**Elektronski fakultet, Niš**

**Niš, 2022.**

**II Projekat**

Veštačka Inteligencija – Prepoznavanje boja i oblika koristeći OpenCV biblioteku

Petar Petković, Nikola Krstić



**Profesor: Leonid Stoimenov**

Sadržaj

[Uvod – OpenCV 3](#_Toc94027171)

[Primena OpenCV-a 4](#_Toc94027172)

[Potencijalni OpenCV projekti 5](#_Toc94027173)

[Ideja projekta 6](#_Toc94027174)

[Podešavanje projekta 7](#_Toc94027175)

[Realizacija projekta 8](#_Toc94027176)

[Postupak prepoznavanja oblika i boja 10](#_Toc94027177)

[Zaključak 11](#_Toc94027178)

[Reference 12](#_Toc94027179)

# Uvod – OpenCV

OpenCV ( eng. Open Source Computer Vision) je besplatna biblioteka funkcija za računarsku viziju. Razvijena je u Intelovom istraživačkom centru u Nižnjem Novgorodu u Rusiji i podržana od strane Itseez-a. Iako je ovaj projekat zvanično pokrenut, sada već davne 1999. godine, prvo stabilno izdanje verzije 4.1.1 je izdano 26. jula 2019. godine.   
  
Napisana je u C/C++ i prvobitna namena je bila istraživanje za unapređivanje CPU-intensive aplikacija, aplikacija koje koriste veliki deo procesora, deo serije projekata, uključujući i praćenja zraka u realnom vremenu i 3D prikaza zida. Glavni doprinosioci projekta su takođe bili i eksperti za optimizaciju iz “Intel Russia”, kao i tim stručnjaka iz “Intel's Perfomance Library Team”.[1]

# Primena OpenCV-a

U početku OpenCV-a, ciljevi projekta su bili :

* Advance vision pretraga pružajući ne samo open source kod, nego i optimizovan kod za glavni vid infrastrukture. Fokus nije bačen na ponovni razvoj koda.
* Prošireni vid znanja time što se koristi uobičajena infrastruktura na kojoj je razvojni tim mogao da nadograđuje tako da je kod mogao biti lakši za čitanje i transfer.
* Naprednije komercijalne aplikacije, praveći prenosiv, perfomansno optimizovan kod, koji je bio besplatan, sa licencom koja nije bila open source.

Danas, primena je mnogo veća, postaje deo svakodnevice a primeri za to su:[2]

* Praćenje objekata ( Slika 1)

A picture containing text, scene, road, way

Description automatically generated

*Slika 1 – Praćenje objekata[3]*

* Face recognition
* Night vision
* Opisivanje slika
* Prepoznavanje informacija sa slika
* Prevod slike u text

# Potencijalni OpenCV projekti

Biblioteka ima više od 2500 optimizovanih algoritama, što uključuje sveobuhvatan skup klasičnih i najsavremenijih algoritama kompjuterskog vida i mašinskog učenja. Kako je open source, broj projekta koji se mogu implementirati putem OpenCV biblioteke je ogroman, i granica je samo naša mašta. Neki od potencijalnih projekata koji se mogu implementirati putem OpenCV biblioteke su:

* Merenje dimenzija objekata putem kamere
* Dokument skener
* Praćenje broja automobila koji pređu zamišljenu liniju
* Merenje brzine vozila
* Prepoznavanje lika
* Primena filtera na prepoznatom liku
* Real-time prevod teksta
* Prepoznavanje objekata na slici

# Ideja projekta

U prethodnom poglavlju nabrojali smo nekoliko ideja za projekte koji se mogu razvijati pomoću OpenCV tehnologije. Jedna od najpopularnijih ideja, a u isto vreme pogodna za učenje OpenCV tehnologije je prepoznavanje objekata na slici.

Kako najduža putovanja kreću jednim korakom, smatramo da naše upoznavanje sa ovom tehnologijom treba biti jedan mali korak, jedan mali projekat koji će prethoditi nizu velikih i kompleksnih projekata. Za početak, uz pomoć NumPy, imutils i pyplot bilioteka i programskog jezika Python, potrudićemo se da prepoznamo geometrijske oblike i njihove boje na slici(Slika 2).

Shape

Description automatically generated

*Slika 2 – Različiti geometrijski oblici u različitim bojama[4]*

# Podešavanje projekta

Osim podrazumevano instaliranog Python-a, neophodno je dodatno instalirati prethodno navedene biblioteke preko terminala, kucanjem sledećih komandi(Slike 3,4,5,6).

Text

Description automatically generated

Graphical user interface, text, email

Description automatically generated

A blue screen with white text

Description automatically generated with medium confidence

A blue screen with white text

Description automatically generated with medium confidence

*Slike 3,4,5,6 - Komande za instalaciju biblioteka [5]*

Nakon instalacije, potrebno je importovati ih(Slika 7).

