

# Vizijska aplikacija za procjenu spola i godina

---

**Cvenk, Vilim**

**Undergraduate thesis / Završni rad**

**2023**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

*Permanent link / Trajna poveznica:* <https://um.nsk.hr/um:nbn:hr:235:968211>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-07-06**

*Repository / Repozitorij:*

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

**Vilim Cvenk**

Zagreb, 2023.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

Mentor:

Prof. dr. sc. Tomislav Stipančić, dipl. ing.

Student:

Vilim Cvenk

Zagreb, 2023.

Izjavljujem da sam ovaj rad izradio samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Posebna zahvala prof. dr. sc. Tomislavu Stipančiću zbog podrške koju je pružio te za pomoć i korisne savjete prilikom izrade ovoga rada.

Vilim Cvenk



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomske ispite  
Povjerenstvo za završne i diplomske ispite studija strojarstva za smjerove:  
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo  
materijala i mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: 602 – 04 / 23 – 6 / 1	
Ur.broj: 15 - 1703 - 23 -	

## ZAVRŠNI ZADATAK

Student: **Vilim Cvenk** JMBAG: **0035222903**

Naslov rada na hrvatskom jeziku: **Vizijska aplikacija za procjenu spola i raspona godina**

Naslov rada na engleskom jeziku: **A vision application for gender and age range estimation**

Opis zadatka:

Računalna analiza i prepoznavanje omogućuju računalu da prepozna značajke na slikama ili video materijalu te ih koristi kod različitih primjena u sklopu stvarne okoline. U tu svrhu se često koriste vizijski sustavi pomoću kojih se vrši akvizicija i učenje informacija o specifičnim točkama ili značajkama od interesa na licu osobe. Te se informacije potom kombiniraju na različite načine kako bi se ostvarila željena primjena.

U radu je potrebno izraditi cjelovito softversko rješenje za procjenu godina i spola osobe na slici ili video materijalu temeljem karakterističnih značajki na licu, prateći sljedeće korake:

- identificirati i proučiti metode računalnog vida te strojnog učenja koje su prikladne za rješavanje ovog zadatka,
- implementirati metodu za klasifikaciju ljudi temeljem spola,
- implementirati metodu za klasifikaciju ljudi s obzirom na različite raspone godina,
- ostvariti prikaz rezultata na slici ili video materijalu u realnom vremenu.

Razvijenu aplikaciju je potrebno testirati s ljudskim subjektima u sklopu Laboratorija za projektiranje izradbenih i montažnih sustava.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:

30. 11. 2022.

Zadatak zadao:

Doc. dr. sc. Tomislav Stipančić

Datum predaje rada:

1. rok: 20. 2. 2023.  
2. rok (izvanredni): 10. 7. 2023.  
3. rok: 18. 9. 2023.

Predviđeni datumi obrane:

1. rok: 27. 2. – 3. 3. 2023.  
2. rok (izvanredni): 14. 7. 2023.  
3. rok: 25. 9. – 29. 9. 2023.

Predsjednik Povjerenstva:

Prof. dr. sc. Branko Bauer

## SADRŽAJ

SADRŽAJ .....	I
POPIS SLIKA .....	II
POPIS TABLICA .....	III
POPIS OZNAKA .....	IV
SAŽETAK.....	V
SUMMARY .....	VI
1. UVOD .....	1
2. OPIS TEORIJSKIH DIJELOVA SUSTAVA.....	2
2.1. Umjetna inteligencija .....	2
2.2. Strojno učenje.....	4
2.3. Računalni vid .....	6
2.4. Detekcija i prepoznavanje lica .....	8
2.5. Prepoznavanje spola i raspona godina .....	9
2.6. Programski jezici za računalni vid, te izabrani programski jezik Python .....	11
2.7. Odabrana programska radna biblioteka - OpenCv.....	12
2.8. Neuronska mreža.....	14
3. RAZRADA PROGRAMSKOG KODA RAČUNALNE APLIKACIJE.....	15
3.1. Pokretanje aplikacije i uvoz potrebnih modula.....	15
3.2. Definiranje funkcije isticanja lica – highlighFace().....	15
3.3. Definiranje argumenata naredbenog retka i staza za model prepoznavanja lica .....	17
3.4. Glavni dio koda – definiranje srednjih vrijednosti modela, dobnih i spolnih kategorija, te učitavanje videa s ulaza.....	18
3.5. Primjena prepoznavanja lica, označivanje detektiranog lica .....	19
3.6. Primjena klasifikacije po dobi i spolu na detektirano lice, prikaz rezultata s predviđanjima i kraj programa .....	20
3.7. Vizualizacija aplikacije ukupnim dijagramom toka naredbi programa .....	22
3.8. Neuronske mreže programa .....	23
4. ANALIZA DOBIVENIH REZULTATA .....	26
5. PRIMJENA .....	29
6. KRITIČKI OSVRT .....	30
7. ZAKLJUČAK .....	31
LITERATURA.....	32
PRILOZI.....	34

**POPIS SLIKA**

Slika 1.	Računalno prepoznavanje lica.....	1
Slika 2.	Načini i vrste strojnog učenja.....	4
Slika 3.	Prikaz strukture slojeva neuronske mreže.....	14
Slika 4.	Ukupni dijagram toka naredbi programa.....	22
Slika 5.	Osnove rada perceptrona.....	23
Slika 6.	Primjer trenirane mreže sa optimiranim parametrima i težinama slojeva.....	24
Slika 7.	Konačni rezultati prikazani na autoru.....	26
Slika 8.	Rezultati prikazani na osobi mlađe dobi.....	26
Slika 9.	Rezultati prikazani na djetetu.....	27
Slika 10.	Rezultati prikazani na osobi srednje dobi.....	27
Slika 11.	Rezultati prikazani na osobi starije životne dobi.....	28

## **POPIS TABLICA**



**POPIS OZNAKA**

<b>Oznaka</b>	<b>Opis</b>
CV	Računalni vid (Computer vision)
AI	Umjetna računalna inteligencija (Artificial intelligence)
ML	Strojno učenje (Machine learning)
OOP	Objektno orijentirano programiranje (Object-oriented programming)
PP	Proceduralno programiranje (Procedural programming)
Caffe	Konvolucijska arhitektura za brzo ugrađivanje značajki
CNN	Konvolucijska neuronska mreža
OCR	Optičko prepoznavanje znakova

## **SAŽETAK**

Rad ukratko opisuje proces kreiranja programa za detekciju i prepoznavanje lica. Prvotno se opisuju potrebne naredbe u softveru za kreiranje programa te onda njihova svrha i primjena. Zatim je objašnjen detaljan opis postupka kreiranja dodatnih funkcija za prepoznavanja spola i raspona godina računalnim vidom. Naposljetku su prikazani rezultati te kritički osvrt na rad konačne verzije programa.

Ključne riječi : programiranje, računalni vid, prepoznavanje lica, detekcija lica

## **SUMMARY**

The paper briefly describes the process of creating a program for face detection and face recognition.

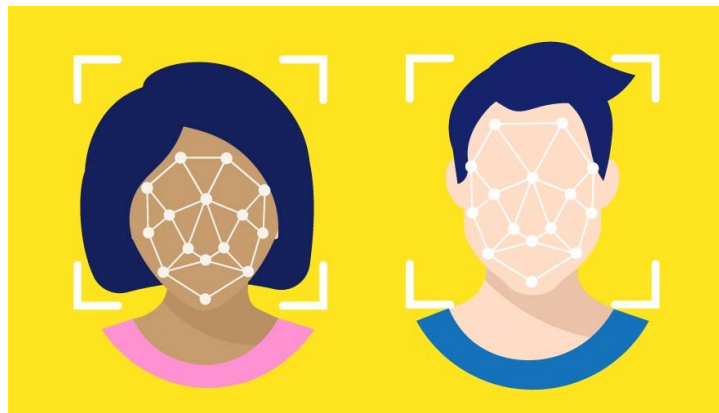
Initially, the necessary commands in the software for creating programs are described, and then their purpose and application. Then a detailed description of the procedure for creating an additional function for gender and age range recognition using computer vision was explained.

Finally, the results and a critical review of the work of the final version of the program are presented.

Key phrases: software programming, computer vision, facial detection, facial recognition

## 1. UVOD

Razvoj tehnologije prepoznavanja lica je i impresivan i zabrinjavajući, a o njezinom utjecaju na društvo još uvijek se raspravlja. Kako se tehnologija nastavlja razvijati, bit će zanimljivo vidjeti kako će se koristiti i regulirati u budućnosti. Tehnologija prepoznavanja lica prešla je dug put od svojih početaka, a njezin razvoj dugo je trajao. Zanimljiva činjenica o povijesti ove tehnologije je kako je rana tehnologija prepoznavanja lica razvijena 1960-ih i koristila je primitivne tehnike kao što je mjerenje crta lica ili udaljenost između očiju i duljina nosa za identifikaciju pojedinaca.[1] Trenutna tehnologija začeta je 1990-ih, a je počela postajati sofisticiranija s razvojem 3D slika, što je omogućilo veću točnost u identifikaciji pojedinaca.



Slika 1. Računalno prepoznavanje lica

## 2. OPIS TEORIJSKIH DIJELOVA SUSTAVA

Naš programski sustav povezuje umjetnu inteligenciju (AI), strojno učenje, računalni vid, mreže podataka, vrstu programiranja i programski jezik koji mora sve to objediniti. U prilogu je teorijski opis svih navedenih dijelova programskog sustava.

### 2.1. Umjetna inteligencija

Umjetna inteligencija brzo je rastuće područje računalne znanosti koje uključuje razvoj algoritama i modela koji omogućuju računalima da uče i donose predviđanja ili odluke na temelju podataka. Ideja o stvaranju strojeva koji mogu "misliti" i "učiti" desetljećima nas je fascinirala, a nedavni napredak u računalnoj snazi, dostupnosti podataka i algoritamskim inovacijama približio je tu viziju stvarnosti. Glavna značajka umjetne inteligencije je sposobnost prilagodbe i učenja iz podataka. U tradicionalnom programiranju, ljudski programer piše kod kako bi eksplicitno rekao računalu što treba učiniti. Nasuprot tome, u umjetnoj inteligenciji računalo uči kako izvršiti zadatak tako što mu se unose podaci i koristi statističke tehnike za prepoznavanje obrazaca i odnosa u podacima. Ova sposobnost učenja i prilagodbe čini AI prikladnim za širok raspon aplikacija, uključujući prepoznavanje slike i govora, obradu prirodnog jezika, autonomne sustave i mnoge druge. Unutar umjetne inteligencije postoji nekoliko različitih potpodručja, svako sa svojim područjima fokusa i metodologijama. Strojno učenje uključuje razvoj algoritama i modela koji omogućuju računalima da uče iz podataka bez eksplicitnog programiranja. Duboko učenje podskup je strojnog učenja koje koristi neuronske mreže, a to su modeli inspirirani strukturom i funkcijom ljudskog mozga. Učenje s pojačanjem još je jedno potpolje umjetne inteligencije koje uključuje obuku agenata za donošenje odluka u okruženju primanjem nagrada ili kazni na temelju njihovih postupaka. Jedno područje umjetne inteligencije koje je posljednjih godina doživjelo brzi rast i razvoj je računalni vid, koji uključuje omogućavanje računalima da tumače i razumiju vizualne podatke, poput slika i videa. Zato ćemo kasnije u daljnjem prilogu detaljno opisati računalni vid jer je ključna teorijska podloga našeg sustava. Kako se umjetna inteligencija nastavlja razvijati i sazrijevati, njezin je potencijal za transformaciju mnogih aspekata društva ogroman.

