Okvir za testiranje uspješnosti detekcije objekata u video isječcima

Češić, Nikolina

Undergraduate thesis / Završni rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:168156

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

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

Sveučilišni studij

OKVIR ZA TESTIRANJE USPJEŠNOSTI DETEKCIJE OBJEKATA U VIDEO ISJEČCIMA

Završni rad

Nikolina Češić

Osijek, 2017. godina.

SADRŽAJ

1. UVOD 1
1.1. Zadatak završnog rada 1
2. RAD S VIDEO ISJEČCIMA 2
2.1. C# programski jezik i EMGU CV 2
2.2. .NET Framework 3
2.2.1. CLR (Common Language Runtime) 4
2.2.2. .NET Framework biblioteka klasa 5
2.3. Microsoft Visual Studio 6
2.3.1. Visual Studio 2013. 7
2.4. Učitavanje i prikaz videa 7
3. TESTIRANJE USPJEŠNOSTI DETEKCIJE OBJEKATA U VIDEO ISJEČCIMA 10
3.1. Zapis ljudske percepcije specifičnih objekata 10
3.2. Zapis računalne detekcije specifičnih objekata 12
4. EKSPERIMENTALNI REZULTATI 15
4.1. Jednostavan algoritam za detekciju objekata crvene boje 15
4.2. Usporedba čovjeka i računala u detekciji 15
5. ZAKLJUČAK 20
LITERATURA 21
SAŽETAK 22
ABSTRACT 23
ŽIVOTOPIS 24

1. UVOD

Tema završnog rada je okvir za testiranje uspješnosti detekcije objekata u video isječcima napravljen u C# programskom jeziku i EmguCV-u. Na samome početku govori se općenito o C# programskom jeziku, za što se koristi i po čemu je bolji od drugih programskih jezika te o EmguCV-u, točnije, što je EmguCV i koje su njegove prednosti. Nadalje, govori se o detekciji objekata i o prikazu rezultata uspješnosti čovjekove i računalne detekcije. Glavni dio je podijeljen u tri dijela. U prvom dijelu se govori o načinu zapisivanja ljudske detekcije, a u drugom o načinu zapisivanja računalne detekcije objekata. U posljednjem dijelu se govori o uspješnosti rezultata, kako su oni dobiveni, što znače i kako su prikazani.

1.1. Zadatak završnog rada

Zadatak ovog završnog rada je izraditi desktop aplikaciju za učitavanje video isječaka gdje će se moći testirati uspješnost računalne detekcije objekata u odnosu na ljudski vid. Potrebno je omogućiti zapisivanje ljudske percepcije specifičnih objekata za vrijeme trajanja video isječaka. Na primjeru usporediti i grafički prikazati odnos računalne detekcije specifičnog objekta i referentnih vrijednosti koje je zapisao čovjek.

2. RAD S VIDEO ISJEČCIMA

2.1. C# programski jezik i EMGU CV

C# je moderan objektno-orijentiran programski jezik koji programerima omogućuje programerima stvaranje različitih sigurnih i snažnih aplikacija koje se pokreću na .NET Frameworku. Može se koristiti za stvaranje Windows aplikacija, XML internetskih usluga, klijent-server aplikacije, aplikacije za upravljanje bazom podataka i još mnoge. Pruža napredno uređivanje koda, pogodne izglede korisničkog sučelja, ugrađeno pronalaženje pogrešaka i mnoge druge alate koji omogućuju lakše razvijanje aplikacija utemeljenima na C# programskom jeziku i .NET Frameworku. [1]

C# je izrazito izražajan jezik, ali vrlo jednostavan i lak za učenje. Programeri koji su upoznati s bilo kojim jezikom, C, C++ ili Java, mogu započeti rad s C# programskim jezikom u vrlo kratkom vremenu. C# sintaksa pojednostavljuje mnoge složenosti C++ i pruža snažne značajke kao što tipovi podataka koji mogu imati vrijednost nula, lambda izraze i izravno pristupanje memoriji što nije moguće u Javi. Kao objektno-orijentiran jezik, podržava koncept enkapsulacije, nasljeđivanja i polimorfizma. Sve varijable i metode, uključujući *Main* metodu, polaznu točku aplikacije, su enkapsulirane unutar definicije klase. Klasa može naslijediti izravno od jedne roditeljske klase, ali može implementirati više sučelja. C# proces izgradnje aplikacije je jednostavan u usporedbi s C i C++ te je fleksibilniji nego Java. Ne postoje odvojene zaglavne datoteke i ne zahtjeva se deklariranje metoda i tipova određenim redoslijedom te je moguće definirati neograničeni broj klasa, struktura, sučelja i događaja. [1]

Tab 2.1. *C# prostori imena korišteni u izradi projekta* [2]

System	Sadrži osnovne klase koje definiraju često korištene vrijednosti i referentne vrste podataka, događaje, sučelja…
System.Collections.Generic	Sadrži sučelja i klase koje definiraju generičke kolekcije
System.ComponentModel	Pruža klase koje se koriste za implementiranje ponašanja vremena izvršavanja i dizajniranja komponenti i kontrola
System.Data	Pruža pristup klasama koje predstavljaju ADO.NET arhitekturu
System.Drawing	Pruža pristup GDI+ osnovnoj grafičkoj funkcionalnosti
System.Linq	Pruža klase i sučelja koji podržavaju upite koji koriste LINQ (Lanuage-Integrated Query)

System.Text	Sadrži klase koje predstavljaju ASCII i Unicode kodiranje znakova
System.Threading.Tasks	Pruža tipove koji pojednostavljuju rad pisanja istovremenog i asinkronog koda
System.Windows.Forms	Sadrži klase za stvaranje aplikacija za Windows OS
System.Diagnostics	Sadrži tipove koji omogućuju odnos sa sustavnim procesima, događajima i brojačima izvođenja

Emgu CV je višeplatformski .Net omotač (engl. *cross platform wrapper*) u OpenCV[1] biblioteci za obradu slike. OpenCV funkcije mogu biti pozvane .NET kompatibilnim jezicima kao što su C#, VB, VC++, IronPython, itd. Omotač (engl. *the wrapper*) može prevesti Visual Studio, Xamarin Studio i Unity, a može biti pokrenut na Windowsu, Linuxu, Mac OS X-u, iOS-u, Androidu i Windows Phoneu. EmguCV je u potpunosti napisan u C#. Prednost je da može biti preveden u Monu što omogućuje pokretanje na bilo kojoj platformi koju podržava Mono[2], uključujući iOS, Android, Windows Phone, Mac OS X i Linux. [4]

Tab. 2.2. *EmguCV prostori imena korišteni u izradi projekta*

Emgu.CV	Omotač OpenCV funkcija za obradu slike [5]
Emgu.CV.GPU	Računalni vid koristi GPU [6]
Emgu.CV.UI	Korisničko sučelje (ImageBox) za prikazivanje objekta slike [7]
Emgu.Util	Skup korisnosti korišten u Emgu projektima [8]

2.2. .NET Framework

Microsoft je razvio C# programski jezik zajedno s .NET okruženjem (*engl. .NET Framework*), novom računalnom platformom koja pojednostavljuje razvoj aplikacije u razvojnom okruženju interneta. .NET okruženje je dizajnirano kako bi se ispunili sljedeći ciljevi:

- Za pružanje dosljednog objektno-orijentiranog programskog okruženja bilo da je objektni kod pohranjen i izvršen lokalno, izvršen lokalno, ali podijeljen na internetu ili izvršen udaljeno.

[1] OpenCV je biblioteka programskih funkcija koje je razvila tvrtka Intel. Koriste se za računalni vid u stvarnom vremenu. Napisano je u optimiziranom C i C++ programskom jeziku, a sadrže preko 500 funkcija koje pokrivaju različita područja računalne vizije. [3]

[2] Mono je projekt besplatnog i otvorenog koda (engl. *open-source*) predvođen Xamarinom, podružnicom Microsofta s odgovarajućim setom alata .NET okruženja između ostalog uključujući C# prevoditelj (engl. *compiler*) i CLR (Common Language Runtime).