Logo

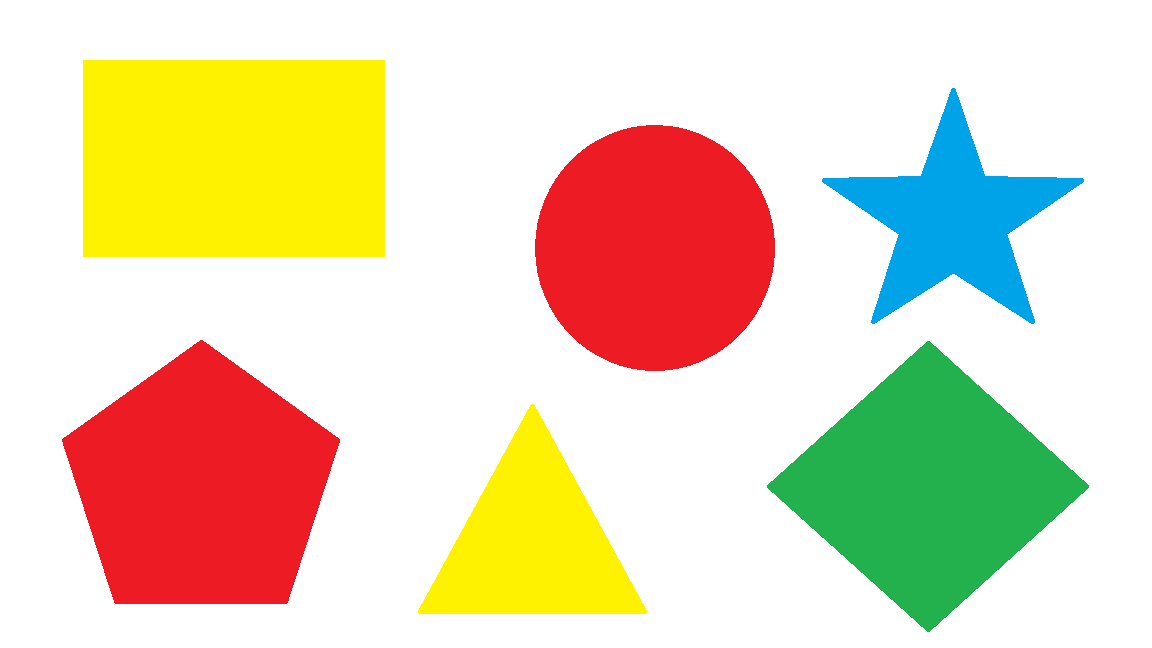
Description automatically generated

*Slika 7 – Import biblioteka*

# Realizacija projekta

Krajnji cilj nam je sledeći:

* Na ulazu programa je slika sa različitim geometrijskim figurama (slika 8).



*Slika 8 – Geometrijske figure obojene u žutoj, plavoj, crvenoj i zelenoj boji*

* Program prepoznaje oblik na osnovu broja uglova kao i jednu od 4 unapred definisane boje(crvena,žuta,plava i zelena).
* *Shape, polygon

  Description automatically generated*Program ispisuje na prepoznatim figurama oblik i boju(Slika 9)

*Slika 9 – Izlaz programa*

## Postupak prepoznavanja oblika i boja

Na početku programa, neophodno je učitati sliku na kojoj će se prepoznavati geometrijske figure i boje. Zatim se ta slika prevodi u HSV sistem. Na osnovu zadatih parametara putem numpy array-a koji definiše opseg boja se određuje pripadnost boje datom opsegu. U vezi sa tim, neophodno je napraviti maske koje sadrže donju i gornju granicu opsega. Sledeći korak je prepoznavanje boja i vraćanje liste figura na osnovu boja. Iteriranjem kroz te liste se ujedno i vrši brojanje temena svake od figura, pa osim prepoznavanje boja imamo i prepoznavanje oblika. Za dobijen broj uglova svake figure se nalazi mapiranje u dictionary-ju, pa se za ključ koji predstavlja broj temena dobije odgovarajuće ime mnogougla. Na kraju se prikazuje novonastala slika sa odgovarajućim atributima(Slika 10).

Text

Description automatically generated

*Slika 10. Kod programa*

# Zaključak

Ako uzmemo u obzir da nam je ovo prvi put da se susrećemo sa ovim načinom programiranja, mislimo da možemo biti i više nego zadovoljni postignutim. Osim krajnjeg rešenja koje je ispravno i radi ono što smo zamislili, jako nam je bilo bitno da razumemo koncepte, funkcije i način razmišljanja kada su u pitanju projekti ovog tipa. Kada smo počinjali, ideja nam je bila svakako da iskoristimo već postojeće implementacije osnovnih funkcija u OpenCV-u, ali i da se rešenje sastoji iz više osnovnih stvari (1. Prepoznavanje boje, 2. Prepoznavanje oblika) i da kombinovanjem i optimizacijom razumemo svaku funkciju pojedinačno. Ovo nije sigurno naš poslednji susret sa ovom tehnologijom, već u sledećem semestru nas čeka izborni predmet “Računarska vizija” i sigurni smo da će nam ovo istraživanje biti odlična osnova za ono što nam predstoji.

# Reference

1. https://sr.wikipedia.org/sr-ec/OpenCV
2. <http://ni.mg.edu.rs/static/resources/v5.0/sre1_opencv_mg_sj.pdf>
3. <https://learnopencv.com/multitracker-multiple-object-tracking-using-opencv-c-python/>
4. <https://www.smartick.com/blog/math/geometry/geometry-2d-shapes-tangram/>
5. <https://pypi.org/project>