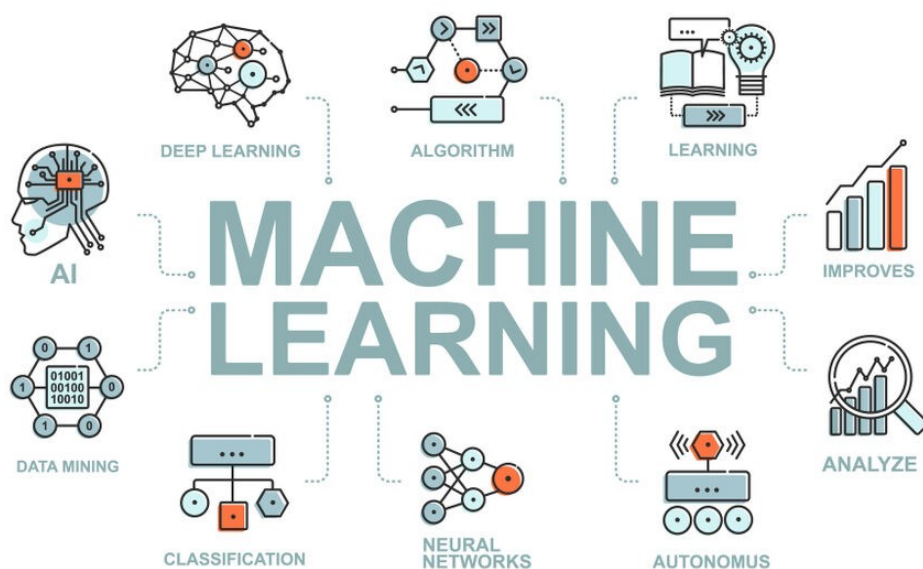
Neka od najzujbudljivijih područja primjene uključuju zdravstvo, gdje se AI koristi za razvoj novih dijagnostičkih alata i tretmana, proizvodnja, gdje se umjetna inteligencija koristi za optimizaciju opskrbnih lanaca i smanjenje otpada, i transport, gdje se AI koristi za razvoj autonomnih vozila i sustava za upravljanje prometom.

Međutim, brz tempo razvoja umjetne inteligencije također postavlja važna etička i društvena pitanja. Na primjer, postoji zabrinutost zbog mogućeg utjecaja umjetne inteligencije na tržište rada, budući da strojevi postaju sposobniji za obavljanje zadataka koje smo tradicionalno obavljali mi ljudi. Također postoji zabrinutost oko problema kao što su pristranost i privatnost, budući da algoritmi umjetne inteligencije mogu ovjekovječiti, pa čak i pojačati postojeće društvene i kulturološke predrasude ako se obučavaju na pristranim podacima. Unatoč tim zabrinutostima, potencijal umjetne inteligencije da transformira društvo i riješi neke od najhitnijih problema s kojima se čovječanstvo suočava je ogroman. Kao takvo, važno je da istraživači, kreatori politika i javnost rade zajedno kako bi osigurali da se umjetna inteligencija razvija na odgovoran i etičan način, s fokusom na povećanje koristi uz smanjenje rizika.

## 2.2. Strojno učenje

Strojno učenje podskup je umjetne inteligencije koji uključuje osposobljavanje računalnih algoritama za učenje iz podataka, bez eksplicitnog programiranja. To se postiže upotrebom statističkih tehnika, koje omogućuju računalu da identificira obrasce i odnose unutar podataka.

Strojno učenje koristi se u raznim aplikacijama, od prepoznavanja slika do obrade prirodnog jezika. Zapravo, mnoge tehnologije koje koristimo svaki dan, kao što su Siri i Alexa, oslanjaju se na strojno učenje kako bi funkcionirale. [2]



Slika 2. Načini i vrste strojnog učenja

Jedna od ključnih prednosti strojnog učenja je njegova sposobnost automatizacije zadataka koji bi inače bili dugotrajni ili nemogući za ljude. Na primjer, algoritmi strojnog učenja mogu se primijeniti za analiziranje velikih količina podataka, poput financijskih transakcija ili medicinske dokumentacije, kako bi se identificirali obrasci koje bi čovjeku bilo teško otkriti.

---

Strojno učenje postaje neophodan alat za mnoge industrije, uključujući financije, zdravstvo i tehnologiju. Kako količina generiranih podataka nastavlja rasti, očekuje se da će se potražnja za stručnjacima za strojno učenje samo povećavati.

Zaključno, strojno učenje moćan je alat koji ima potencijal promijeniti mnoge aspekte naših života. Dopuštajući računalima da uče iz podataka, možemo automatizirati zadatke, stvarati predviđanja i dobiti uvide koje bi bilo teško ili nemoguće postići tradicionalnim tehnikama programiranja. Iako strojno učenje nije bez izazova, ono je postalo ključna tehnologija za tvrtke i istraživače, a očekuje se da će njegov utjecaj značajno pojačati u nadolazećim godinama.



### 2.3. Računalni vid

Računalni vid je područje studija fokusirano na omogućavanje računalima da tumače i razumiju vizualni svijet oko sebe. To je podskup umjetne inteligencije koji se bavi obradom, analizom i interpretacijom vizualnih podataka iz stvarnog svijeta. Računalni vid je interdisciplinarno područje koje uključuje računalne znanosti, matematiku, inženjerstvo i psihologiju. Ima širok raspon primjena, uključujući obradu slike i videa, autonomna vozila, robotiku, medicinsko snimanje, sigurnost i nadzor i mnoge druge.

Porijeklo računalnog vida može se pratiti unazad do 1960-ih, kada su istraživači prvi počeli istraživati ideju korištenja računala za obradu i analizu vizualnih podataka. Rani rad na računalnom vidu bio je usredotočen na jednostavne zadatke poput otkrivanja rubova i prepoznavanja objekata. Međutim, kako je tehnologija napredovala, istraživači su počeli razvijati sofisticiranije algoritme i tehnike za analizu vizualnih podataka. Unatoč velikom napretku računalnog vida od 1960-ih godina, još uvijek postoji nekoliko izazova koje istraživači moraju prevladati. Jedan od najznačajnijih izazova je razvoj algoritama koji mogu obraditi i analizirati vizualne podatke u stvarnom vremenu, posebno u složenim i dinamičnim okruženjima.

Jedan od najznačajnijih napredaka u računalnom vidu posljednjih godina bio je razvoj algoritama dubokog učenja. Duboko učenje podskup je strojnog učenja koje koristi neuronske mreže za obradu i analizu velikih količina podataka. Uvježbavanjem ovih mreža na golemim količinama podataka, istraživači ih mogu naučiti prepoznati i kategorizirati objekte na slikama i videozapisima s visokim stupnjem točnosti. Duboko učenje korišteno je za razvoj nekih od najimpresivnijih aplikacija računalnog vida do danas, uključujući samovozeće automobile, sustave za prepoznavanje lica i alate za analizu medicinske slike. Ovi sustavi sposobni su obraditi ogromne količine podataka u stvarnom vremenu, što im omogućuje brzo i precizno donošenje složenih odluka.

Trenutni izazov je razvoj sustava računalnog vida koji mogu učinkovito raditi u slabo osvijetljenim okruženjima. Ovo je osobito važno za aplikacije poput autonomnih vozila i sigurnosnih i nadzornih sustava, gdje kvaliteta vizualnih podataka može biti ugrožena.

Računalni vid je područje koje se brzo razvija i ima širok raspon primjena u raznim industrijama sa potencijalom promijene načina na koji komuniciramo sa svijetom oko nas.

Jedno od područja u kojem računalni vid već ima značajan utjecaj je zdravstvo. Računalni vid se koristi za razvoj novih medicinskih tehnika snimanja koje mogu otkriti i dijagnosticirati bolesti na učinkovitiji i precizniji način. Na primjer, istraživači koriste algoritme računalnog vida za analizu medicinskih slika i otkrivanje znakova raka, omogućujući liječnicima da ih ranije dijagnosticiraju i time počnu s liječenjem ranije.[3] Osim toga, računalni vid se koristi za razvoj novih alata za kirurško planiranje i obuku. Korištenjem računalnog vida za analizu medicinskih slika i stvaranje 3D modela organa pacijenata, kirurzi mogu bolje planirati i pripremiti se za složene operacije. [4]

Računalni vid primjenu je našao i u razvoju novih iskustava virtualne i proširene stvarnosti. Korištenjem računalnog vida za praćenje pokreta i gesti korisnika, aplikacije virtualne i proširene stvarnosti mogu pružiti impresivnije i interaktivnije iskustvo.

Također, računalni vid se koristi za razvoj novih alata za stvaranje i proizvodnju sadržaja u industriji zabave. Na primjer, algoritmi računalnog vida mogu se koristiti za analizu snimke i prepoznavanje potencijalnih pogrešaka ili nedosljednosti, pomažući filmskim stvarateljima i montažerima da pojednostave proizvodni proces i poboljšaju kvalitetu konačnog proizvoda.

Sve u svemu, računalni vid ima potencijal transformirati način na koji pristupamo zabavi i stvaranju sadržaja, čineći ta područja učinkovitima i personaliziranima za korisnike. Kako istraživači nastavljaju razvijati nove algoritme i tehnike, potencijal računalnog vida da utječe na ova područja samo raste. Gledajući unaprijed, budućnost računalnog vida obećava, s novim aplikacijama i napretkom na horizontu.

## 2.4. Detekcija i prepoznavanje lica

Bilo je poziva na regulaciju tehnologije prepoznavanja lica, a Europski parlament, uvodi stroga pravila o njezinoj upotrebi. [5]

Tehnologija prepoznavanja lica također je izazvala zabrinutost oko privatnosti, budući da omogućuje praćenje i nadziranje pojedinaca bez njihova pristanka, oko čega se brine Europski GDPR. [6]

Prepoznavanje lica vrsta je biometrijske tehnologije koja koristi algoritme umjetne inteligencije (AI) za identifikaciju i provjeru pojedinaca na temelju njihovih crta lica. Prepoznavanje lica nije ograničeno samo na fotografije, već se može koristiti i na video snimkama za praćenje i identifikaciju ljudi u stvarnom vremenu. Tehnologija funkcionira tako što analizira jedinstvene karakteristike nečijeg lica, kao što je udaljenost između očiju, oblik čeljusti i konture nosa i usta. [7]

Prepoznavanje lica koristi se u raznim aplikacijama, uključujući sigurnost i nadzor, provedbu zakona, marketing i oglašavanje, pa čak i otključavanje pametnih telefona i drugih uređaja.

Međutim, postoji zabrinutost oko točnosti i mogućih pristranosti tehnologije prepoznavanja lica, osobito kada je riječ o identificiranju ljudi različitih rasa i etničkih pripadnosti. Također postoji zabrinutost oko privatnosti i potencijala zlouporabe tehnologije.