- Za pružanje okruženja za izvršavanje koda koje smanjuje razvoj softvera i sukob verzija.
- Za pružanje okruženja za izvršavanje koda koje osigurava sigurno izvršavanje koda uključujući kod kojeg je stvorila nepoznata ili djelomično pouzdana treća strana.
- Za stvaranje iskustva programera dosljednog širokom rasponu različitih tipova aplikacija kao što su aplikacije za Windows operacijski sustav i internetske aplikacije.
- Za izgradnju svih komunikacija industrijskog standarda kako bi se osiguralo da .NET Framework kodovi mogu integrirati s bilo kojim drugim kodom.

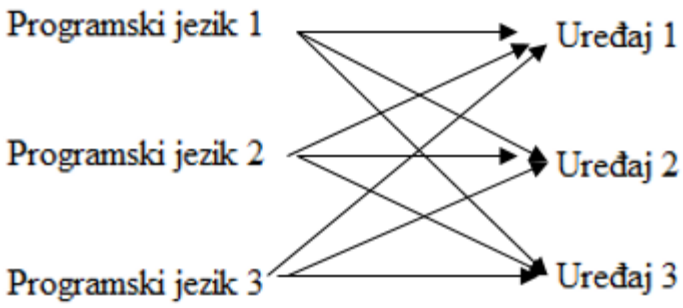
Svrha .NET okruženja ispunjava zahtjeve profesionalnog programera, a dva glavna dijela tog okruženja su CLR *(engl. Common Language Runtime)* i .NET Framework biblioteka klasa. [9]

2.2.1. CLR (Common Language Runtime)

CLR upravlja izvršenjem koda i pruža usluge koje to izvršenje čine lakšim. „Runtime“ znači da se kod pokreće što je drugo značenje izvršavanja. „Common Language“ znači da to vrijeme pokretanja upravlja izvršenjem koda koji je pisan u više različitih jezika koji dijele pružatelja usluga.

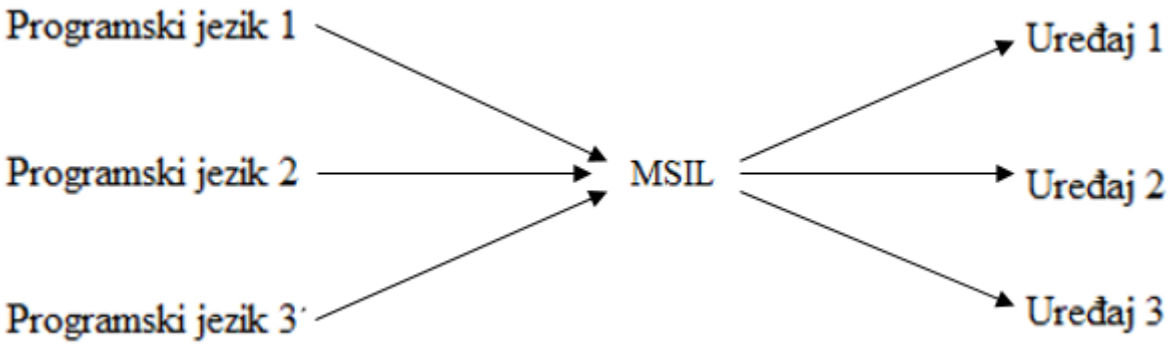
Microsoft je razvio C# kako bi iskoristio prednosti CLR-a. Njegove značajke rade posebno dobro s CLR-om. Popularni Visual Basic programski jezik se razvio u Visal Basic .NET koji je objektno-orijentiran jezik kako bi iskoristio prednosti CLR-a. Visual Basic programeri moraju učiti mnogo novih značajki kako bi iskoristi prednosti CLR-a koristeći Visual Basic .NET. C++, kao i njegov C prethodnik, ima mnogo mogućnosti koje ne odgovaraju novom pristupu. Verzija C++-a, zvana prilagođeni C++, prilagođava C++ kako bi mogao raditi s CLR-om da bi C++ programeri mogli integrirati kod s drugim CLR korisnicima.

Veliki problem s kojim se programeri susreću su veliki broj različitih tipova procesora koji pokreću kod. Windows, Macintosh i Unix strojevi koriste široku raznolikost hardvera kao i osobni digitalni asistenti (PDA), mobiteli, velika računala i druge platforme. Jedan način omogućavanja rada programa na svakom od tih uređaja je prevesti program u izvorni skup naredbi za taj uređaj. Koristeći ovaj pristup bilo bi potrebno onoliko prevoditelja koliki je umnožak korištenih jezika i uređaja.



Sl. 2.1. *Prevođenje tri jezika u izvorni kod za tri uređaja*

Drugi pristup, koristeći CLR, je pružanje posrednog jezika koji je sličan izvornim jezicima uređaja. Taj jezik se zove MSIL (*engl. Microsoft Intermediate Language).* Svaki jezik se prevede u MSIL. Tijekom vremena pokretanja CLR koristi JIT (*engl. Just In Time)* prevoditelja koji prevodi MSIL kod u izvorni kod za svaki korišteni uređaj. To zahtjeva jednog JIT prevoditelja za svaki uređaj za prijevod MSIL koda u izborni kod za taj uređaj. Taj proces prevođenja nije kompliciran kao prevođenje višeprogramskih jezika u izvorni kod jer je MSIL kod sličan izvornom kodu. [9]



Sl. 2.2. *Provođenje koristeći posredni jezik*

2.2.2. .NET Framework biblioteka klasa

.NET Framework biblioteka klasa pruža veliki i vrlo koristan skup tipova koji ubrzavaju razvojni proces. Biblioteka grupira tipove u prostore imena koji kombiniraju povezane tipove. Sadrži oko 100 prostora imena od kojih se koristi samo mali broj. Neki od prostora imena su:

- System – sadrži osnovne tipove
- System.Collections – definira različite kolekcije objekata
- System.Data – upravlja podatcima iz različitih izvora uključujući baze podataka

- System.Drawing – pruža grafike
- System.IO – omogućava čitanje i pisanje
- System.Net – pruža sučelje koje koriste računala za mrežnu komunikaciju
- System.Runtime.Remoting – podržava distribuirane aplikacije
- System.Text – upravlja kodiranjem znakova
- System.Threading – omogućuje višenitno programiranje
- System.Web – omogućuje komunikaciju preglednik-server
- System.Web.Services – omogućuje izgradnju i korištenje internetskih usluga
- System.Windows.Forms – za korisnička sučelja u aplikacijama za Windows OS
- System.Xml – pruža podršku za obradu XML-a [9]

2.3. Microsoft Visual Studio

Microsoft Visual Studio je integrirano razvojno okruženje *(engl. IDE – Integrated Development Environment)* Micrososfta. Koristi se za razvoj računalnih programa za Microsoft Windows kao i za internetske stranice, internetske aplikacije, internetske usluge i mobilne aplikacije. Visual Studio za razvoj softvera koristi Microsoftove platforme kao što su Windows API, Windows Forms, Windows Presentation Foundation, Windows Store i Microsoft Silverlight. Može proizvesti i izvorni i prilagođeni kod.

Visual Studio uključuje uređivač koda koji podržava komponente završavanja koda *(engl. IntelliSense)* kao i obnavljanja koda. Osim ugrađenog ispravljača pogrešaka, postoje i drugi ugrađeni alati kao što su dizajner formi za izradu GUI aplikacija, internetski dizajner, dizajner klasa i dizajner nacrta baza podataka. Prihvaća priključke koji povećavaju funkcionalnost gotovo svake razine uključujući dodavanje podrške za sustave kontrole izvora i dodavanje novih setova alata kao što su uređivači i dizajneri izgleda za jezike određenog područja ili setove alata za druga stajališta softverskog razvoja informacijskog sustava.

