešto veća, iako ne drastično. Najveća razlika u vremenu obrade zabilježena je u Testnoj grupi 2, gdje je unaprijeđeni algoritam bio za približno dvije minute sporiji od originalnog. Svaka testna grupa ima drugačije objekte na slikama te su određene testne grupe teže za obradu od ostalih što je vidljivo na tablici 5.4. Nekim grupama je potrebno oko pola sata za obradu dok je drugim grupama potrebno više od dva sata.

Ovo povećanje vremena može se pripisati dodatnim koracima koji su uvedeni radi poboljšanja točnosti usporedbe, kao što su filtriranje KT pomoću RANSAC algoritma te napredna analiza kontura i boja objekata. Iako je vrijeme obrade nešto malo duže, unaprijeđeni algoritam pruža precizniju usporedbu slika, što je ključno za aplikacije koje zahtijevaju visok stupanj točnosti.

Tablica 5.4. *Prikaz potrebnog vremena obrade originalnog i unaprijeđenog DISM algoritma te algoritma zasnovanog na sijamskoj neuronskoj mreži na CGIAD bazi*

Naziv grupe slike	Potrebno vrijeme		
	Originalnog DISM algoritma	Unaprijeđenog DISM algoritma	Algoritam zasnovan na sijamskoj neuronskoj mreži
Testna grupa 1	2:28:00	2:28:54	0:08:54
Testna grupa 2	3:33:45	3:36:03	0:08:41
Testna grupa 3	0:39:52	0:40:34	0:08:45
Testna grupa 4	1:13:40	1:14:48	0:08:30
Testna grupa 5	0:21:51	0:22:03	0:07:41
Testna grupa 6	0:33:15	0:33:35	0:08:45
Testna grupa 7	0:46:57	0:47:46	0:08:41
Testna grupa 8	2:27:21	2:28:13	0:08:53
Testna grupa 9	0:28:47	0:29:47	0:09:01
Testna grupa 10	0:29:33	0:31:11	0:08:43
Ukupno vrijeme	13:03:01	13:20:54	01:26:34

5.2.2. Verifikacija na CG-STB-DI bazi slika

Verifikacija na CG-STB-DI bazi slika obuhvatila je testiranje sposobnosti algoritma da prepozna sličnosti između različitih stanja izbornika u STB grafičkom sučelju. Algoritam je primijenio isti niz koraka kao i za CGIAD bazu: prvo je detektirao KT, zatim pokušao generirati BB, a na kraju provjeravao konture i boje.

Rezultati verifikacije prikazani su u Tablici 5.5. Unaprijeđeni DSIM algoritam pokazao je visoku točnost u prepoznavanju sličnosti slika u svim testnim grupama. Svi zabilježeni rezultati imali su „Rezultat_sličnosti“ iznad 50% što ih je klasificiralo kao TP, bez zabilježenih lažno negativnih rezultata (FN), što dodatno potvrđuje pouzdanost algoritma u ovom specifičnom kontekstu.

Tablica 5.5. *Prikaz rezultata unaprijeđenog i originalnog DISM algoritma na CG-STB-DI bazi*

	Unaprijeđeni DISM algoritam			Originalni DISM algoritam		
Naziv grupe slika	TP	FN	Srednja vrijednost rezultata	TP	FN	Srednja vrijednost rezultata
apps_selected	170	0	93,72	170	0	80,70
apps_not_selected	170	0	86,66	170	0	78,77
media_selected	170	0	96,00	170	0	97,75
media_not_selected	170	0	88,00	170	0	88,52
search_selected	170	0	98,51	170	0	88,43
search_not_selected	170	0	85,36	170	0	86,14
settings_selected	170	0	98,41	170	0	87,27
settings_not_selected	170	0	88,88	170	0	83,67
tv_guide_selected	170	0	91,93	170	0	89,06
tv_guide_not_selected	170	0	83,56	170	0	87,10
vod_selected	170	0	93,26	170	0	86,71
vod_not_selected	170	0	83,40	170	0	77,31

5.3. Usporedba rezultata unaprijeđenog i originalnog DISM algoritma te algoritma zasnovanog na sijamskoj neuronskoj mreži

Usporedba rezultata originalnog i unaprijeđenog DISM algoritma te algoritma zasnovanog na sijamskoj neuronskoj mreži na bazama slika CGIAD i CG-STB-DI pokazuje poboljšanja koja su ostvarena s unaprijeđenim algoritmom, posebno u kontekstu dinamičkih slika.