Tehnologija prepoznavanja lica može se koristiti u pasivnom i aktivnom načinu rada. U pasivnom načinu rada tehnologija skenira i analizira lica u gomili, dok se u aktivnom načinu rada koristi za identifikaciju određenih pojedinaca u bazi podataka.

Prepoznavanje lica nije sigurno i može biti sklono pogreškama, posebno kada su uvjeti osvjetljenja loši ili kada ljudi nose šešire, naočale ili druge dodatke koji im zaklanjaju lica.

Korištenje tehnologije prepoznavanja lica potaknulo je rasprave o privatnosti i građanskim slobodama, budući da su neki ljudi zabrinuti zbog moguće zlouporabe tehnologije od strane vlada i korporacija.

Gradovi i zemlje imaju različite pristupe na korištenje prepoznavanja lica, poput San Francisca, koji je zabranio korištenje tehnologije policiji i drugim gradskim odjelima. [8]

Druge zemlje, poput Velike Britanije i Kine, primijenile su tehnologiju u mnogo većoj mjeri.

## 2.5. Prepoznavanje spola i raspona godina

Prepoznavanje lica i prepoznavanje spola temelje se na algoritmima umjetne inteligencije (AI) koji analiziraju slike lica kako bi identificirali i klasificirali pojedince na temelju različitih fizičkih značajki.

Tehnologija prepoznavanja lica radi tako da prvo detektira i izdvoji značajke nečijeg lica sa slike ili video feeda, kao što su udaljenost između očiju, oblik nosa i konture obraza i čeljusti. Te se značajke zatim uspoređuju s bazom podataka pohranjenih poznatih lica kako bi se identificirala osoba. Prepoznavanje spola potpolje je prepoznavanja lica koje se posebno fokusira na identifikaciju spola osobe na temelju njezinih crta lica. Da bi to učinili, algoritmi umjetne inteligencije analiziraju niz fizičkih karakteristika, poput oblika linije čeljusti, udaljenosti između očiju i prisutnosti dlaka na licu ili šminke.

Algoritmi koriste te značajke za predviđanje spola osobe na slici ili video feedu. Na primjer, uglatija linija čeljusti i manji razmak između očiju češće se povezuju s muškim licima, dok se okruglija linija čeljusti i veći razmak između očiju češće povezuju s licima žena.

Međutim, važno je napomenuti da algoritmi za prepoznavanje lica i prepoznavanje spola nisu uvijek točni i postoji zabrinutost zbog njihove potencijalne pristranosti i netočnosti. Na primjer, pokazalo se da neki algoritmi imaju veću stopu pogrešaka pri identificiranju ljudi različitih rasa. Kao rezultat toga, u tijeku su napori da se poboljša točnost i pravednost ovih tehnologija kroz bolje podatke o obuci i algoritamske prilagodbe. [9]

Kako bi se poboljšala točnost prepoznavanja spola, algoritmi umjetne inteligencije često zahtijevaju velike skupove podataka označenih slika koje se koriste za obuku modela. Ti skupovi podataka mogu uključivati slike lica iz različitih kutova, uvjeta osvjetljenja i etničkog podrijetla kako bi se osiguralo da algoritam može točno prepoznati raznolik raspon lica.

Postoje i različiti pristupi prepoznavanju spola, uključujući korištenje neuronskih mreža, dubokog učenja i drugih tehnika strojnog učenja. Ovi pristupi mogu analizirati više crta lica istovremeno kako bi se napravila preciznija predviđanja o spolu.

Prepoznavanje spola koristi se u širokom rasponu aplikacija, od platformi društvenih medija koje ga koriste za pružanje personaliziranih preporuka, do agencija za provođenje zakona koje ga koriste za sigurnost i nadzor. Međutim, postoji i zabrinutost zbog moguće zlouporabe tehnologije prepoznavanja spola, osobito u kontekstu privatnosti i građanskih sloboda. Kao rezultat toga, u tijeku su rasprave o prikladnoj upotrebi priznavanja spola i potrebi za većom regulacijom i nadzorom.

Jedna zanimljiva činjenica o prepoznavanju spola je ta da ono ima primjenu izvan puke identifikacije spola osobe. Na primjer, neki istraživači istražuju korištenje algoritama za prepoznavanje spola kako bi bolje razumjeli kako spol utječe na izraze lica i emocije. Analizirajući kako se izrazi lica razlikuju među spolovima, istraživači bi mogli steći nove uvide u društvene i psihološke čimbenike koji utječu na ponašanje i donošenje odluka. To bi moglo imati potencijalne primjene u područjima kao što su marketing, oglašavanje i mentalno zdravlje. Prepoznavanje dobi ima i potencijalne primjene u području medicine. Na primjer, istraživači istražuju korištenje algoritama za prepoznavanje dobi za analizu medicinskih slika i predviđanje starosti pacijenta, što bi liječnicima moglo pomoći u dijagnosticiranju bolesti povezanih sa starenjem i u razvoju personaliziranih planova liječenja. Prepoznavanje dobi također bi se moglo koristiti za identifikaciju pacijenata koji su u riziku od stanja povezanih sa starenjem, kao što su osteoporoza ili kognitivni pad, i omogućiti rane intervencije za poboljšanje zdravstvenih ishoda. Osim toga, tehnologija prepoznavanja dobi mogla bi se koristiti za razvoj personaliziranih tretmana protiv starenja, kao što su proizvodi za njegu kože ili preporuke za stil života, na temelju jedinstvenog profila starenja osobe.

## 2.6. Programski jezici za računalni vid, te izabrani programski jezik Python

Osim matematičkih i teorijskih osnova, računalni vid zahtijeva implementaciju algoritama i tehnika korištenjem programskih jezika.

Evo nekih od najpopularnijih programskih jezika koji se koriste u računalnom vidu:

C++ : jezik visokih performansi koji se često koristi u aplikacijama računalnog vida koje zahtijevaju performanse u stvarnom vremenu, kao što su praćenje objekata, prepoznavanje gesta i proširena stvarnost. Biblioteke poput OpenCV i DLIB često se koriste s C++, a popularan je u industriji igara i automobilske industriji.

MATLAB : popularan alat za izradu prototipa i implementaciju algoritama računalnog vida, zahvaljujući jeziku visoke razine i ugrađenom alatnom okviru za obradu slike. Naširoko se koristi u akademskim istraživanjima i primjenama za izradu prototipova, ali se rjeđe koristi u industrijskim primjenama.

Java : popularan jezik za razvoj aplikacija računalnog vida za mobilne i web platforme. Često se koristi u aplikacijama kao što su prepoznavanje lica, proširena stvarnost i skeniranje crtičnog koda.

JavaScript : jezik za web-bazirane aplikacije računalnog vida, zahvaljujući bibliotekama kao što su Tensorflow.js i OpenCV.js koje olakšavaju pokretanje algoritama računalnog vida u pregledniku. Često se koristi u aplikacijama kao što su otkrivanje objekata, prepoznavanje slika i praćenje kretanja.

Lua: lagani jezik koji se često koristi u aplikacijama računalnog vida koje zahtijevaju performanse u stvarnom vremenu, kao što su motori videoigara i robotika. Često se koristi s okvirima kao što su Torch i Caffe.

Za kraj ćemo opisati jezik koji smo uzeli za izradu ovoga rada, Python te razloge izbora baš toga programskog jezika.

[10] Python : jedan od najpopularnijih jezika za računalni vid, zahvaljujući jednostavnosti upotrebe

i širokom rasponu dostupnih biblioteka. Biblioteka poput OpenCV(koji ćemo opisati u nastavku), Scikit-image i TensorFlow olakšavaju implementaciju algoritama računalnog vida, a Python se koristi u mnogim istraživačkim i industrijskim aplikacijama.

## 2.7. Odabrana programska radna biblioteka - OpenCv

OpenCV (Open Source Computer Vision Library) široko je korištena biblioteka i za strojno učenje otvorenog koda i za računalni vid. Prvi ga je 1999. razvio Intel, a danas ga održava OpenCV.org. te je osmišljen kako bi pružio zajedničku infrastrukturu za aplikacije računalnog vida i ubrzao korištenje strojne percepcije u komercijalnim proizvodima.

Biblioteka je napisana u C++ i ima sučelja za Python, Java i MATLAB. OpenCV uključuje i preko 2500 optimiziranih algoritama koji se mogu koristiti za prepoznavanje objekata, otkrivanje i prepoznavanje lica, klasificiranje ljudskih radnji u videozapisima, praćenje pokreta kamere i još mnogo toga. Podržava različite platforme, uključujući Windows, Linux, Android, iOS i Raspberry Pi.

OpenCV-a ima mnogo ključnih značajki koje ga čine popularnim izborom među programerima. Neke od ovih značajki su da je jednostavan za korištenje jer OpenCV ima intuitivan API koji programerima olakšava korištenje i implementaciju. OpenCV je višeplatformski, što znači da može raditi na različitim operativnim sustavima i hardveru. Štoviše optimiziran je za visoke performanse, što znači da može brzo obraditi velike količine podataka. Također ima i ugrađenu podršku za algoritme strojnog učenja, koji se mogu koristiti za stvaranje modela za otkrivanje, prepoznavanje i praćenje objekata.

Valja podržati i to što ima veliku zajednicu programera koji doprinose biblioteci i nude podršku drugima. Zato OpenCV ima veliku kolekciju uzoraka koda i vodiča koji programerima olakšavaju početak rada s bibliotekom. To pomaže u smanjenju krivulje učenja za nove programere.

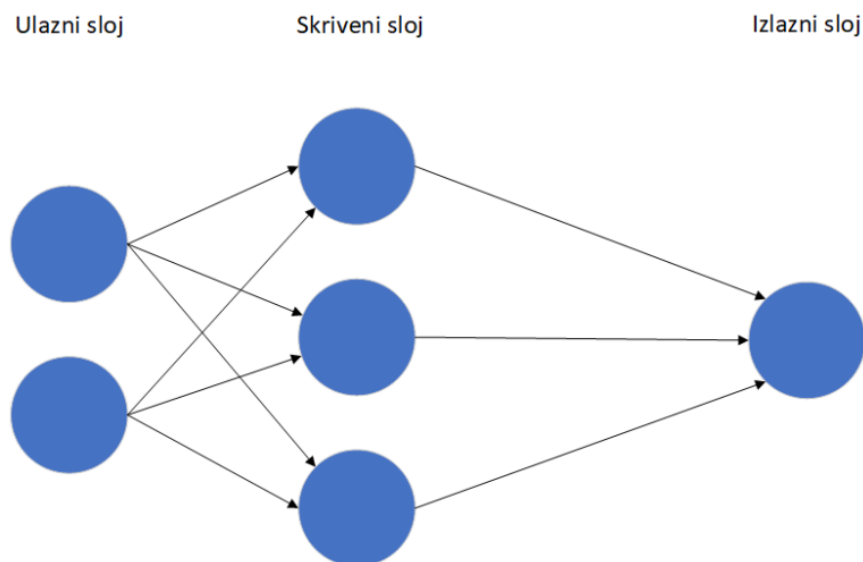
OpenCV ima širok raspon primjena u računalnom vidu i strojnom učenju. Može se koristiti za otkrivanje objekata na slikama i videozapisima, poput lica, što ima primjenu u sigurnosti i nadzoru. Zanimljiva primjena je i za stvaranje aplikacija proširene stvarnosti, koje prekrivaju digitalni sadržaj na stvarnom svijetu. Optičko prepoznavanje znakova (OCR) se koristi za prepoznavanje teksta na slikama, što ima primjenu u skeniranju i digitalizaciji dokumenata. Bitna industrijska primjena je i za izradu inteligentnih robota koji mogu navigirati i komunicirati s okolinom.