Podržava 36 različitih programskih jezika i omogućuje uređivaču koda i ispravljaču pogrešaka da podržavaju gotovo sve programske jezike. Ugrađeni jezici su C, C++ i C++/CLI (preko Visual C++-a), VB.NET (preko Visual Basic .NET), C#, F# i TypeScript. Podrška drugim jezicima kao što su Python, Ruby, Node.js i M i drugi je dostupna ukoliko se posebno instaliraju usluge jezika.

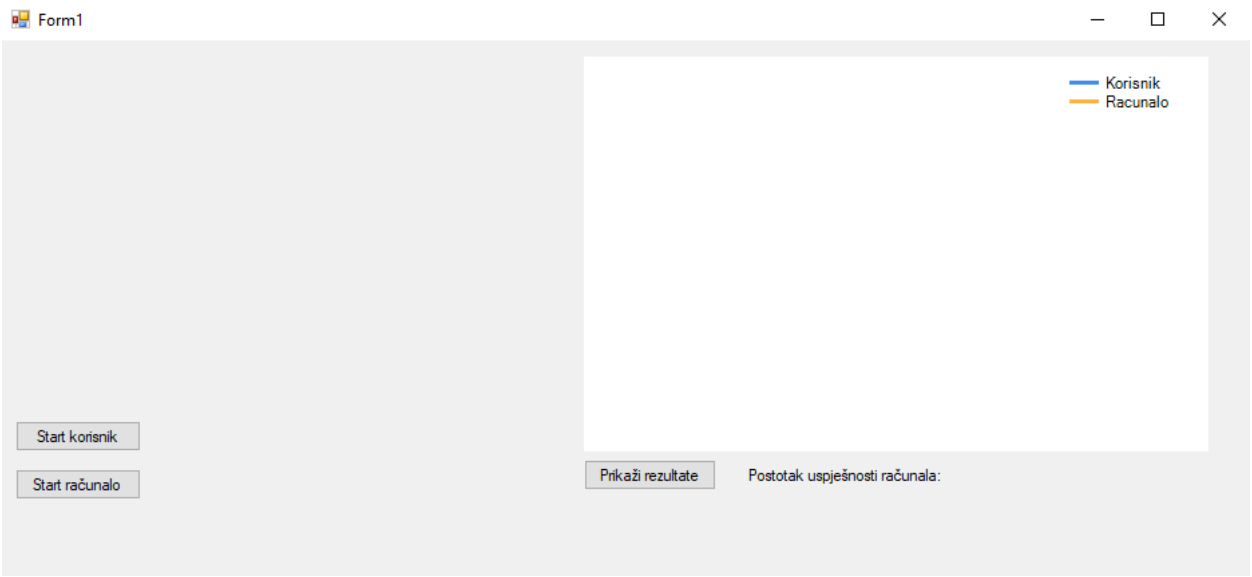
Microsoft pruža besplatnu verziju Visual Studia koji se zove Community edition koji podržava priključke i više različitih jezika. Prema funkcionalnosti, ova verzija je slična verziji Visual Studio Professional. [10]

2.3.1. Visual Studio 2013.

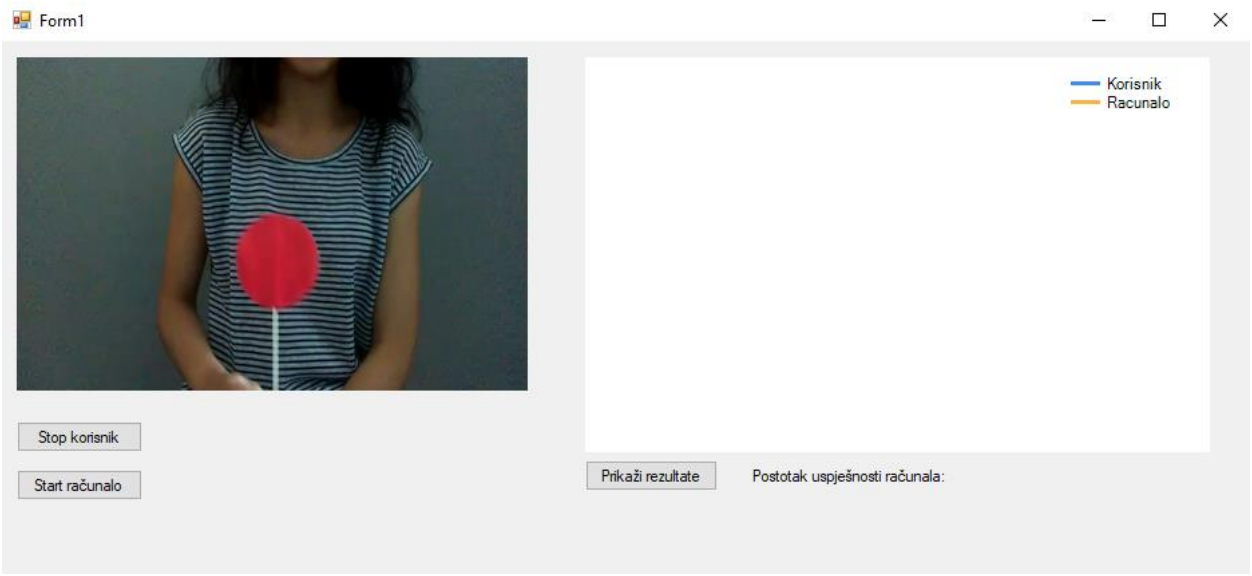
Projekt je izrađen u Visual Studiu 2013. čiji je pregled postao dostupan 26. lipnja 2013. godine, a programerima je postao dostupan na Microsoftovoj stranici 9. rujna 2013. godine. Završna verzija je bila dostupna 17. listopada iste godine zajedno s .NET 4.5.1. Visual Studio ima pet nadopuna, a zadnja se zove Visual Studio 2013.5 i bila je objavljena 20. lipnja 2015. godine.[10]

2.4. Učitavanje i prikaz videa

Pri pokretanju aplikacije i pritiskom na gumb „Start korisnik“ ili „Start računalo“ pokreće se video isječak koji se prikazuje u ImageBoxu kojeg smo postavili u dizajnu aplikacije.



Sl. 2.1. *Prikaz aplikacije prije učitavanja i pokretanja video isječka*



Sl. 2.2. *Prikaz aplikacije nakon učitavanja i pokretanja video isječka*

Kako bi se video isječak uspješno učitao i prikazao u ImageBoxu, potrebno je napisati odgovarajuće naredbe kao što je prikazano na slici 2.3.

```
private Capture _capture = null;
private Image<Bgr, Byte> colImage;
_capture = new Capture("RedDot.mp4");
_capture.QueryFrame();
colImage = _capture.RetrieveBgrFrame();
imageBox1.Image = colImage;
```

Sl. 2.3. *Prikaz koda za učitavanje i prikazivanje video isječka za ljudsku detekciju objekata*

- Capture(„RedDot.avi“) – omogućuje učitavanje videa iz datoteke.
- Image <Bgr, Byte> - omogućuje stvaranje prazne slike u koju ćemo spremati uhvaćenu sliku, odnosno okvir video isječka i ta slika će biti u boji.
- QueryFrame() – omogućuje hvatanje slika, odnosno okvira.
- RetrieveBgrFrame() – omogućuje vraćanje, odnosno spremanje uhvaćenih slika, tj. okvira u boji.
- imageBox1.Image – omogućuje prikazivanje video isječka u ImageBoxu, gdje je imageBox1 naziv ImageBoxa.

Ukoliko želimo učitavati video izravno s web kamere, pišemo Capture(0) za integriranu kameru ili Capture(1) za kameru spojenu USB kabelom, odnosno eksternu kameru.
Ukoliko želimo spremati crno-bijelu sliku, a ne u boji, pišemo: Image<Gray, Byte>.
Ukoliko želimo spremiti crno-bijeli okvir pišemo: RetrieveGrayFrame().