Originalni DISM algoritam, kada je testiran na CGIAD bazi slika, postigao je izuzetnu točnost od 99,96% [11], čime je potvrdio svoju sposobnost da precizno prepozna sličnosti između slika koje su prošle kroz različite transformacije. Ovaj rezultat je gotovo savršen, što nije iznenađujuće s obzirom na prirodu slika u CGIAD bazi i način na koji je originalni algoritam optimiziran za rad s takvim podacima. Povećana točnost postignuta je korištenjem unaprijeđenog algoritma na istoj bazi slika, a iznosi 100%.

Algoritam zasnovan na sijamskoj neuronskoj mreži, testiran na CGIAD bazi slika postigao je točnost od 93,66%, čime je pokazao da sijamska neuronska mreža predstavlja potencijalno

rješenje za usporedbu dinamičkih slika. Međutim kao što je navedeno u radu [13], navedeni algoritam trenutno nije najbolja opcija za usporedbu dinamičkih slika.

Međutim, pri testiranju originalnog DISM algoritma na CG-STB-DI bazi slika, iako je zadržao visoku točnost od 100% u prepoznavanju ispravnih sličnosti slika, pokazalo se da su srednje vrijednosti rezultata u nekim slučajevima bile niže u usporedbi s unaprijeđenim algoritmom. Na primjer, za skupinu slika "apps_not_selected", originalni algoritam je ostvario srednju vrijednost od 78,77%, dok je unaprijeđeni algoritam postigao bolji rezultat od 86,66%. Ove razlike u srednjim vrijednostima ukazuju na to da, iako je originalni algoritam bio vrlo učinkovit, unaprijeđeni algoritam donosi poboljšanja u preciznosti, posebno u scenarijima gdje su slike podvrgnute složenijim transformacijama. Rezultati iz tablice, prikazani u Tablici 5.5. za unaprijeđeni algoritam i za originalni algoritam, jasno pokazuju ove razlike.

Uz sve navedene rezultate, ključno je istaknuti osnovne razlike u principima rada između originalnog DISM algoritma i njegove unaprijeđene verzije. Originalni DISM algoritam oslanja se isključivo na detekciju i usporedbu ključnih točaka (KT) između referentne slike i slike koja se analizira. Ovaj pristup omogućava algoritmu postizanje visoke točnosti kroz precizno prepoznavanje i usporedbu tih točaka, što je osobito učinkovito u slučajevima gdje su objekti na slikama relativno slični ili gdje transformacije nisu previše složene. Međutim, ovaj pristup može biti ograničen u situacijama gdje se promjene na slici ne mogu jednostavno opisati samo pomoću ključnih točaka. To može rezultirati nižim srednjim vrijednostima, kao što je zabilježeno u rezultatima za CG-STB-DI bazu slika.

Također jedan od bitnijih faktora za usporedbu navedenih algoritama je vrijeme obrade na CGIAD bazi slika. Originalnom DISM algoritmu je za obradu bilo potrebno ukupno 13 sati, 3 minute i 1 sekunda, dok je unaprijeđenom algoritmu, u usporedbi s originalnim, bilo potrebno nešto više vremena. Točnije, unaprijeđenom algoritmu je za obradu bilo potrebno 13 sati, 20 minuta i 54 sekunde. Navedeno povećanje vremena za unaprijeđeni algoritam rezultat je dodatnih metoda koje se provode prilikom obrade slika. Algoritmu zasnovanom na sijamskoj neuronskoj mreži je za obradu bilo potrebno ukupno 1 sat, 26 minuta i 34 sekunde. Navedeno vrijeme potrebno za obradu je značajno manje uspoređujući s originalnim i unaprijeđenim algoritmom. Međutim, algoritam zasnovan na sijamskoj neuronskoj mreži ima najmanju točnost uspoređujući s navedenim algoritmima.