Zaključno, OpenCV je moćna i svestrana biblioteka računalnog vida i strojnog učenja otvorenog koda. Ima širok raspon značajki i aplikacija koje ga čine popularnim izborom među programerima. Sa svojim algoritmima visokih performansi i podrškom za više platformi, OpenCV je postao ključni alat za razvoj aplikacija računalnog vida u raznim navedenim industrijama.



## 2.8. Neuronska mreža

Neuronska mreža je računalni model inspiriran biološkim neuronima korišten za obradu informacija, posebno u strojnom učenju i dubokom učenju.



**Slika 3. Struktura slojeva neuronske mreže, svaka plava točka označava jedan perceptron**

Sastoji se od slojeva umjetnih neurona povezanih težinskim faktorima koji se optimiziraju kako bi model naučio određeni zadatak. Trenirane neuronske mreže su one koje su prošle proces učenja na velikom skupu podataka i prilagodile svoje težine kako bi postigle željeni rezultat, dok netrenirane mreže nisu prošle kroz taj proces i obično nemaju korisnu funkcionalnost izvan svoje inicijalne konfiguracije.

Postoje različite arhitekture neuronskih mreža, uključujući konvolucijske mreže za obradu signala poput zvuka i slike, rekurentne mreže za obradu sekvencijalnih podataka i duboke neuronske mreže za složenije zadatke.

### 3. RAZRADA PROGRAMSKOG KODA RAČUNALNE APLIKACIJE

#### 3.1. Pokretanje aplikacije i uvoz potrebnih modula

```
import cv2
import math
import argparse
```

Kao što vidimo iz priloženog za obradu slika i videozapisa potreban nam je modul OpenCV 'cv2'. Ključnu ulogu igra i modul za matematičke operacije 'math' koji služi za usporedbu srednjih vrijednosti modela lica sa ulaznom jedinicom. Za prihvatanje slike kao ulaznog argumenta potreban nam je modul za obradu argumenata naredbenog retka odnosno 'argparse'.

#### 3.2. Definiranje funkcije isticanja lica – highlightFace()

```
def highlightFace(net, frame, conf_threshold=0.7):
    frameOpencvDnn=frame.copy()
    frameHeight=frameOpencvDnn.shape[0]
    frameWidth=frameOpencvDnn.shape[1]
    blob=cv2.dnn.blobFromImage(frameOpencvDnn, 1.0, (300, 300), [104, 117, 123],
    True, False)

    net.setInput(blob)
    detections=net.forward()
    faceBoxes=[]
    for i in range(detections.shape[2]):
        confidence=detections[0,0,i,2]
        if confidence>conf_threshold:
            x1=int(detections[0,0,i,3]*frameWidth)
            y1=int(detections[0,0,i,4]*frameHeight)
            x2=int(detections[0,0,i,5]*frameWidth)
            y2=int(detections[0,0,i,6]*frameHeight)
            faceBoxes.append([x1,y1,x2,y2])
            cv2.rectangle(frameOpencvDnn, (x1,y1), (x2,y2), (0,255,0),
            int(round(frameHeight/150)), 8)
    return frameOpencvDnn,faceBoxes
```

Ovaj dio koda predstavlja funkciju nazvanu 'highlightFace', koja se koristi za naglašavanje lica na ulaznom video okviru. Ova funkcija je bitna komponenta za detekciju lica na ulaznom videu i označava detektirana lica pravokutnicima kako bi se dalje analizirala dob i spol tih lica. Evo detaljnog opisa što radi svaki korak unutar ove funkcije.

Funkcija 'highlightFace' koja igra ključnu ulogu u procesu detekcije lica na ulaznom video okviru prima tri glavna argumenta: 'net' (neuronsku mrežu za detekciju lica), 'frame' (ulazni video okvir koji se analizira) te opcionalni prag pouzdanosti ('conf\_threshold') za detekciju lica.

Prvo, kopira se ulazni video okvir 'frame' kako bi se osiguralo da se izmjene ne primjenjuju izravno na originalnom okviru. Nakon toga, funkcija dobiva dimenzije kopiranog okvira, visinu i širinu, koje će koristiti za pravilno skaliranje i postavljanje koordinata detektiranih lica.

Zatim slijedi značajan korak u kojem se pripremaju binarni veliki objekti (engl. "binary large objects" ili "blobovi") iz kopiranog okvira 'frame'. Blobovi su oblik podataka koji se koristi za unos u neuronsku mrežu i ti objekti sadrže informacije o slikama koje se obrađuju. Točnije binarni veliki objekti (blobovi), su posebni oblik podataka koji sadrži podatke sa informacijama o skaliranju slike, normalizaciji boja i drugim transformacijama potrebnim za unos slike u neuronsku mrežu. Binarni veliki objekti koriste se kako bi se osiguralo da ulazni podaci odgovaraju očekivanjima modela dubokog učenja, omogućujući mu precizno izvršavanje detekcije lica. Nakon što su binarni veliki objekti pripremljeni, postavljaju se kao ulaz za neuronsku mrežu 'net' pomoću funkcije 'net.setInput()'. To omogućuje mreži da analizira ulazni okvir i izvrši detekciju lica. Detekcija lica izvodi se pozivom funkcije 'net.forward()', a zatim se provjerava pouzdanost svake detekcije. Ako je pouzdanost (vjerojatnost) detekcije lica veća od zadane granice 0.7 ('conf\_threshold'), detekcija se smatra valjanom. Za svaku valjanu detekciju, izračunavaju se koordinate lica na okviru kako bi se označila njihova pozicija.

Konačno, označavanjem lica na kopiranom okviru 'frame' obavlja se crtanje pravokutnika oko svakog detektiranog lica. Pravokutnici su zeleni i jasno označavaju poziciju lica na okviru.

### 3.3. Definiranje argumenata naredbenog retka i staza za model prepoznavanja lica

```
parser=argparse.ArgumentParser()
parser.add_argument('--image')

args=parser.parse_args()

faceProto="opencv_face_detector.pbtxt"
faceModel="opencv_face_detector_uint8.pb"
ageProto="age_deploy.prototxt"
ageModel="age_net.caffemodel"
genderProto="gender_deploy.prototxt"
genderModel="gender_net.caffemodel"
```

U Pythonu, ‘argparse’ je ugrađena biblioteka koja omogućuje jednostavno definiranje i obradu naredbenih argumenata. Sa naredbenim argumentima možemo prilagoditi ponašanje skripte ili proslijediti informacije. U ovoj skripti koja obrađuje slike, koristeći naredbeni argument ‘--image’ specificiramo putanju do slike koju želimo obraditi, ‘argparse’ parser omogućuje programerima da definiraju koji su naredbeni argumenti važeći za njihovu skriptu, što čini skriptu fleksibilnijom i olakšava interakciju s korisnicima.

Naknadno, u ovom dijelu koda definiraju se staze za učitavanje modela prepoznavanja lica, dobi i spola, zajedno s popisima za klasifikaciju dobi i spola. Ove staze su bitne jer omogućavaju aplikaciji pristup treniranim modelima i konfiguracijama za prepoznavanje različitih karakteristika lica.

‘FaceProto’ i ‘faceModel’ predstavljaju staze do konfiguracijske i modelske datoteke za prepoznavanje lica, dok ‘ageProto’ i ‘ageModel’ pružaju staze za prepoznavanje dobi, a ‘genderProto’ i ‘genderModel’ za prepoznavanje spola.

### 3.4. Glavni dio koda – definiranje srednjih vrijednosti modela, dobnih i spolnih kategorija, te učitavanje videa s ulaza

```
MODEL_MEAN_VALUES=(78.4263377603, 87.7689143744, 114.895847746)
ageList=['(0-2)', '(4-6)', '(8-12)', '(15-20)', '(20-25)', '(38-43)',
'(48-53)', '(60-100)']
genderList=['Male', 'Female']
faceNet=cv2.dnn.readNet(faceModel, faceProto)
ageNet=cv2.dnn.readNet(ageModel, ageProto)
genderNet=cv2.dnn.readNet(genderModel, genderProto)

video=cv2.VideoCapture(args.image if args.image else 0)
padding=20
```

Definira se 'MODEL\_MEAN\_VALUES', to je niz srednjih vrijednosti koji se koristi za normalizaciju slika prije nego što se podaci unesu u model za prepoznavanje dobi. Također, 'ageList' i 'genderList' su popisi klasifikacija dobi i spola koje model koristi za generiranje rezultata. Ovi parametri i definicije ključni su za pravilno funkcioniranje aplikacije i ispravno tumačenje rezultata dobivenih iz modela za prepoznavanje lica, dobi i spola.

Naknadne sljedeće tri linije, 'faceNet=cv2.dnn.readNet(faceModel, faceProto)', 'ageNet=cv2.dnn.readNet(ageModel, ageProto)' i 'genderNet=cv2.dnn.readNet(genderModel, genderProto)', obavljaju inicijalizaciju neuronskih mreža za prepoznavanje lica, procjenu dobi i određivanje spola. Ovdje se koristi OpenCV (cv2) za čitanje prethodno obučениh modela, gdje već navedeni 'faceModel', 'ageModel' i 'genderModel' kao što je spomenuto predstavljaju putanje do modela, a 'faceProto', 'ageProto' i 'genderProto' predstavljaju putanje do konfiguracijskih datoteka tih navedenih modela. Nakon izvođenja ovih linija, 'faceNet', 'ageNet' i 'genderNet' su spremni za korištenje za svoje specifične zadatke.

Zatim, linija 'video=cv2.VideoCapture(args.image if args.image else 0)' stvara objekt za snimanje videa pomoću OpenCV. Ovdje se koristi 'cv2.VideoCapture', a argument 'args.image' sadrži putanju do slike ili videozapisa koji se želi obraditi. Ako korisnik nije specificirao putanju kroz naredbeni argument '--image', tada se postavlja na 0, što znači da će se koristiti ugrađena kamera računala. Ovo omogućuje skripti da otvori izvor videa i započne procesiranje kad se izvrši. Na kraju, 'padding=20' je jednostavna linija koja postavlja varijablu 'padding' na vrijednost 20. Ova vrijednost predstavlja dodatnu okolinu (granica) oko detektiranih lica kako bi se poboljšala preciznost i vizualni izlaz pri crtanju kvadrata oko lica.