Kako bi čovjek mogao zabilježiti percepciju traženog objekta, video isječak se mora pokretati u stvarnom vremenu. Za omogućavanje toga korišten je mjerač vremena (engl. *Timer*) koji se nalazi u alatima i jednostavno se dodaje na dizajn aplikacije. U njemu se namješta interval, odnosno namješta se broj okvira koji će se prikazati u jednoj sekundi što omogućuje usporavanje ili ubrzavanje video isječka, po potrebi. Video isječak korišten u projektu prikazuje 24 okvira u sekundi što znači da interval mjerača vremena mora biti namješten na 1000 ms (milisekundi)/ 24 okvira, odnosno prikazat će se 24 okvira u jednoj sekundi.
Za prikaz video isječka za računalnu detekciju objekata, kod je nešto drugačiji, a dio je prikazan na slici 2.4.

```
_capture2 = new Capture("RedDot.mp4");
_capture2.ImageGrabbed += ProcessFrame;
```

Sl 2.4. *Prikaz koda za učitavanje i prikazivanje video isječka za računalnu detekciju objekata*

Kod prikazuje ponovno učitavanje video isječka za računalnu detekciju objekata nakon čega se hvataju okviri, odnosno slike koje se zatim jedna po jedna analiziraju u funkciji ProcessFrame. ImageGrabbed – poziva se kad je slika uhvaćena i koristi se kad se slike obrađuju jedna za drugom, bez obzira na brzinu videa.

3. TESTIRANJE USPJEŠNOSTI DETEKCIJE OBJEKATA U VIDEO ISJEČCIMA

Nakon pokrenutog video isječka koji se prikazuje u ImageBoxu, bilježi se detekcija traženog objekta. Bilježiti detekciju traženog objekta može korisnik, odnosno čovjek ili računalo. Ukoliko to obavlja korisnik, u ovom projektu, pritiskom bilo koje tipke na tipkovnici bilježi trenutak pojavljivana objekta kojeg traži u video isječku. Bilježiti se može još i pritiskom tipke na računalnom mišu ili neke određene tipke na tipkovnici. Ukoliko pojavljivanje određenog objekta bilježi računalo, obavlja se pomoću funkcije koja prepoznaje objekt koji se traži.

3.1. Zapis ljudske percepcije specifičnih objekata

U projektu se provjerava prepoznavanje i bilježenje crvene boje u pokrenutom video isječku. Za pokretanje video isječka korisnik koristi gumb (*engl. button*) na čiji pritisak se učitava i pokreće određeni video isječak koji se u kodu postavlja odgovarajućim naredbama.

```
_capture = new Capture("RedDot.mp4");
```

Sl. 3.1. *Linija koda za učitavanje videa za ljudsku percepciju*

Nakon što korisnik pokrene video isječak pritiskom gumba „Start korisnik“, naziv gumba se mijenja u „Stop korisnik“ sve dok ponovno ne bude pritisnut. Video isječak se pritiskom istog tog gumba može pauzirati, ukoliko korisnik želi, te isto tako i pokrenuti od tog trenutka gdje se video isječak zaustavio, odnosno pauzirao. Dok se video isječak prikazuje, korisnik prati prikazivanje crvene boje i to bilježi pritiskom bilo koje tipke na tipkovnici. Tijekom pritisnute tipke na tipkovnici smatra se da je crvena boja vidljiva, odnosno prikazana na ekranu, tj. u *ImageBoxu,* a kada nijedna tipka nije pritisnuta, odnosno kada korisnik prestane držati bilo koju tipku tipkovnice znači da u tom trenutku korisnik ne vidi crvenu boju. Svaki od tih događaja se bilježi u polje cjelobrojnog tipa podatka *(*engl. *integer)* ovisno o tome vidi li se ili ne crvena boja. Ukoliko je crvena boja vidljiva, sprema se „1“, odnosno istina, a ukoliko nije vidljiva, sprema se „0“, odnosno laž.

```
private int[] korisnik = new int[851];
```

Sl. 3.2. *Linija koda za deklariranje polja naziva „korisnik“*

Veličina polja se određuje prema duljini videa i broju okvira koji se prikazuju u jednoj sekundi. U projektu se koristi video duljine 37 sekundi s prikazivanje 23 okvira u jednoj sekundi i kada se to pomnoži dobije se broj 851, odnosno dobijemo ukupan broj okvira u video isječku koji nam određuje veličinu polja u koje spremamo podatke o pritisnutoj ili nepritisnutoj tipki tipkovnice.

```
private void txt_KeyDown(object sender, KeyEventArgs e)
{
     kor = 1;
}

private void txt_KeyUp(object sender, KeyEventArgs e)
{
     kor = 0;
}
```

Sl. 3.3. *Linije koda za reakciju pritisnute odnosno puštene tipke*

Varijabla „kor“ je varijabla u koju se sprema „1“ ili „0“ što označava pojavljivanje crvene boje te se ona pridružuje određenom elementu polja u ranije spomenutom mjeraču vremena koji omogućuje prolaženje okvir po okvir u učitanom videu, a to je prikazano na slici 3.4.

```
private void MyTimer_Tick(object sender, EventArgs e)
{

     korisnik[br] = kor;
     br++;
     if(br==850)
       {
           MyTimer.Stop();
       }

}
```

Sl. 3.4. *Linije koda koje prikazuju pridruživanje varijable „kor“ polju „korisnik“ te provjeru je li prikazan cijeli video isječak*

Varijabla „br“ je brojač koji broji okvire koji se provjeravaju u video isječku i kada dođe do kraja, odnosno do konačnog broja okvira, zaustavlja se brojač vremena, a time i završava pokrenuti video te korisnikovo bilježenje prepoznavanja crvene boje.
Popunjeno polje „korisnik“ se prikazuje u obliku grafa i daje nam tražene rezultate, odnosno prikaz trenutaka gdje je prepoznata crvena boja. Rezultati koje je stvorio korisnik se upotrebljavaju kao

referentni rezultati pomoću kojih se provjerava točnost prepoznavanja i bilježenja računalne detekcije crvene boje.

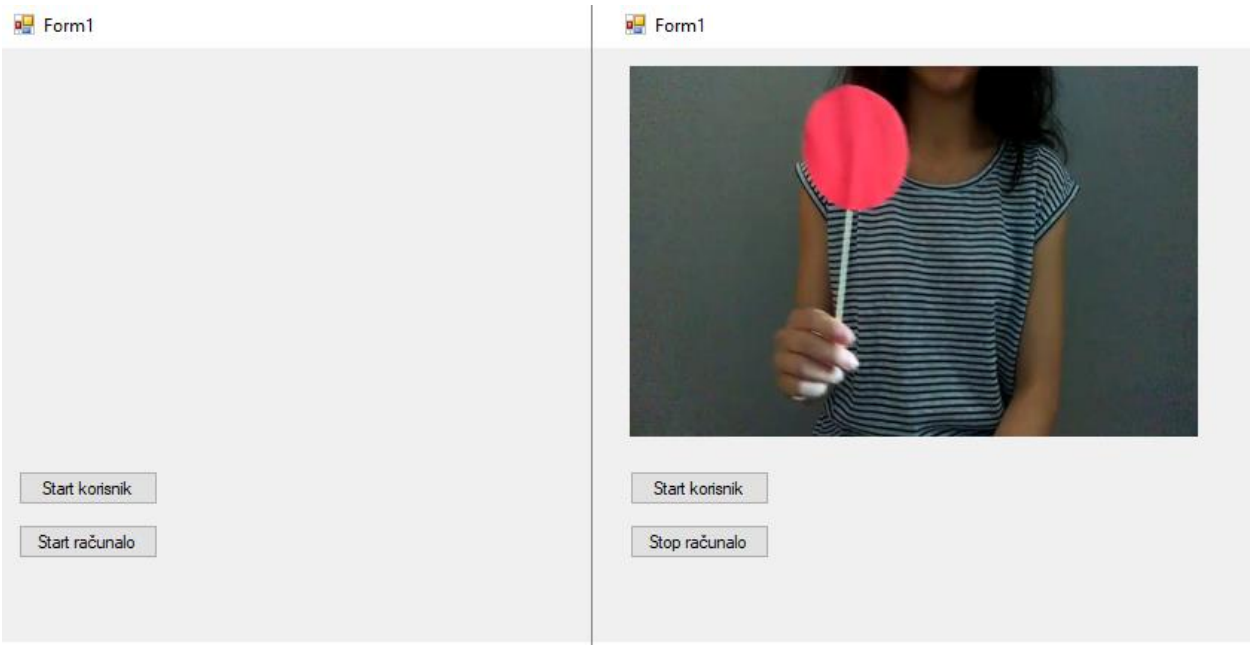
3.2. Zapis računalne detekcije specifičnih objekata