Unaprijeđeni DISM algoritam proširuje ovaj osnovni pristup dodajući analizu boje i kontura uz detekciju KT. Ovaj prošireni pristup omogućuje algoritmu da prepoznaje sličnosti ne

samo na osnovu prostornog rasporeda ključnih točaka, već i na osnovu vizualnih karakteristika objekata, poput boje i oblika. Boja i konture pružaju dodatne informacije koje su ključne u situacijama gdje KT same po sebi nisu dovoljne za precizno prepoznavanje sličnosti. Na primjer, u slučajevima kada su objekti na slici izloženi promjenama u osvjetljenju ili su djelomično zaklonjeni, boja i konture mogu pružiti dodatne tragove koji pomažu algoritmu da bolje razlikuje i prepozna slične objekte.

6. ZAKLJUČAK

Ovaj diplomski rad usmjeren je na razvoj i testiranje unaprijeđenog algoritma za usporedbu dinamičkih slika zasnovanog na DISM algoritmu, s ciljem povećanja učinkovitosti i preciznosti. Proširenje metodologije uključilo je analizu i filtriranje ključnih točaka (KT), kontura i boje objekata, čime je algoritam značajno unaprijeđen. Verifikacija je uspješno provedena na dvjema različitim bazama slika, CGIAD i CG-STB-DI, a dobiveni rezultati uspoređeni su s onima ostvarenim korištenjem originalnog DISM algoritma i algoritma zasnovanog na sijamskoj neuronskoj mreži. Eksperimenti su potvrdili da unaprijeđeni algoritam postiže visoku razinu točnosti u prepoznavanju sličnosti slika, unatoč složenim transformacijama primijenjenima na slikama. Na CGIAD bazi slika, algoritam zasnovan na sijamskoj neuronskoj mreži ostvario je točnost od 93,66%, originalni DISM algoritam ostvario je točnost od 99,96%, a unaprijeđeni algoritam je postigao točnost od 100%. Na CG-STB-DI bazi slika, gdje originalni DISM algoritam nije prethodno testiran, originalni algoritam postigao je 100% točnosti u prepoznavanju objekata. Unaprijeđeni algoritam pokazuje istu točnost, ali s poboljšanim srednjim vrijednostima rezultata.

Glavna razlika između originalnog i unaprijeđenog algoritma leži u načinu analize slika. Dok se originalni DISM algoritam oslanja isključivo na detekciju i usporedbu ključnih točaka, unaprijeđeni algoritam proširuje ovu analizu uključivanjem dodatnih faktora kao što su boja, konture objekata i filtriranje KT korištenjem RANSAC algoritma. Ovaj prošireni pristup omogućuje algoritmu bolje prepoznavanje sličnosti u situacijama gdje ključne točke same po sebi nisu dovoljne za precizno prepoznavanje.

Rezultati ovog rada pokazuju da je unaprijeđeni DISM algoritam pouzdano rješenje za usporedbu dinamičnih slika, pružajući precizne rezultate u raznolikim scenarijima. Integracijom višestrukih faktora analize, unaprijeđeni algoritam postaje prilagodljiviji i robusniji za primjenu u različitim područjima, uključujući automatizirano testiranje softvera, analizu video sadržaja i digitalnu forenziku. Ovaj rad predstavlja značajan doprinos u području računalnog vida i analize slika, potvrđujući učinkovitost i primjenjivost unaprijeđenog DISM algoritma. Iako su postignuti značajni rezultati, rad također otvara vrata budućim istraživanjima usmjerenim na daljnju optimizaciju algoritma, posebno u smislu povećanja brzine obrade i smanjenja računalnih resursa potrebnih za njegovu primjenu. S obzirom na postignute rezultate, unaprijeđeni algoritam ima potencijal postati standard u usporedbi dinamičkih slika, posebno u aplikacijama gdje je visoka točnost ključna.