### 3.5. Primjena prepoznavanja lica, označivanje detektiranog lica

```
while cv2.waitKey(1)<0:
    hasFrame, frame=video.read()
    if not hasFrame:
        cv2.waitKey()
        break

    resultImg, faceBoxes=highlightFace(faceNet, frame)
    if not faceBoxes:
        print("No face detected")

    for faceBox in faceBoxes:
        face=frame[max(0, faceBox[1]-padding):
                    min(faceBox[3]+padding, frame.shape[0]-1), max(0, faceBox[0]-padding)
                    :min(faceBox[2]+padding, frame.shape[1]-1)]
```

U ovom dijelu koda stvaramo dinamičnu i kontinuiranu analizu videozapisa radi prepoznavanja lica i njihovog označavanja. Ovaj proces se odvija u petljama i ima nekoliko ključnih koraka. Za početak, koristimo ‘while’ petlju za čitanje i analizu videozapisa, čime omogućavamo kontinuirano praćenje lica u stvarnom vremenu. U svakom koraku petlje, čitamo sljedeći okvir (‘frame’) iz videozapisa i provjeravamo ima li tog okvira (‘hasFrame’). Ako nema okvira, znači da smo došli do kraja videa i izlazimo iz petlje.

Zatim, pozivamo funkciju ‘highlightFace’ kako bismo detektirali lica na trenutnom okviru ‘frame’. Rezultat te detekcije je ‘resultImg’, koji je kopija ulaznog okvira s označenim detektiranim licima, te ‘faceBoxes’, koji sadrži koordinate detektiranih lica.

Ako nema detektiranih lica, ispisuje se poruka "No face detected", što znači da na trenutnom okviru nisu pronađena lica. U slučaju da postoje detektirana lica (zabilježena u ‘faceBoxes’), koristimo petlju ‘for faceBoxes’ za prolazak kroz svako detektirano lice. Za svako lice, izračunavaju se koordinate pravokutnika oko lica kako bi se izoliralo samo područje lica za dodatnu analizu.

Ova petlja omogućava postupno i kontinuirano označavanje svih detektiranih lica na svakom okviru videozapisa. Kroz ovaj postupak, stvaramo vizualni trag lica na videozapisu, što je ključno za daljnju analizu i prepoznavanje dobi i spola za svako od tih označenih lica.

### 3.6. Primjena klasifikacije po dobi i spolu na detektirano lice, prikaz rezultata s predviđanjima i kraj programa

```
blob=cv2.dnn.blobFromImage(face, 1.0, (227,227), MODEL_MEAN_VALUES,
swapRB=False)
genderNet.setInput(blob)
genderPreds=genderNet.forward()
gender=genderList[genderPreds[0].argmax()]
print(f'Gender:{gender}')

ageNet.setInput(blob)
agePreds=ageNet.forward()
age=ageList[agePreds[0].argmax()]
print(f'Age: {age[1:-1]} years')

cv2.putText(resultImg, f'{gender}, {age}', (faceBox[0], faceBox[1]-10),
cv2.FONT_HERSHEY_SIMPLEX, 3.0, (0,255,255), 2, cv2.LINE_AA)

resultImg = cv2.resize(resultImg, None, fx=0.5, fy=0.5,
interpolation=cv2.INTER_AREA) #Dodani_Resize

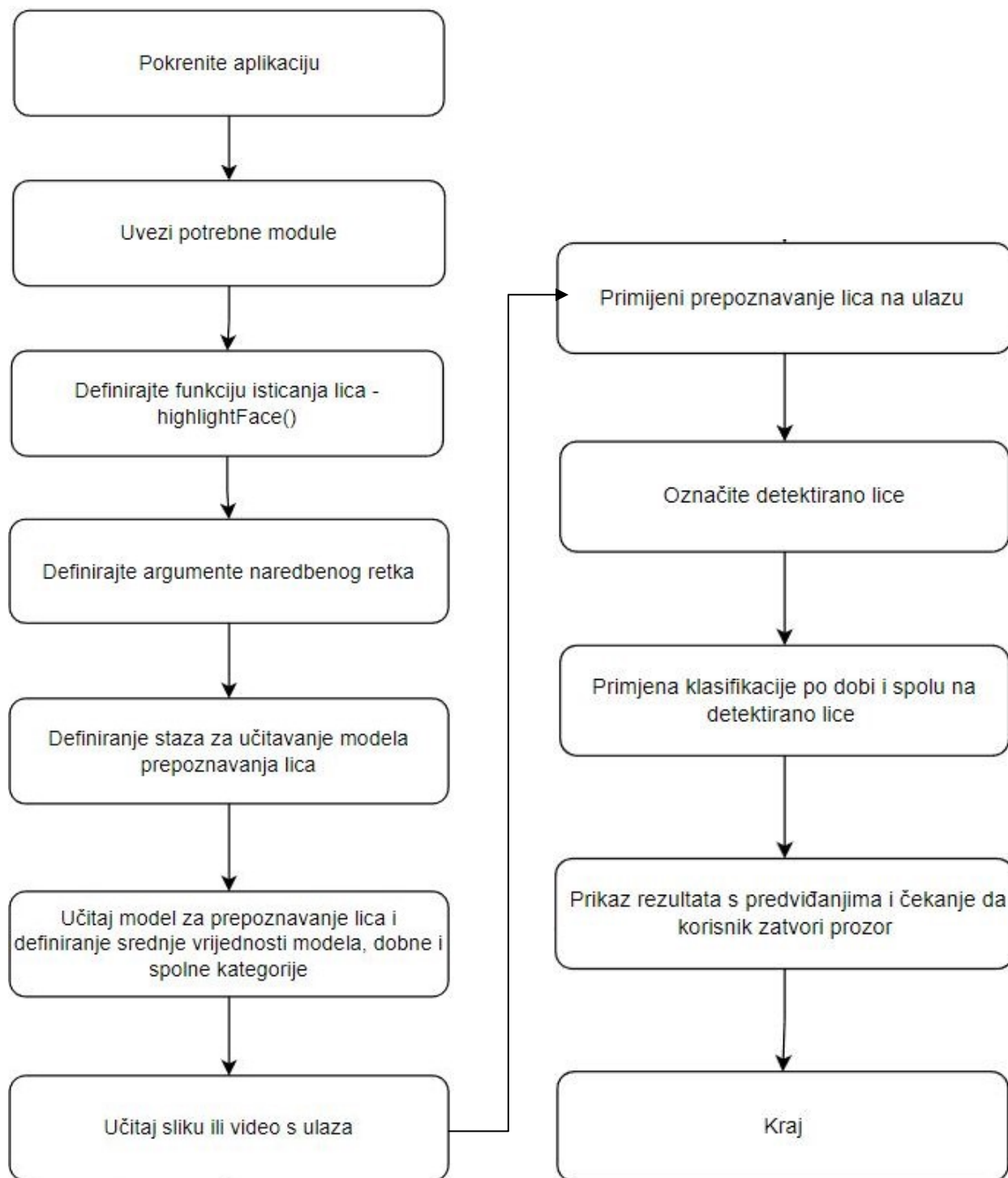
cv2.imshow("Detecting age and gender", resultImg)
```

U ovom dijelu koda provodimo proces klasifikacije po dobi i spolu za svako prethodno detektirano lice, te prikazujemo rezultate predviđanja na okviru videa. Ovaj postupak se odvija korak po korak kako bismo dobili informacije o dobi i spolu za svako lice i vizualno označili rezultate. Počinjemo koristeći funkciju 'cv2.dnn.blobFromImage' kako bismo pripremili binarni veliki objekt (blob) iz izdvojenog lica. Ovaj blob sadrži informacije o slici, uključujući normalizirane boje i veličinu (227x227 piksela), kako bi se mogao koristiti kao ulaz u modele za prepoznavanje dobi i spola. Nakon toga, postavljamo ovaj blob kao ulaz za modele prepoznavanja dobi i spola, odnosno 'ageNet' i 'genderNet'. Korištenjem funkcije 'forward', dobivamo predviđanja za svaku od tih karakteristika. Zatim, iz rezultata predviđanja za spol i dob, dobivamo konačne klasifikacije. Na primjer, za spol, koristimo 'genderList' kako bismo dobili spol koji model smatra najvjerojatnijim. Slično, za dob, koristimo 'ageList' kako bismo dobili interval dobi s najvećom vjerojatnošću.

Rezultati predviđanja za spol i dob ispisuju se na ekranu, prateći format "Gender: {spol}" i "Age: {dob}". Kako bismo vizualno označili lice s dobivenim informacijama o spolu i dobi, koristimo funkciju 'cv2.putText' kako bismo napisali tekst na okviru oko lica. Ovdje se koristi stil fonta, boja i pozicija kako bi se informacije čitale jasno i vidljivo. Nakon što smo dodali tekst na okvir, smanjujemo veličinu rezultirajućeg okvira s rezultatima (kako bi bio kompaktniji) pomoću funkcije 'cv2.resize'. Konačno, koristimo 'cv2.imshow' kako bismo prikazali rezultate prepoznavanja dobi i spola na ekranu. Ovaj korak omogućuje stvarno vrijeme praćenja i vizualizaciju rezultata na videozapisu.



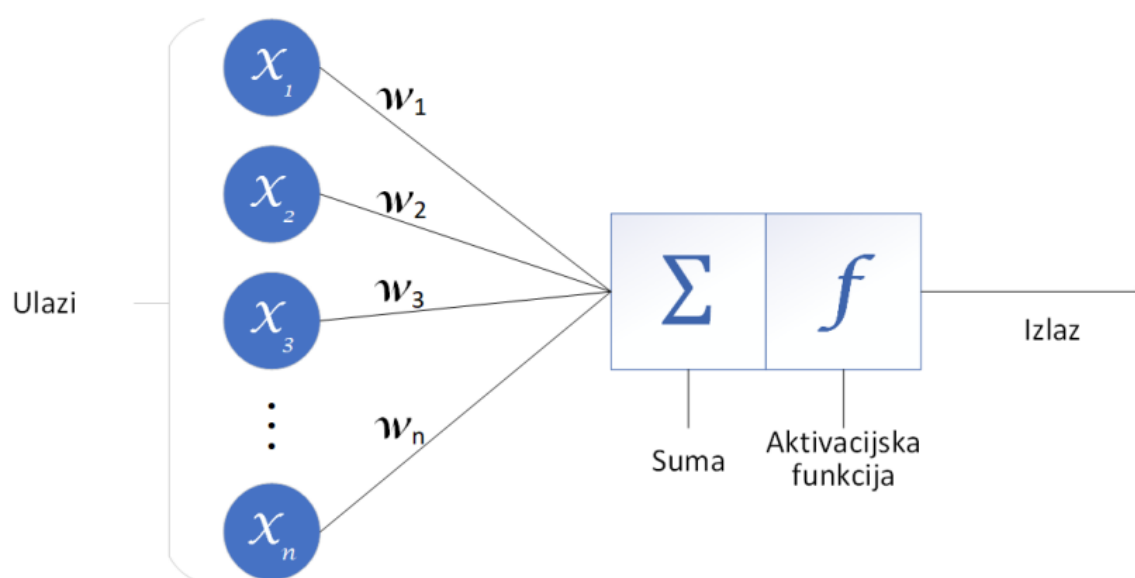
### 3.7. Vizualizacija aplikacije ukupnim dijagramom toka naredbi programa



Slika 4. Ukupni dijagram toka naredbi programa

### 3.8. Neuronske mreže programa

Modeli za prepoznavanje godina i spola su trenirani kako bi se mogle klasificirati godine/spol na temelju slika lica, a .caffemodel datoteke koje imamo sadrže naučene težine i parametre mreže nakon završenog treninga. Treniranje neuronske mreže podrazumijeva postupak optimizacije težina kako bi se minimizirala greška klasifikacije na trening skupu podataka. Tijekom ovog procesa, model se prilagođava tako da bolje odgovara ulaznim podacima i ciljnoj klasifikaciji.