Za učitavanje i pokretanje video isječka za računalnu detekciju specifičnih objekata također se koristi gumb (engl. *button)* kojeg pritisne korisnik, a sve ostalo obavlja računalo. U projektu računalo također detektira crvenu boju čiji se rezultati na kraju uspoređuju s dobivenim rezultatom korisnika.

```
_capture2 = new Capture("RedDot.mp4");
```

Sl. 3.5. *Linija koda za učitavanje videa za računalnu detekciju*

Kada korisnik pokrene video isječak, kojeg računalo obrađuje, pritiskom gumba „Start računalo“, naziv gumba se promijeni u „Stop računalo“. Pokrenuti video isječak se može pauzirati ukoliko korisnik to želi i u tom slučaju se naziv gumba vraća na „Start računalo“ te ponovnim pokretanjem video se pokreće od tog trenutka gdje je zaustavljen.



Sl. 3.6. *Prije i poslije pritisnutog gumba*

Za obradu video isječka, odnosno za detekciju crvene boje, koristi se funkcija *ProcessFrame* u kojoj se obrađuje okvir po okvir i u svakom okviru se traži crvena boja te ukoliko se detektira, taj trenutak, odnosno okvir se bilježi sa „1“, odnosno istinom i to se sprema u novo polje naziva „racunalo“. Ukoliko se u određenom okviru ne detektira crvena boja, u polje se bilježi „0“, odnosno laž.

```
private int[] racunalo = new int[851];
```

Sl. 3.7. *Linija koda za deklariranje polja naziva „racunalo“*

```
private void ProcessFrame(object sender, EventArgs arg)
{
      imageBox1.Image = colImage2;
      brr++;

      if (brr == 851)
            _capture2.Stop();
}
```

Sl. 3.8. *Prikaz dijela funkcije ProcessFrame*

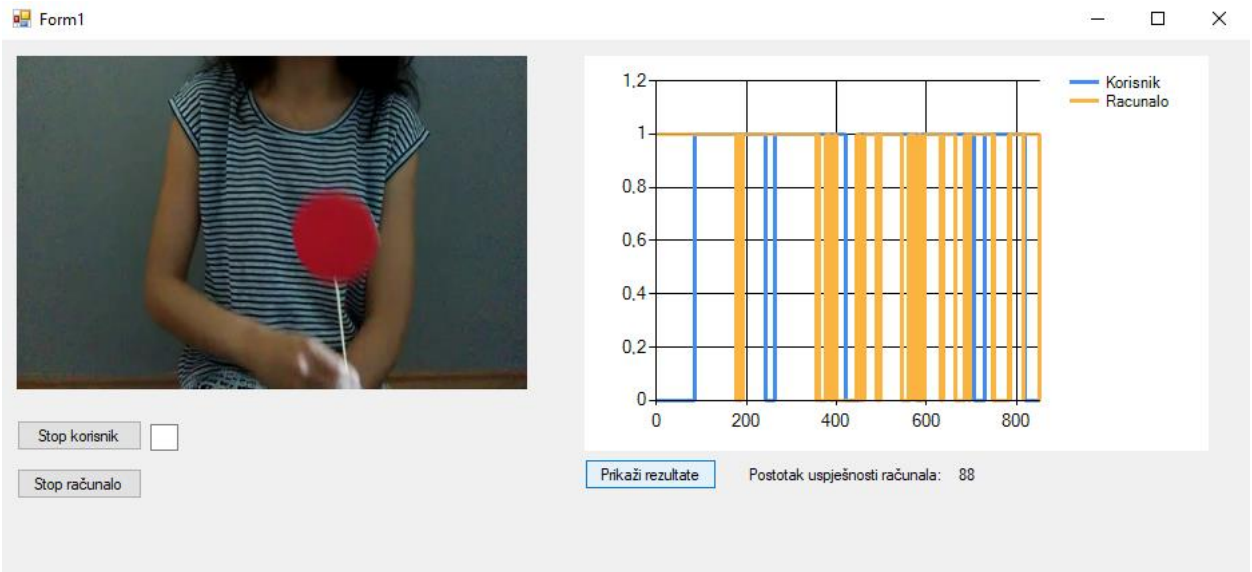
Slika 3.7. prikazuje učitavanje video isječka u *ImageBoxu* te varijablu „brr“ koja nam služi kao brojač koji provjerava je li prikazan cijeli video, odnosno je li cijelo polje popunjeno i ako je istina, zaustavlja se prikazivanje video isječka.
Funkcija *ProcessFrame* se poziva u *Main* funkciji nakon učitanog videa gdje svaki uhvaćeni okvir obrađuje.

```
_capture2.ImageGrabbed += ProcessFrame;
```

Sl. 3.9. *Pozivanje ProcessFrame funckije*

Iako se u projektu traži i detektira crvena boja, ta boja nije nužna, može se promijeniti te se može tražiti i zelena ili plava (RGB sustav boja) i može se tražiti različit raspon nijansi tih boja.
Popunjeno polje „racunalo“ se prikazuje u obliku grafa koji nam daje tražene rezultate, odnosno trenutke u kojima je računalo detektiralo crvenu boju. Ti rezultati se uspoređuju s referentnim rezultatima, odnosno rezultatima koje je stvorio korisnik i time dobivamo uspješnost i točnost računalne detekcije. Rezultati se uspoređuju tako da se prati svaki trenutak video isječka te se zabilježava svaki trenutak u kojem se graf s rezultatima korisnika poklapa s grafom s rezultatima

računala. Ukupan broj tih trenutaka dijeli se s ukupnim brojem okvira video isječka i time se dobije postotak uspješnosti računalne detekcije koji se ispisuje na oznaku (engl. *label*).



Sl. 3.10. *Prikaz postotka uspješnosti računalne detekcije*

4. EKSPERIMENTALNI REZULTATI

U projektu se izvršava ljudska percepcija i računalna detekcija crvene boje u video isječku gdje korisnik i računalo bilježe trenutke pojavljivanja te boje. Trenutci pojavljivanja se spremaju u polja cjelobrojnog tipa i iz toga se dobiju traženi rezultati iz kojih se može zaključiti koliko je uspješna računalna detekcija u odnosu na referentne vrijednosti koje je zapisao čovjek.

4.1. Jednostavan algoritam za detekciju objekata crvene boje

Računalo u projektu detektira crvenu boju pomoću jednostavnog algoritma za detekciju objekata crvene boje. Definirana je funkcija „ProcessFrame“ koja provjerava okvir po okvir video isječka, odnosno u svakom od okvira traži crvenu boju. Svaki element polja „racunalo“ se prvo postavlja u nulu, tj. gleda se kao da u tom elementu polja crvena boja nije pronađena. Svaki od elemenata polja predstavlja jedan okvir video isječka. Nakon što je određenom elementu pridodana nula, provjerava se okvir i traže se određene nijanse crvene boje. U slučaju pronalaska crvene boje, vrijednost elementa polja se mijenja iz nula u jedan.

```
for (int i = 0; i < colImage2.Height; i++)
{
    for (int j = 0; j < colImage2.Width; j++)
    {
         if (colImage2[i, j].Red > 245)
         {
              racunalo[brr] = 1;
         }
     }
}
```