LITERATURA

[1] Introduction to SIFT, https://medium.com/@deepanshut041/introduction-to-sift-scale-invariant-feature-transform-65d7f3a72d40 , pristup ostvaren 26.07.2024.

[2] Introduction to ORB, https://medium.com/@deepanshut041/introduction-to-orb-oriented-fast-and-rotated-brief-4220e8ec40cf, pristup ostvaren 26.07.2024.

[3] Z. Othman, H. Habibollah, R. A. K. Mohammed i R. Mohammed, Comparison of Canny and Sobel Edge Detection in MRI Images, 133-136, 2009.

[4] A. Gabillon, M. Loesdau, S. Chabrier, Hue and Saturation in the RGB Color Space, Lecture Notes in Computer Science, Vol. 8509, str. 203-212, 2014.

[5] K Means, https://stanford.edu/~cpiech/cs221/handouts/kmeans.html, pristup ostvaren 27.07.2024.

[6] Computer Vision: Advanced Image Processing all You Need, https://medium.com/@koushikkushal95/beyond-pixels-mastering-advanced-image-processing-in-computer-vision-e903535440a5, pristup ostvaren 27.07.2024.

[7] What is the difference between k-means and mean shift clustering algorithms?, https://eitca.org/artificial-intelligence/eitc-ai-mlp-machine-learning-with-python/clustering-k-means-and-mean-shift/k-means-with-titanic-dataset/examination-review-k-means-with-titanic-dataset/what-is-the-difference-between-k-means-and-mean-shift-clustering-algorithms/, pristup ostvaren 27.7.2024.

[8] How to Compute the Similarity of Colours, https://www.baeldung.com/cs/compute-similarity-of-colours, pristup ostvaren 28.07.2024.

[9] Y. Furgala, A. Velhosh, S. Velhosh i B. Rusyn, Using Color Histograms for Shrunk Images Comparison, 12. International Conference on Electronics and Information Technologies (ELIT), str. 130-133, svibanj, 2021.

[10] M. Pul, Usporedba dinamičnih slika zasnovana na izdvajanju i usporedbi ključnih točaka. Diplomski rad, Osijek, 2016.

[11] M. Pul, V. Peković, M. Vranješ i R. Grbić, Automatic Functionality Verification of Hybrid Set-Top Boxes With Dynamic User Interface, IEEE Transactions on Consumer Electronics, Vol. 64, str. 409-417, rujan, 2018.

[12] W. Jiansheng, P. Shuguo, G. Wang i Z. Tao, A dynamic object filtering approach based on object detection and geometric constraint between frames, IET Image Processing, Vol. 16, str. 1636-1647, 2022.

[13] D. Steiner, Korištenje neuronskih mreža za uspoređivanje dinamičnih slika. Diplomski rad, Osijek, 2022.

[14] S. M. Jameel, M. A. Hashmani, M. Rehman i A. Budiman, An Adaptive Deep Learning Framework for Dynamic Image Classification in the Internet of Things Environment, Perceptual Deep Learning in Image Processing and Computer Vision, Vol. 20, listopad, 2020.

[15] G. Kanojia i S. Raman, Simultaneous Detection and Removal of Dynamic Objects in Multi-view Images, Winter Conference on Applications of Computer Vision (WACV), str. 1979-1988., USA, ožujak, 2020.

[16] A. Ibrahim i C. Valli, Image similarity using dynamic time warping of fractal features, 13th Australian Digital Forensics Conference, str. 111–122., Edith Cowan University Joondalup Campus, Perth, Western Australia, prosinac, 2015.

[17] T. Marošević, The Hausdorff distance between some sets of points, Mathematical Communications, Vol. 23, str. 247–257, Osijek, 2017.

[18] D. Steiner, M. Vranješ, M. Pul i M. Herceg, Dynamic Images Comparison using Siamese Neural Network, International Conference on Software, Telecommunications and Computer Networks (SoftCOM), str. 1-6, Split, Hrvatska, rujan, 2023.