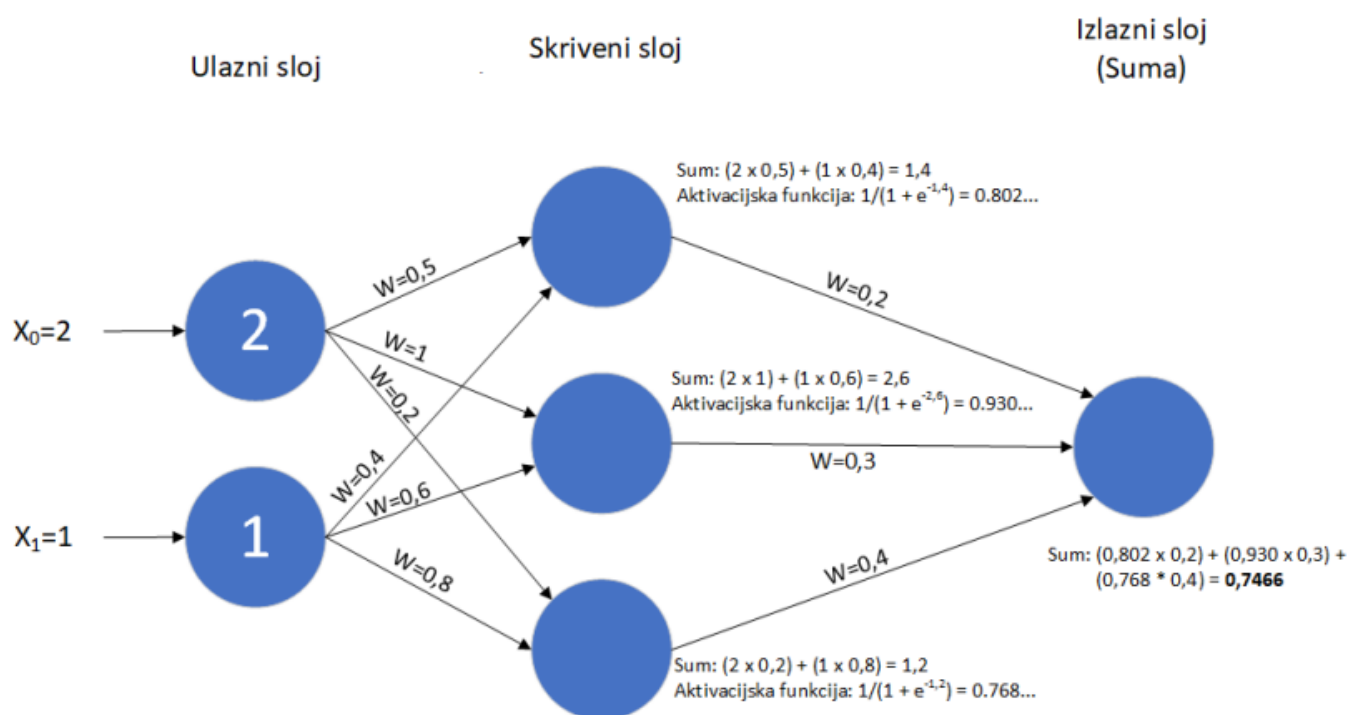


Slika 5. Osnove rada perceptrona

Treniranje utječe na arhitekturu mreže jer se tijekom tog procesa optimiziraju težine svih slojeva kako bi se postigla bolja klasifikacijska točnost. Učenje težina slojeva omogućuje mreži da prepozna relevantne značajke u ulaznim slikama i donosi bolje odluke pri klasifikaciji. Treniranjem mreže za prepoznavanje spola i godina postignuto je stvaranje modela dubokog učenja sposobnog za automatsko prepoznavanje i klasifikaciju spola te dobne skupine pojedinca na temelju slika lica. Treniranje mreže omogućava joj da nauči relevantne obrasce i značajke u slikama te tako donese precizne zaključke o spolu i dobi pojedinca, čime se otvaraju vrata raznovrsnim aplikacijama u kojima je potrebna automatska analiza vizualnih podataka.

Treniranjem neuronske mreže za prepoznavanje spola i godina, glavni cilj je optimizirati težine i parametre mreže. Točnije, optimira se funkcija gubitka (loss function) mreže, koja kvantificira razliku između stvarnih rezultata (pravilnih klasifikacija) i predviđenih rezultata mreže. Cilj optimizacije je minimiziranje tj. minimizirati ovu funkciju gubitka. [11]

Tijekom treninga, neuronska mreža prilagođava svoje težine tako da s vremenom postaje bolja u prepoznavanju i klasifikaciji različitih uzoraka u ulaznim podacima, koji u ovom slučaju predstavljaju slike lica. Kroz iteracije treninga, mreža "uči" relevantne značajke i obrasce kako bi što preciznije predviđala spol i dob pojedinca. Optimizacijom se također može kontrolirati i regulirati kako mreža generalizira svoje sposobnosti. Cilj je postići dobru performansu ne samo na podacima na kojima je trenirana, već i na novim i nepoznatim podacima, čime se osigurava da mreža ne bude preopterećena treniranjem. U konačnici, optimizacijom tijekom treninga mreža postaje bolja u prepoznavanju spola i dobi na nepoznatim slikama lica, što je glavna svrha ovog procesa.

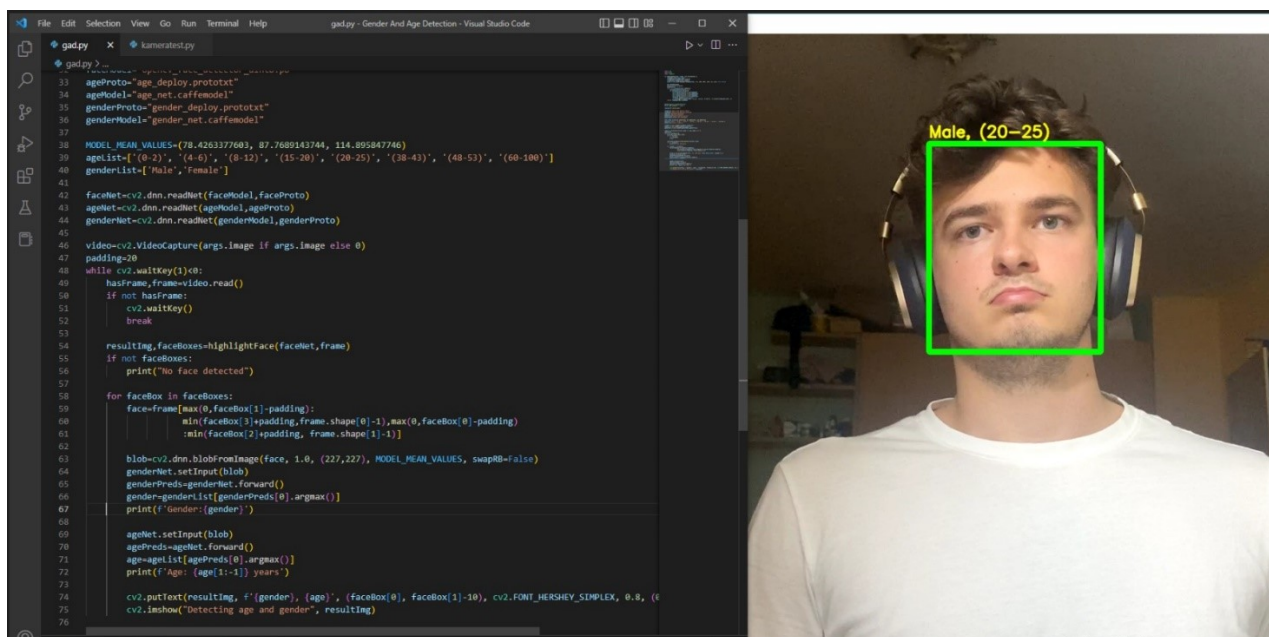


Slika 6. Primjer trenirane mreže sa optimiranim parametrima i težinama slojeva

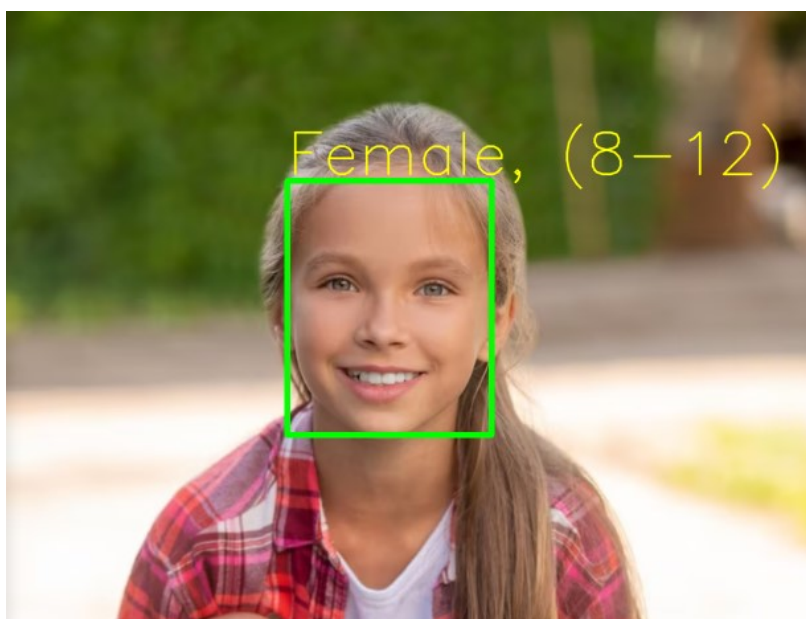
"CaffeNet" je arhitektura neuronske mreže koja se koristi u kodu za prepoznavanje spola i dobi na slikama lica. [12] Ova arhitektura temelji se na popularnom deep learning frameworku nazvanom Caffe (Convolutional Architecture for Fast Feature Embedding). Dakle "CaffeNet" je varijanta Convolutional Neural Network (CNN) arhitekture koja je slična AlexNetu [13], poznatom modelu koji je postigao izvanredne rezultate na natjecanju ImageNet [14]. CaffeNet ima duboku strukturu s nekoliko slojeva konvolucije, slojevima normalizacije, slojevima ReLU aktivacije te potpornim slojevima za izvlačenje značajki iz slika. Ova arhitektura je posebno prilagođena za obradu slika i sposobna je naučiti složene obrasce i značajke iz slikovnih podataka. CaffeNet se koristi za ekstrakciju značajki iz lica na slikama, a zatim se te značajke koriste za prepoznavanje spola i dobi pojedinca. Važno je napomenuti da je arhitektura CaffeNet-a definirana u konfiguracijskim datotekama poput "gender\_deploy.prototxt" i "age\_deploy.prototxt", dok su težine i parametri mreže pohranjeni u odvojenim datotekama kao što je "gender\_net.caffemodel" ili "age\_net.caffemodel". Tijekom treninga, ove težine i parametri su optimizirani kako bi se postigla što bolja točnost u prepoznavanju spola i dobi na slikama lica.