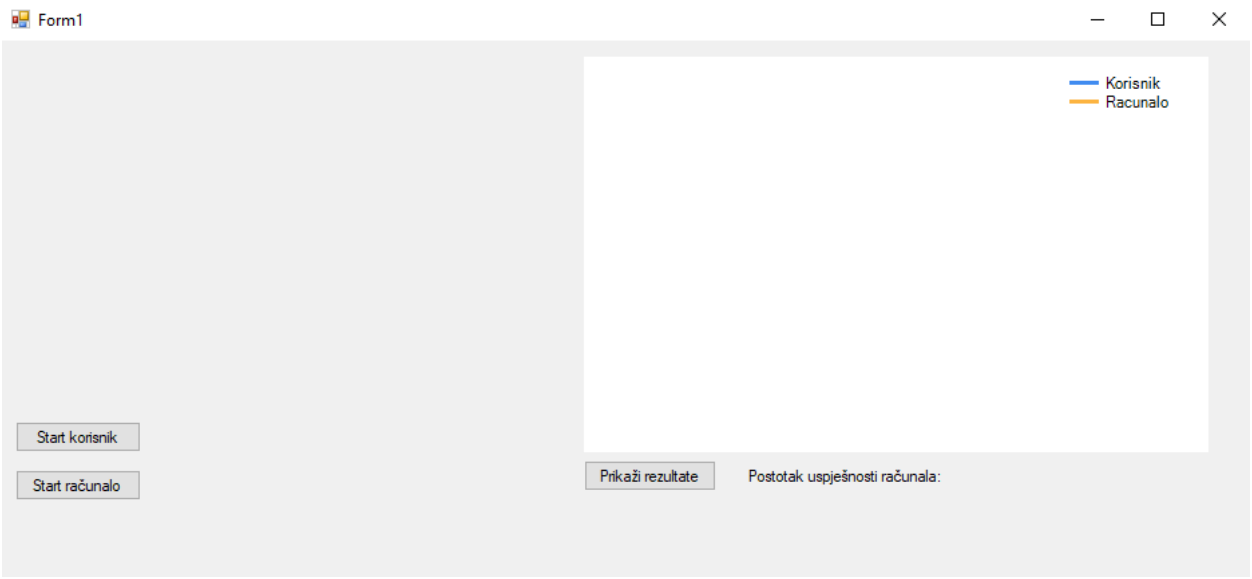
Sl. 4.1 *Dio funkcije ProcessFrame za traženje crvene boje*

Funkcija pretražuje okvir po visini i širini, odnosno pretražuje svaki piksel okvira. Pretraživati se ne mora nužno crvena boja, može se pretraživati i plava ili zelena po RGB sustavu boja, a u ovoj funkciji *.Red* označuje da se pretražuje crvena boja. Dio koji slijedi iza *.Red, >245,* označuje da se pretražuju sve nijanse crvene boje veće od 245, znači da pretražuje raspon nijansi od 245 do 255. Iz toga možemo zaključiti da se pretražuju jače nijanse crvene boje.

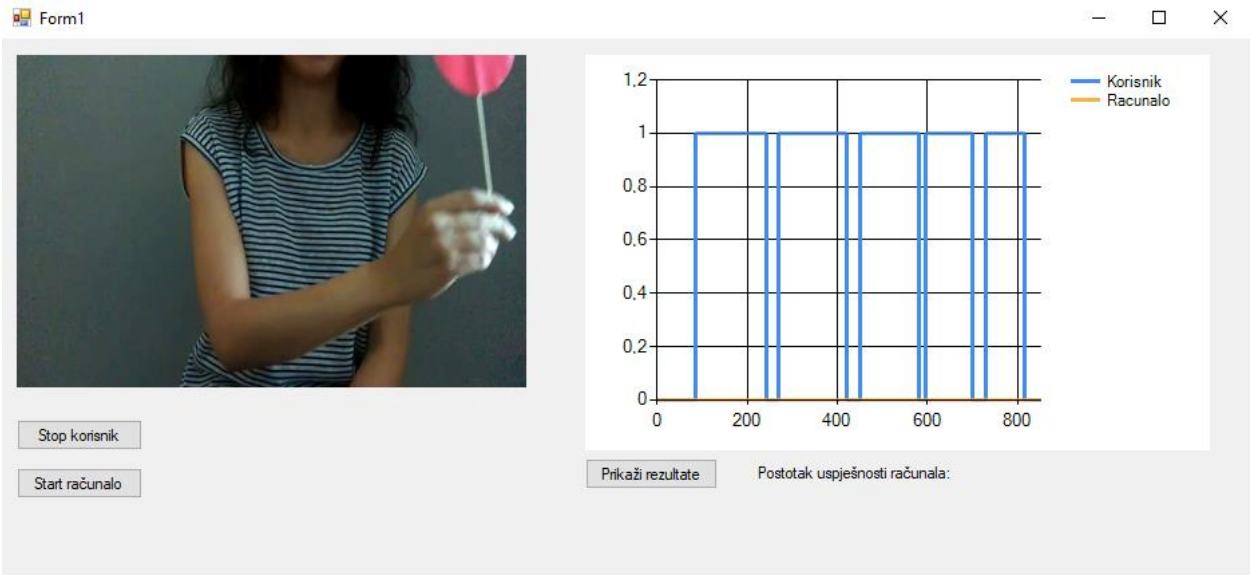
4.2. Usporedba čovjeka i računala u detekciji

Korisnikovo prepoznavanje i bilježenje crvene boje obavlja se pritiskom bilo koje tipke na tipkovnici. Pritisak bilo koje tipke tipkovnice može se ograničiti, odnosno može se definirati točno određena tipka na tipkovnici ili tipka računalnog miša kojom će se bilježiti prepoznavanje određenog objekta. Sve dok je tipka pritisnuta, u definirano polje „korisnik“ sprema se vrijednost „1“ što predstavlja trenutak u kojem se pojavljuje crvena boja. Kada tipka nije pritisnuta, u polje

se sprema vrijednost „0“ što predstavlja trenutak u kojem se crvena boja ne pojavljuje. Veličina polja se određuje umnoškom dužine video osječka, odnosno brojem sekundi i brojem okvira po jednoj sekundi istog video isječka. Nakon što je cijeli video prikazan, odnosno cijelo polje „korisnik“ popunjeno, mogu se dobiti traženi rezultati. Rezultati se prikazuju pritiskom gumba „Ispis rezultata“ čime se dobije graf.



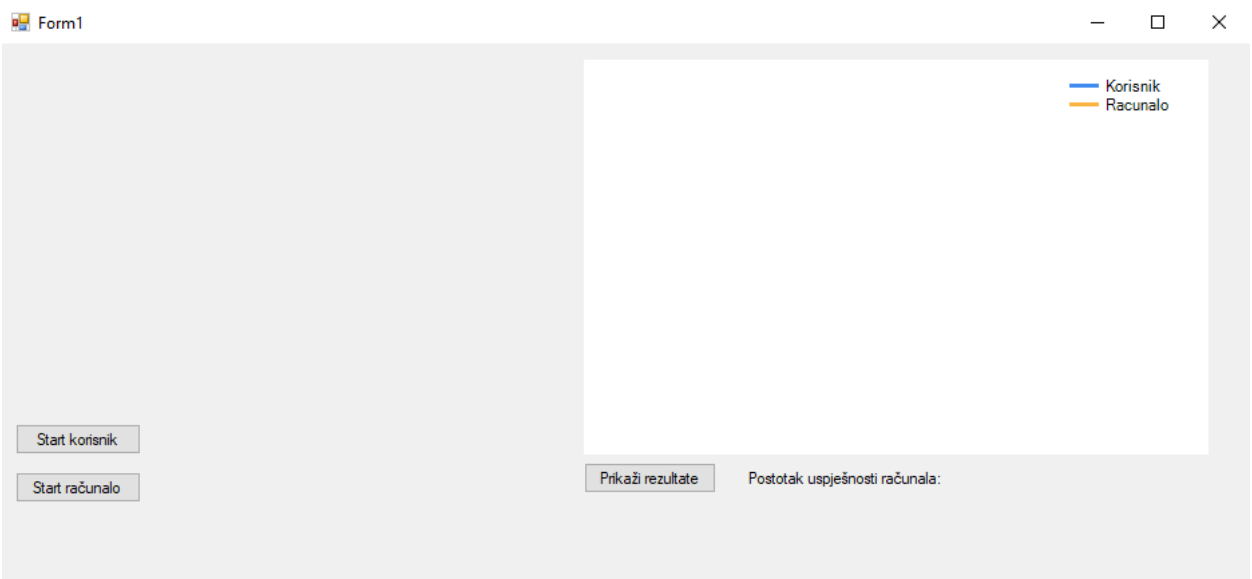
Sl. 4.2. *Izgled prije ispisa rezultata*