[19] R. Rajan i S. N. Kumar, Gauss Gradient Algorithm for Edge Detection in Retinal Optical Coherence Tomography Images, Procedia Computer Science, Vol. 218, str. 1014-1026, 2023.

[20] B. A. Lungisani, C. K. Lebekwe, A. M. Zungeru i A. Yahy, The current state on usage of image mosaic algorithms, Scientific African, Vol. 18, listopad, 2022.

[21] C. Shu, X. Ding i C. Fang, Histogram of the Oriented Gradient for Face Recognition, Tsinghua Science & Technology, Vol. 16, str. 216-224, travanj, 2011.

[22] G. Wang, S. Guo, L. Han i A. Baris, CekderiTwo-dimensional reciprocal cross entropy multi-threshold combined with improved firefly algorithm for lung parenchyma segmentation of COVID-19 CT image, Biomedical Signal Processing and Control, Vol. 78, lipanj, 2022.

[23] C. Maple, Geometric Design and Space Planning Using the Marching Squares and Marching Cube Algorithms, u International Conference on Geometric Modeling and Graphics, str. 90-95, IEEE, London, UK, kolovoz, 2003.

[24] R. O. Duda i P. E. Hart, Use of the Hough Transformation To Detect Lines and Curves in Pictures, Communications of the ACM, Vol. 15, str. 11-15, siječanj, 1972.

[25] Y. Cheng, Mean Shift, Mode Seeking, and Clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17, str. 790-799, kolovoz, 1995.

[26] S. Baker i I. Matthews, Lucas-Kanade 20 Years On: A Unifying Framework: Part 1, International Journal of Computer Vision, Vol. 56, str. 221-255, siječanj, 2004.

[27] O. A. Omer, Region-based Horn-Schunck Optical Flow Estimation, Japan-Egypt Conference on Electronics, Communications and Computers, str. 73-78, Egypt, 2012.

SAŽETAK

U ovom diplomskom radu razvijen je i testiran unaprijeđeni algoritam za usporedbu dinamičkih slika, zasnovan na DISM algoritmu. Glavni cilj bio je poboljšati točnost prepoznavanja sličnosti slika proširivanjem metodologije koja obuhvaća detekciju ključnih točaka, analizu kontura i boje objekata. Verifikacija algoritma provedena je na CGIAD i CG-STB-DI bazama slika, a rezultati su pokazali da unaprijeđeni algoritam zadržava visoku razinu točnosti originalnog DISM algoritma i pruža dodatnu preciznost u složenim scenarijima. Točnost unaprijeđenog algoritma je 100% za CGIAD i CG-STB-DI baze slika, a vrijeme obrade na CGIAD bazi slika se, u odnosu na originalni DISM algoritam, povećalo za 2,23%. Ključna razlika između algoritma zasnovanog na sijamskoj neuronskoj mreži, originalnog i unaprijeđenog algoritma leži u sveobuhvatnijoj analizi slika koja uključuje više faktora, što rezultira boljim performansama u različitim uvjetima. Najmanje vrijeme, ali zato i najmanju točnost, postigao je algoritam zasnovan na sijamskoj neuronskoj mreži. S obzirom na originalni algoritam, unaprijeđeni algoritam daje veću točnost, ali i nešto veći utrošak vremena za obradu CGIAD baze slika. Ovaj rad predstavlja značajan doprinos području računalnog vida, postavljajući temelje za daljnji razvoj i primjenu algoritama za usporedbu dinamičnih slika u raznim industrijama.