## 4. ANALIZA DOBIVENIH REZULTATA

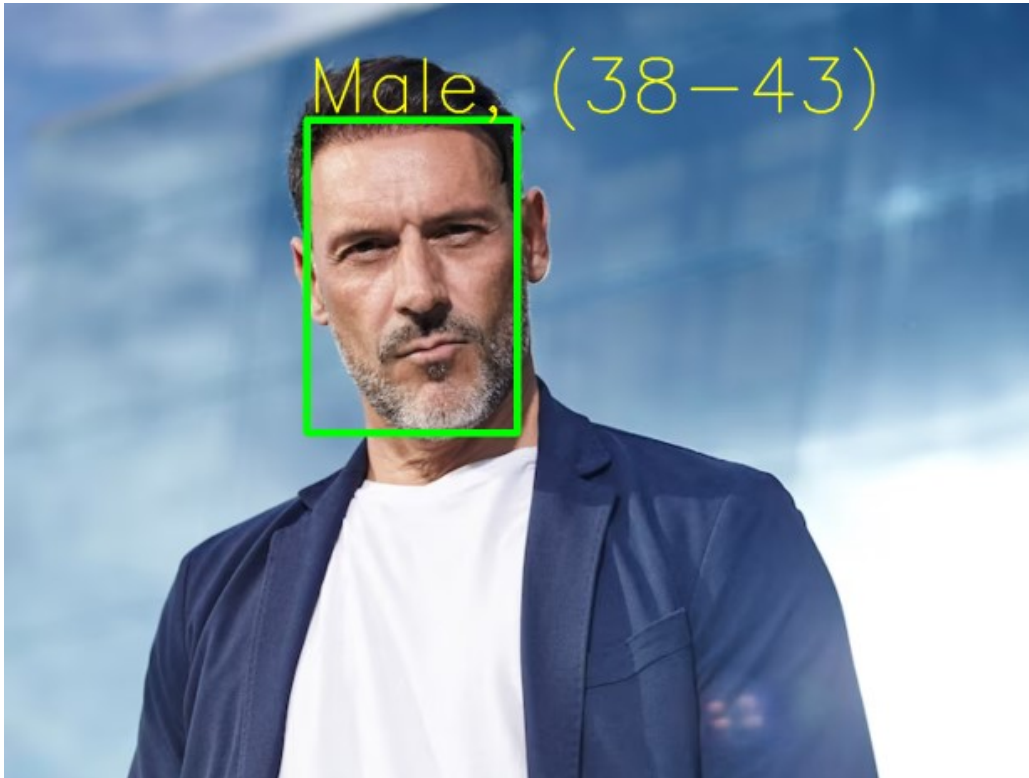
Rezultati koji su donijeli točno prepoznavanje spola i dobne skupine na temelju dobre slike lica su iznimno zadivljujući. Analiza ovih dobivenih rezultata ukazuje na fascinantne mogućnosti i izazove koje donosi napredak u računalnoj viziji i dubokom učenju.



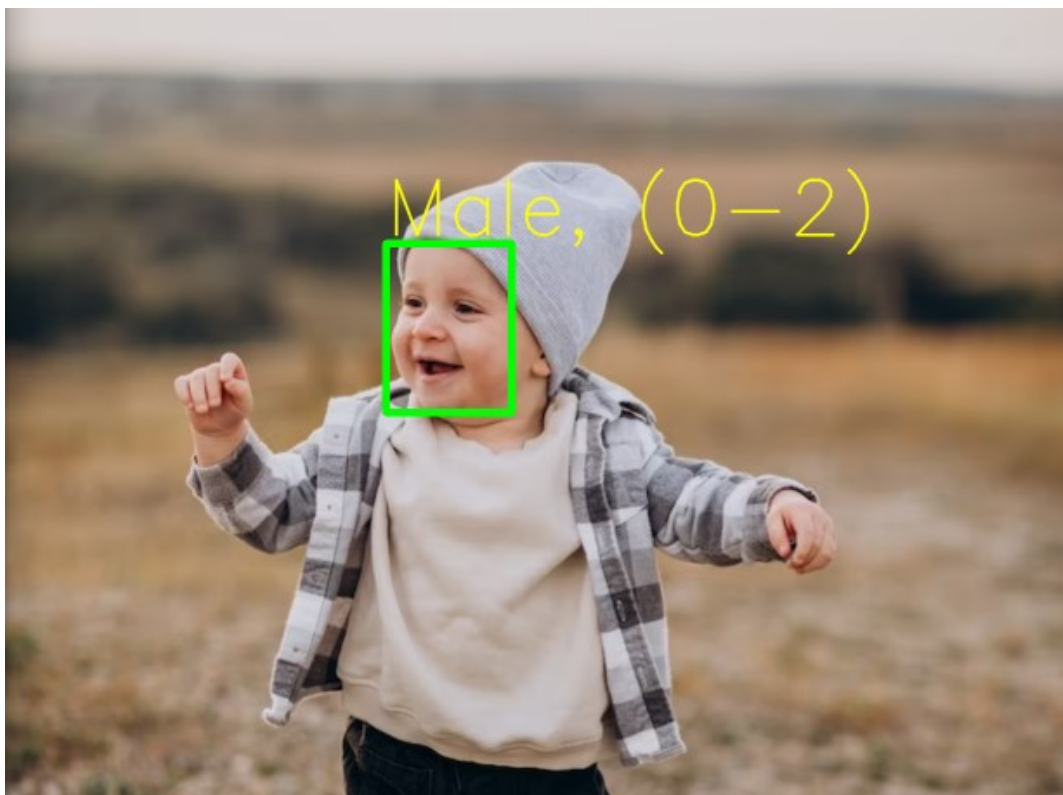
Slika 7. Konačni rezultati prikazani na autoru



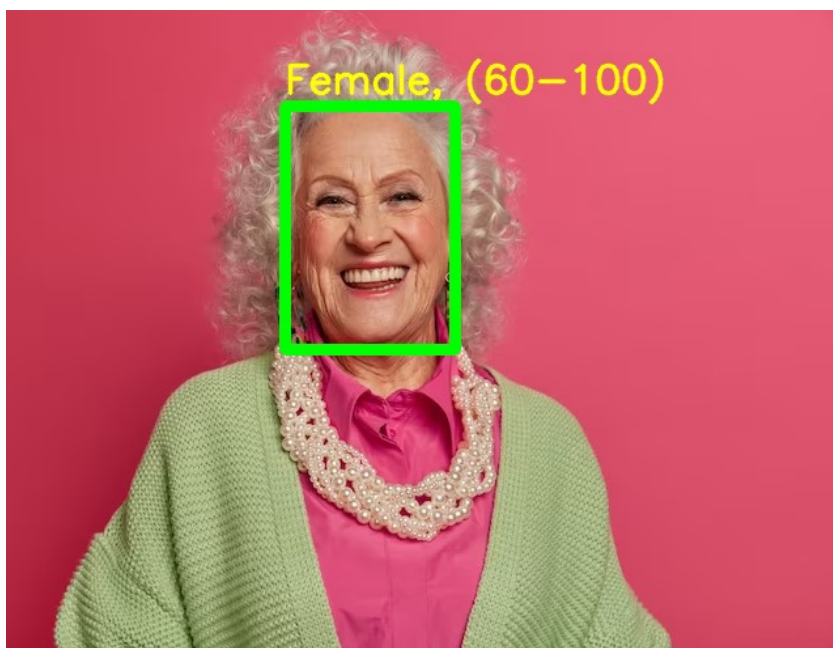
Slika 8. Rezultati prikazani na osobi mlađe dobi



**Slika 9. Rezultati prikazani na osobi srednje dobi**



**Slika 10. Rezultati prikazani na djetetu**



**Slika 11. Rezultati prikazani na osobi starije životne dobi**

Zahvaljujući dobro treniranim neuronskim mrežama, tehnologija detekcije lica i prepoznavanja spola i godina ostvarila je izvanredan napredak. Rezultati su zadovoljavajući jer dobro trenirane neuronske mreže za detekciju spola i dobi imaju za cilj pružiti što preciznije rezultate za sve spolove i dobne skupine kao što vidimo u priloženim slikama 7.-11. Ključni cilj takvih modela je postizanje visoke točnosti u prepoznavanju spola i dobi, bez obzira na varijacije u fizičkim karakteristikama pojedinaca.

## **5. PRIMJENA**

Točno prepoznavanje spola omogućuje ciljanje marketinških kampanja i proizvoda prema određenom spolu, povećavajući učinkovitost oglašavanja. Na primjer, proizvodi i usluge mogu se prilagoditi potrebama muških ili ženskih korisnika, što rezultira boljim korisničkim iskustvom i povećanom prodajom. Također precizna identifikacija dobi omogućuje personalizaciju sučelja i sadržaja na temelju dobne skupine korisnika. Ovo je ključno u sektorima kao što su obrazovanje i zabava, gdje se sadržaj može prilagoditi različitim dobnim grupama kako bi se osiguralo bolje angažiranje i razumijevanje. S pravim pristupom, možemo iskoristiti tehnologiju za unapređenje našeg svakodnevnog života i društva u cjelini.



## 6. KRITIČKI OSVRT

Kao što vidimo iz rezultata ovaj program je poput magičnog zrcala koje nam otkriva tajne o nečijem licu - točan spol i raspon godina. Bez ikakvog dodira ili razgovora, ovo računalno sa programom čita informacije s lica i pruža nam odgovore. Ali to nije samo zabava, to je korak naprijed u svijetu računalne vizije. Ova tehnologija otvara vrata bezbrojnim mogućnostima jer može poboljšati marketinške strategije prilagođavajući proizvode i usluge ciljanoj publici, osigurati sigurnost prepoznavanjem lica na ulazima i čak prilagoditi računalne sučelja prema dobi korisnika.

No, uz sve ove mogućnosti, postavlja se i pitanje privatnosti i etike. Tehnologija prepoznavanja lica ima potencijal za zloupotrebu i ugrožavanje privatnosti pojedinaca, stoga je važno balansirati napredak s odgovornošću i zaštitom osobnih podataka. Kritična je i transparentnost u načinu na koji se takvi alati koriste. Kako bi se osiguralo odgovorno korištenje, razgovor o etici i regulaciji tehnologije prepoznavanja lica postaje sve važniji.

## **7. ZAKLJUČAK**

Sve u svemu, ovaj program nije samo alat nego je i podsjetnik na čuda modernog računalnog svijeta i izazove koje donosi. Donosi nam bolje razumijevanje i komunikaciju s tehnologijom, ali također postavlja pitanja o našoj ulozi u tom sve složenijem odnosu između čovjeka i stroja. Duboko učenje omogućava modelima da otkriju obrasce i značajke u slikama koje bi bile teške ili gotovo nemoguće za ljudski um. U konačnici, ovaj program nam pokazuje da su granice tehnološkog napretka još uvijek daleko od dosega, a da istovremeno postavlja izazove i pitanja koja zajedno moramo razmotriti kako bismo osigurali da tehnologija služi dobrobiti svih nas.