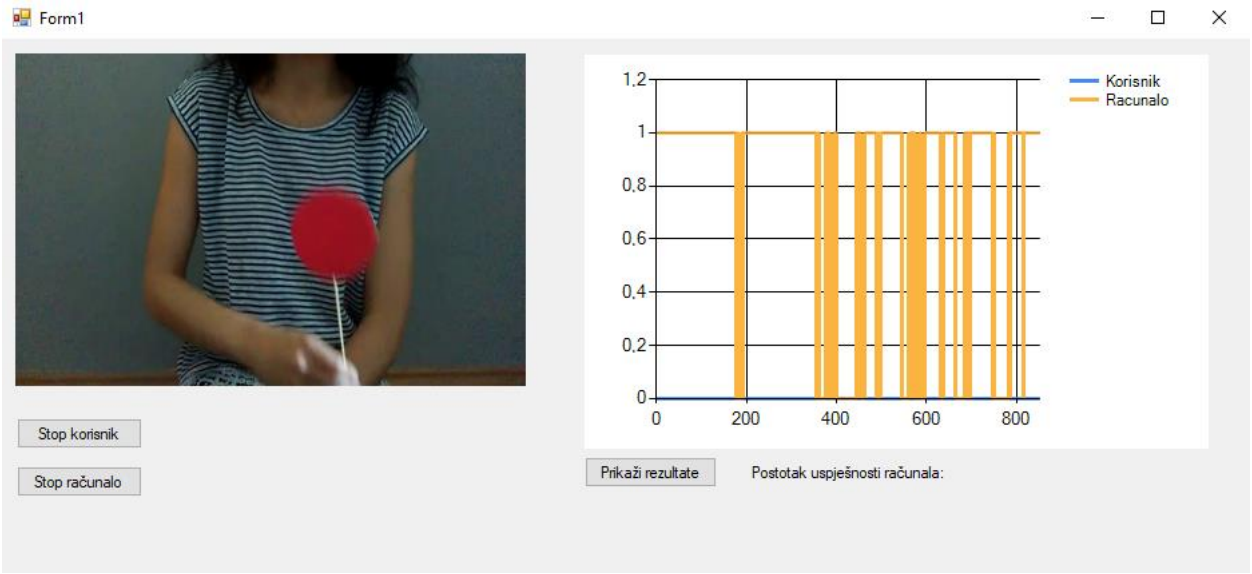
Sl. 4.3. *Izgled nakon ispisa rezultata*

Korišteni graf je step-graf koji ide od nula do jedan gdje jedan predstavlja trenutak pojavljivanja crvene boje, a nula trenutak gdje se crvena boja ne pojavljuje. S desne strane grafa je vidljiva legenda koja pokazuje da plava lijina u grafu prikazuje rezultate koje je zapisao korisnik i ti rezultati se uzimaju kao referentni rezultati.

Računalno prepoznavanje crvene boje obavlja se pomoću ranije spomenute funkcije „ProcessFrame“ koja provjerava svaki okvir video isječka i traži crvenu boju. Funkcija popunjava polje „racunalo“ jedinicama i nulama. Ukoliko je u element polja upisana nula, znači da u tom trenutku, odnosno okviru nije pronađena crvena boja, a ukoliko je jedan, znači da je crvena boja pronađena. Veličina polja „racunalo“ se određuje na isti način kao i veličina polja „korisnik“. Funkcija „ProcessFrame“ koristi brojač koji provjerava jesu li prikazani svi okviri video isječka i ukoliko jesu, zaustavlja se video i svi elementi polja „racunalo“ su tada popunjeni. Rezultati se dobiju na isti način kao i rezultati koje je zapisao korisnik, pritiskom gumba „Ispis rezultata“, čime se dobije graf.

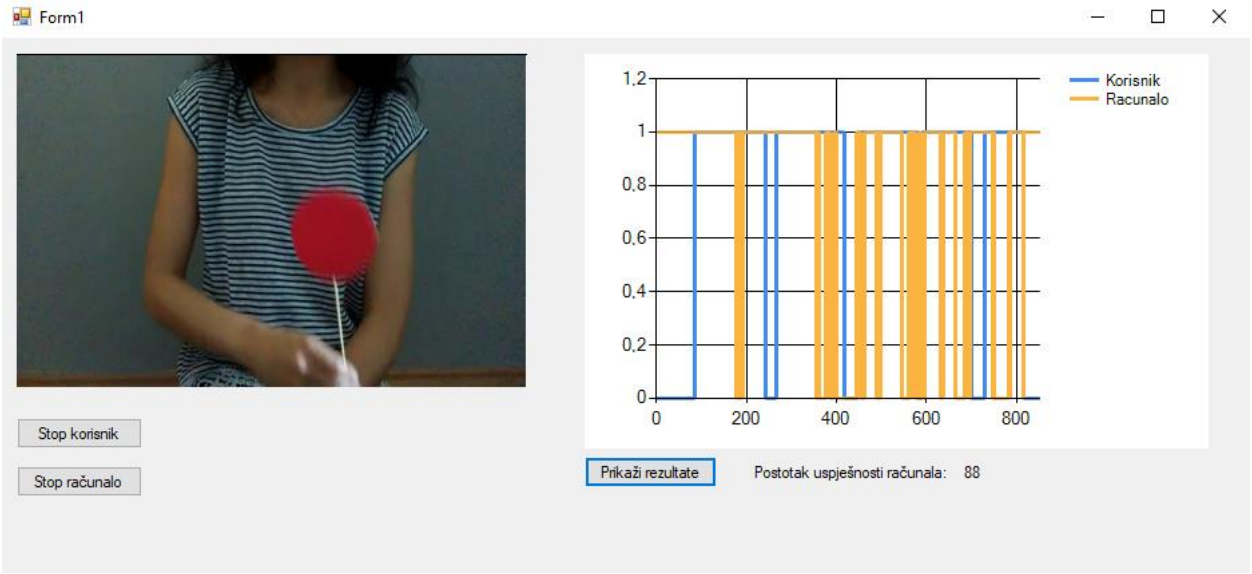


Sl. 4.4. *Izgled prije ispisa rezultata*



Sl. 4.5. *Izgled nakon ispisa rezultata*

S desne strane grafa je vidljiva legenda koja pokazuje da žuta lijina u grafu prikazuje rezultate koje je zapisalo računalo i te rezultate uspoređujemo s referentnim rezultatima koje je zapisao korisnik.



Sl. 4.6. *Referentni rezultati i rezultati računala*

Iz dobivenih rezultata se vidi učinkovitost računala i njegova točnost u detekciji crvene boje s obzirom na referentne rezultate koje je zapisao korisnik.

Kako bi se spoznalo još jedno svojstvo računalne detekcije, u projekt je dodana štoperica (engl. *Stopwatch)* koja bilježi koliko je vremena potrebno da računalo prikaže cijeli video isječak i zabilježi trenutke pojavljivanja crvene boje. Budući da funkcija *ProcessFrame* omogućuje računalnu detekciju crvene boje, kod za štopericu se dodaje na početku, odnosno na kraju te funkcije, a on je vidljiv na slici 4.8.