Ključne riječi: usporedba dinamičnih slika, DISM algoritam, ključne točke, analiza kontura, analiza boje, računalni vid, algoritmi za prepoznavanje sličnost

COMPARISON OF DYNAMIC IMAGES BASED ON OBJECT CHARACTERISTICS

ABSTRACT

In this thesis, an improved algorithm for comparing dynamic images, based on the DISM algorithm, was developed and tested. The main goal was to improve the accuracy of image similarity recognition by extending the methodology to include keypoint detection, contour analysis, and object color. Algorithm verification was performed on CGIAD and CG-STB-DI image databases, and the results showed that the improved algorithm maintains the high level of accuracy of the original DISM algorithm and provides additional precision in complex scenarios. The accuracy of the improved algorithm is 100% for the CGIAD and CG-STB-DI image bases, and the processing time on the CGIAD image base, compared to the original DISM algorithm, has increased by 2.23%. The key difference between the Siamese neural network-based algorithm, the original algorithm, and the improved algorithm lies in the more comprehensive image analysis that includes multiple factors, resulting in better performance under different conditions. The least time, but therefore the least accuracy, was achieved by an algorithm based on a Siamese neural network. With regard to the original algorithm, the improved algorithm provides higher accuracy, but also a slightly higher time consumption for processing the CGIAD image database. This work represents a significant contribution to the field of computer vision, laying the foundations for further development and application of dynamic image comparison algorithms in various industries.

Keywords: comparison of dynamic images, DISM algorithm, key points, contour analysis, color analysis, computer vision, similarity recognition algorithms

ŽIVOTOPIS

Luka Lučić rođen je 28.12.1999. godine u Slavonskom Brodu. Pohađao je Osnovnu školu Josip Kozarac Slavonski Šamac. Nakon osnovne škole upisuje Tehničku školu Slavonski Brod smjer elektrotehničar. Završetkom srednje škole upisao je preddiplomski sveučilišni studij Elektrotehnike na FERIT-u, gdje se opredjeljuje za blok Komunikacije i informatika. Godine 2022. završava preddiplomski studij i nastavlja isti smjer Komunikacija i informatika, na istom fakultetu. Tijekom završne godine studija, postaje stipendist tvrtke TTTechAuto d.o.o.

PRILOZI

Prilog P.4.1. *Python* kod za filtriranje ključnih točaka korištenjem RANSAC algoritma

```
def get_matches(self, ref_image: FeatureImage, test_image: FeatureImage) ->
tuple:
        matches = self._matcher.match(ref_image.descriptors,
test_image.descriptors)

        if self.configuration.filter_matches:
            matched_keypoints_ref = [ref_image.keypoints[m.queryIdx].pt for
m in matches]
            matched_keypoints_test = [test_image.keypoints[m.trainIdx].pt
for m in matches]

            ref_points = np.float32(matched_keypoints_ref).reshape(-1, 1,
2)
            test_points = np.float32(matched_keypoints_test).reshape(-1, 1,
2)
            H, mask = cv2.findHomography(ref_points, test_points,
cv2.RANSAC, 10.0)
            matches = [m for m, inlier in zip(matches, mask) if inlier]

            referent_group = self._get_referent_group(ref_image,
test_image, matches)

            matches = self._remove_false_matches(
                ref_image.keypoints, test_image.keypoints, referent_group,
self.configuration.max_desc_distance, matches
            )
        return matches
```