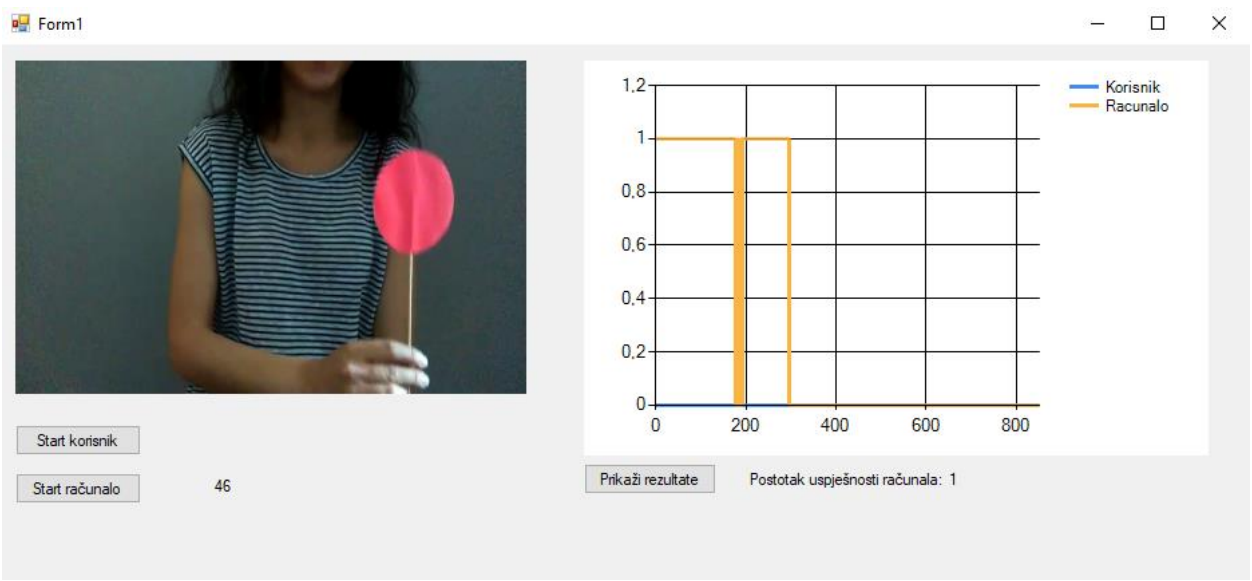
```
private Stopwatch time = new Stopwatch();
```

Sl. 4.7. *Deklaracija štoperice*

```
time.Start();
//Preskoceni dio koda
time.Stop();
```

Sl. 4.8. *Prikaz koda za štopericu*

Rezultat štoperice je vrijeme u sekundama, a ispisuje se na oznaku (engl. *Label*) na svaki pritisak gumba „Stop računalo“, odnosno svaki put kad zaustavimo prikazivanje videa, prikaže se vrijeme koje je proteklo do tog trenutka te ponovnim pokretanjem video isječka, vrijeme se nastavlja od trenutka zaustavljanja i na kraju možemo dobiti ukupno vrijeme potrebno za prikazivanje videa i pretraživanje crvene boje.



Sl. 4.9. *Prikaz trenutka u kojem je ispisano proteklo vrijeme zabilježeno štopericom*

5. ZAKLJUČAK

Ovaj završni rad, čija je zadaća bila napraviti desktop aplikaciju za prepoznavanje objekata u video isječcima, rezultirao je prepoznavanjem crvene boje. Pojavljivanje crvene boje bilježe korisnik ili računalo. Rezultati zapisivanja korisnika uzimaju se kao referentni rezultati s kojima se uspoređuju rezultati koje stvori računalo i iz toga se može zaključiti koliko točno i precizno računalo detektira crvenu boju. Aplikacija je izrađena u C# programskom jeziku i EmguCV-u koji omogućuje obradu slike i videa što je olakšalo izradu aplikacije zbog provjere svakog okvira.

Osim crvene boje, koja je u ovom projektu uzeta za primjer, mogu se pretraživati i zelena ili plava, odnosno ovakvo okruženje i ovakav pristup može se koristiti za analizu detekcije bilo kojih objekata (korištenjem drugih algoritama) te dodavanjem toga, aplikacija se može poboljšati. Ukratko, sam sustav analize i zapis čovjekove i računalne odluke univerzaalan je za mnoge slične probleme detekcije.

LITERATURA

[1] BillWagner, „Introduction to the C# Language and the .NET Framework“. [Na internetu]. Dostupno na: https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework. [Pristupljeno: 11-lip-2017].

[2] „Microsoft API and Reference Catalog“. [Na internetu]. Dostupno na: https://msdn.microsoft.com/library. [Pristupljeno: 25-kol-2017].

[3] P. Joshi, D. M. Escriva, i V. Godoy, *OpenCV By Example*. Packt Publishing Ltd, 2016.

[4] „Emgu CV: OpenCV in .NET (C#, VB, C++ and more)“. [Na internetu]. Dostupno na: http://www.emgu.com/wiki/index.php/Main_Page. [Pristupljeno: 12-lip-2017].

[5] „Emgu.CV Namespace“. [Na internetu]. Dostupno na: http://www.emgu.com/wiki/files/1.4.0.0/html/b72c032d-59ae-c36f-5e00-12f8d621dfb8.htm. [Pristupljeno: 24-lip-2017].

[6] „Emgu.CV.GPU Namespace“. [Na internetu]. Dostupno na: http://www.emgu.com/wiki/files/2.3.0/document/html/0029e0f2-fc71-3cfc-2fa6-c5aa6f94f117.htm. [Pristupljeno: 24-lip-2017].

[7] „Emgu.CV.UI Namespace“. [Na internetu]. Dostupno na: http://www.emgu.com/wiki/files/1.3.0.0/html/ec0d4663-4383-06f5-f0a4-deb44e42fafb.htm. [Pristupljeno: 24-lip-2017].

[8] „Emgu.Util Namespace“. [Na internetu]. Dostupno na: http://www.emgu.com/wiki/files/1.4.0.0/html/bd835675-8ca9-6b2c-03cc-8afcdd7a3bae.htm. [Pristupljeno: 24-lip-2017].

[9] A. Gittleman, *Computing with C# and the .NET Framework*. Jones & Bartlett Publishers, 2011.

[10] R. B. Dunaway, *The Book of Visual Studio .NET: A Guide for Developers*. No Starch Press, 2002.

SAŽETAK

Cilj ovog završnog rada bilo je izraditi desktop aplikaciju za učitavanje video isječaka gdje će se moći testirati uspješnost računalne detekcije u odnosu na ljudski vid. Potrebno je omogućiti zapisivanje ljudske detekcije objekata za vrijeme trajanja video isječaka te na primjeru usporediti i grafički prikazati odnos računalne detekcije specifičnog objekta i referentnih vrijednosti koje je zapisao čovjek. Završni rad je rezultirao prepoznavanjem crvene boje gdje se rezultati zapisivanja korisnika uzimaju kao referentni rezultati s kojima se uspoređuju rezultati koje stvori računalo i iz toga se može zaključiti koliko točno i precizno računalo detektira crvenu boju.

Ključne riječi: C#, EmguCV, detekcija, .NET Framework

ABSTRACT

Framework for testing the detection performance of objects in video clips.

The aim of this final year project was to create desktop application for uploading video clips, where it will be possible to test the performance of computer detection in relation to human vision. It was necessary to allow the recording of human detection of objects during the duration of the video. Furthermore, it was compared and graphically illustrated in the example the relationship between the computer detection of some specific object and the human detection which is taken as the reference result. Final year project has resulted in red color recognition, where user's results are taken as reference results which are compared to computer's results and from that can be inferred how accurate and precise the computer detects red color.

Keywords: C#, EmguCV, detection, .NET Framework

ŽIVOTOPIS

Nikolina Češić, rođena 21.6.1995. godine u Novoj Gradiški. Završila je osnovnu školu Ante Starčevića u Rešetarima te Gimnaziju u Novoj Gradiški koju završava 2014. godine, a iste godine upisuje Elektrotehnički fakultet, smjer računarstvo. Tijekom školovanja je nekoliko puta sudjelovala na školskom, a nekoliko puta i na županijskom natjecanju iz informatike.