Prilog P.4.2. *Python* kod za izdvajanje područja od interesa

```
def _process_bounding_box(self, ref_keypoints: array, test_keypoints:
array, ref_image: FeatureImage, test_image: FeatureImage):
   ...
     r_x, r_y, r_w, r_h = cv2.boundingRect(all_keypoints)
     cv_ref_image = ref_image.cv_image
     ref_bbox_region = cv_ref_image[r_y:r_y + r_h, r_x:r_x + r_w]

     M, _ = cv2.estimateAffinePartial2D(ref_keypoints, test_keypoints)
     if M is None:
        return ref_bbox_region, None, (r_x, r_y), (r_x + r_w, r_y + r_h),
None, None
     M_inv = cv2.invertAffineTransform(M)
     transformed_image = cv2.warpAffine(test_image.cv_image, M_inv,
(ref_image.cv_image.shape[1], ref_image.cv_image.shape[0]))
     cropped_warped_image = transformed_image[r_y:r_y + r_h, r_x:r_x + r_w]

     ref_corners = np.array([[r_x, r_y], [r_x + r_w, r_y], [r_x + r_w, r_y
+ r_h], [r_x, r_y + r_h]], dtype=np.float32).reshape(-1, 1, 2)
     transformed_corners = cv2.transform(ref_corners, M)
     mask = np.zeros(test_image.cv_image.shape[:2], dtype=np.uint8)
     cv2.fillPoly(mask, [np.int32(transformed_corners)], 255)
     masked_test_image = cv2.bitwise_and(test_image.cv_image,
test_image.cv_image, mask=mask)

     x, y, w, h = cv2.boundingRect(np.int32(transformed_corners))
     cropped_test_image = masked_test_image[y:y + h, x:x + w]

     if matches_percentage < 5 or test_bbox_area == 0:
        return ref_bbox_region, None, (r_x, r_y), (r_x + r_w, r_y + r_h),
None, None

     return ref_bbox_region, cropped_test_image, (r_x, r_y), (r_x + r_w,
r_y + r_h), transformed_corners, cropped_warped_image
```

Prilog P.4.3. *Python* kod za detekciju kontura na slici

```
def detect_contours(self, image: np.ndarray, min_area: int = 65) -> list:
   preprocessed = self.preprocess_image(image)
   contours, _ = cv2.findContours(preprocessed, cv2.RETR_CCOMP,
cv2.CHAIN_APPROX_SIMPLE)

   filtered_contours = [cnt for cnt in contours if cv2.contourArea(cnt) >=
min_area]

   return filtered_contours

def preprocess_image(self, image: np.ndarray) -> np.ndarray:
   gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

   gray = self.apply_combined_denoising(gray)
   gray = cv2.GaussianBlur(gray, (5, 5), 0)

   threshold = self.apply_combined_threshold(gray)
   threshold = self.remove_small_contours(threshold, min_area=70)
   return threshold
```

Prilog P.4.4. *Python* kod za usporedbu kontura referentne i testne slike

```
def _contour_matching_hausdorff(self, ref_contours: list, test_contours:
list, threshold: float = 0.85) -> float:
        def compute_hausdorff(contour1, contour2):
            return max(directed_hausdorff(contour1, contour2)[0],
directed_hausdorff(contour2, contour1)[0])

        if not ref_contours or not test_contours:
            return 0.0
        matched_scores = []
        used_test_contours = set()
        for ref_contour in ref_contours:
            best_match_score = 0.0
            best_match_index = -1
            for i, test_contour in enumerate(test_contours):
                if i in used_test_contours:
                    continue

                hausdorff_distance = compute_hausdorff(ref_contour[:, 0,
:], test_contour[:, 0, :])
                max_possible_distance = 100.0
                similarity_score = max(0, 100 - (hausdorff_distance /
max_possible_distance) * 100)

                if similarity_score > best_match_score:
                    best_match_score = similarity_score
                    best_match_index = i

            if best_match_score >= threshold * 100:
                matched_scores.append(best_match_score)
                used_test_contours.add(best_match_index)

        average_score = sum(matched_scores) / len(ref_contours)
        return average_score
```

Prilog P.4.5. *Python* kod za usporedbu boja referentne i testne slike

```
def _histogram_HSV(self, ref_bbox: any, test_bbox: any, mask1=None,
mask2=None) -> float:
        if test_bbox is None or test_bbox.size == 0:
            logging.error("Test bounding box is empty.")
            return 0.0

        image1_hsv = cv2.cvtColor(ref_bbox, cv2.COLOR_BGR2HSV)
        image2_hsv = cv2.cvtColor(test_bbox, cv2.COLOR_BGR2HSV)

        hist_img1_hue = cv2.calcHist([image1_hsv], [0] , [180], [0, 180])
        hist_img2_hue = cv2.calcHist([image2_hsv], [0] , [180], [0, 180])

        similarity_score = cv2.compareHist(hist_img1_hue, hist_img2_hue,
cv2.HISTCMP_CORREL) + 1
        return similarity_score * 50
```