## LITERATURA

- [1] Sama Group (training-data company), Computer Vision : History & How it Works  
<https://www.sama.com/blog/computer-vision-history-how-it-works/>
- [2] University of Waterloo, How machine learning helps Siri and Alexa understand you?  
<https://uwaterloo.ca/news/global-impact/machine-learning-helps-siri-and-alexa-understand-you>
- [3] University of Michigan, New computer vision technique enhances microscopy image analysis for improved cancer diagnosis  
<https://cse.engin.umich.edu/stories/new-computer-vision-technique-enhances-microscopy-image-analysis-for-improved-cancer-diagnosis>
- [4] United States National Library of Medicine, Computer Vision in the Surgical Operating Room  
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7768144/>
- [5] EPRS | European Parliamentary Research Service, Regulating facial recognition in the EU  
[https://www.europarl.europa.eu/RegData/etudes/IDAN/2021/698021/EPRS\\_IDA\(2021\)698021\\_EN.pdf](https://www.europarl.europa.eu/RegData/etudes/IDAN/2021/698021/EPRS_IDA(2021)698021_EN.pdf)
- [6] GDPR.eu, EU-What is GDPR?  
<https://gdpr.eu/what-is-gdpr>
- [7] California Stanford University, The Challenges of Facial Recognition Technologies  
<https://hai.stanford.edu/news/challenges-facial-recognition-technologies>
- [8] The Economist, Why San Francisco banned the use of facial-recognition technology?  
<https://www.economist.com/democracy-in-america/2019/05/16/why-san-francisco-banned-the-use-of-facial-recognition-technology>
- [9] Harvard Social Impact Review, Understanding Gender and Racial Bias in AI  
<https://www.sir.advancedleadership.harvard.edu/articles/understanding-gender-and-racial-bias-in-ai>
- [10] Python Software Foundation  
<https://www.python.org/about/>
- [11] Towards Data Science, Loss Functions and Their Use In Neural Networks by V. Yathish  
<https://towardsdatascience.com/loss-functions-and-their-use-in-neural-networks-a470e703f1e9>

- [12] BAIR - Berkeley Artificial Intelligence Research Lab, Caffe  
<https://bair.berkeley.edu/software>, <https://caffe.berkeleyvision.org/>
- [13] Towards Data Science, AlexNet: The Architecture that Challenged CNNs  
<https://medium.com/analytics-vidhya/concept-of-alexnet-convolutional-neural-network-6e73b4f9ee30>
- [14] Review: AlexNet, CaffeNet — Winner of 2012 ILSVRC (ImageNet Large Scale Visual Recognition Competition)  
<https://medium.com/coinmonks/paper-review-of-alexnet-caffenet-winner-in-ilsvrc-2012-image-classification-b93598314160>
- [15] Face Recognition - OpenCV  
<https://opencv.org/opencv-face-recognition/>
- [16] The Berkman Klein Center for Internet & Society at Harvard University  
<https://cyber.harvard.edu/story/2021-04/theres-no-such-thing-flawless-facial-recognition-technology>
- [17] Stipančić, T.: predavanja iz kolegija Umjetna inteligencija, Fakultet strojarstva i brodogradnje, Zagreb, 2022.
- [18] Majetić, D.: predavanja iz kolegija Neuronske mreže, Fakultet strojarstva i brodogradnje, Zagreb, 2022.
- [19] What is NumPy?  
<https://numpy.org/doc/stable/user/whatisnumpy.html>
- [20] Github (face-detection)  
<https://github.com/topics/face-detection>
- [21] Stackoverflow (face-recognition)  
<https://stackoverflow.com/questions/5412387/face-recognition>

## **PRILOZI**

- I. Programski kod u Python jeziku, ulazni parametar je video u realnom vremenu
- II. Programski kod u Python jeziku, ulazni parametar je video snimanja zaslona

I – Programski kod u Python jeziku, ulazni parametar je video u realnom vremenu

```
import cv2
import math
import argparse

def highlightFace(net, frame, conf_threshold=0.7):
    frameOpencvDnn=frame.copy()
    frameHeight=frameOpencvDnn.shape[0]
    frameWidth=frameOpencvDnn.shape[1]
    blob=cv2.dnn.blobFromImage(frameOpencvDnn, 1.0, (300, 300), [104, 117, 123],
    True, False)

    net.setInput(blob)
    detections=net.forward()
    faceBoxes=[]
    for i in range(detections.shape[2]):
        confidence=detections[0,0,i,2]
        if confidence>conf_threshold:
            x1=int(detections[0,0,i,3]*frameWidth)
            y1=int(detections[0,0,i,4]*frameHeight)
            x2=int(detections[0,0,i,5]*frameWidth)
            y2=int(detections[0,0,i,6]*frameHeight)
            faceBoxes.append([x1,y1,x2,y2])
            cv2.rectangle(frameOpencvDnn, (x1,y1), (x2,y2), (0,255,0),
            int(round(frameHeight/150)), 8)
    return frameOpencvDnn,faceBoxes

parser=argparse.ArgumentParser()
parser.add_argument('--image')

args=parser.parse_args()

faceProto="opencv_face_detector.pbtxt"
faceModel="opencv_face_detector_uint8.pb"
ageProto="age_deploy.prototxt"
ageModel="age_net.caffemodel"
genderProto="gender_deploy.prototxt"
genderModel="gender_net.caffemodel"

MODEL_MEAN_VALUES=(78.4263377603, 87.7689143744, 114.895847746)
ageList=['(0-2)', '(4-6)', '(8-12)', '(15-20)', '(20-25)', '(38-43)', '(48-53)',
'(60-100)']
genderList=['Male', 'Female']
```

```
faceNet=cv2.dnn.readNet(faceModel,faceProto)
ageNet=cv2.dnn.readNet(ageModel,ageProto)
genderNet=cv2.dnn.readNet(genderModel,genderProto)

video=cv2.VideoCapture(args.image if args.image else 0)
padding=20
while cv2.waitKey(1)<0:
    hasFrame,frame=video.read()
    if not hasFrame:
        cv2.waitKey()
        break

    resultImg,faceBoxes=highlightFace(faceNet,frame)
    if not faceBoxes:
        print("No face detected")

    for faceBox in faceBoxes:
        face=frame[max(0,faceBox[1]-padding):
                    min(faceBox[3]+padding,frame.shape[0]-1),max(0,faceBox[0]-
padding)
                    :min(faceBox[2]+padding, frame.shape[1]-1)]

        blob=cv2.dnn.blobFromImage(face, 1.0, (227,227), MODEL_MEAN_VALUES,
swapRB=False)
        genderNet.setInput(blob)
        genderPreds=genderNet.forward()
        gender=genderList[genderPreds[0].argmax()]
        print(f'Gender:{gender}')

        ageNet.setInput(blob)
        agePreds=ageNet.forward()
        age=ageList[agePreds[0].argmax()]
        print(f'Age: {age[1:-1]} years')

    cv2.putText(resultImg, f'{gender}, {age}', (faceBox[0], faceBox[1]-10),
cv2.FONT_HERSHEY_SIMPLEX, 3.0, (0,255,255), 2, cv2.LINE_AA)

    resultImg = cv2.resize(resultImg, None, fx=0.5, fy=0.5,
interpolation=cv2.INTER_AREA) #Dodani_Resize

    cv2.imshow("Detecting age and gender", resultImg)
```

II – Programski kod u Python jeziku, ulazni parametar je video snimanja zaslona

```
import cv2
import numpy as np
import pyautogui
import time
import argparse

def highlightFace(net, frame, conf_threshold=0.7):
    frameOpencvDnn = frame.copy()
    frameHeight = frameOpencvDnn.shape[0]
    frameWidth = frameOpencvDnn.shape[1]
    blob = cv2.dnn.blobFromImage(frameOpencvDnn, 1.0, (300, 300), [104, 117,
123], True, False)

    net.setInput(blob)
    detections = net.forward()
    faceBoxes = []
    for i in range(detections.shape[2]):
        confidence = detections[0, 0, i, 2]
        if confidence > conf_threshold:
            x1 = int(detections[0, 0, i, 3] * frameWidth)
            y1 = int(detections[0, 0, i, 4] * frameHeight)
            x2 = int(detections[0, 0, i, 5] * frameWidth)
            y2 = int(detections[0, 0, i, 6] * frameHeight)
            faceBoxes.append([x1, y1, x2, y2])
            cv2.rectangle(frameOpencvDnn, (x1, y1), (x2, y2), (0, 255, 0),
int(round(frameHeight / 150)), 8)
    return frameOpencvDnn, faceBoxes

parser = argparse.ArgumentParser()
parser.add_argument('--image')

args = parser.parse_args()

faceProto = "opencv_face_detector.pbtxt"
faceModel = "opencv_face_detector_uint8.pb"
ageProto = "age_deploy.prototxt"
ageModel = "age_net.caffemodel"
genderProto = "gender_deploy.prototxt"
genderModel = "gender_net.caffemodel"

MODEL_MEAN_VALUES = (78.4263377603, 87.7689143744, 114.895847746)
ageList = ['(0-2)', '(4-6)', '(8-12)', '(15-20)', '(20-25)', '(38-43)', '(48-
53)', '(60-100)']
genderList = ['Male', 'Female']

faceNet = cv2.dnn.readNet(faceModel, faceProto)
ageNet = cv2.dnn.readNet(ageModel, ageProto)
genderNet = cv2.dnn.readNet(genderModel, genderProto)
```



```
while cv2.waitKey(1) < 0:
    screenshot = pyautogui.screenshot() # Snimi trenutni ekran
    frame = np.array(screenshot) # Pretvori screenshot u NumPy sliku
    frame = cv2.cvtColor(frame, cv2.COLOR_RGB2BGR) # Konvertiraj u BGR format

    resultImg, faceBoxes = highlightFace(faceNet, frame)
    if not faceBoxes:
        print("No face detected")

    for faceBox in faceBoxes:
        face = frame[max(0, faceBox[1]-20):
                    min(faceBox[3]+20, frame.shape[0]-1), max(0, faceBox[0]-20):
                    min(faceBox[2]+20, frame.shape[1]-1)]

        blob = cv2.dnn.blobFromImage(face, 1.0, (227, 227), MODEL_MEAN_VALUES,
swapRB=False)
        genderNet.setInput(blob)
        genderPreds = genderNet.forward()
        gender = genderList[genderPreds[0].argmax()]
        print(f'Gender: {gender}')

        ageNet.setInput(blob)
        agePreds = ageNet.forward()
        age = ageList[agePreds[0].argmax()]
        print(f'Age: {age[1:-1]} years')

        cv2.putText(resultImg, f'{gender}, {age}', (faceBox[0], faceBox[1]-10),
cv2.FONT_HERSHEY_SIMPLEX, 1.0, (0, 255, 255), 2, cv2.LINE_AA)

    resultImg = cv2.resize(resultImg, None, fx=1, fy=1,
interpolation=cv2.INTER_AREA)

    cv2.imshow("Detecting age and gender", resultImg)
    time.sleep(0.1) # Pauza kako bi se smanjilo opterećenje CPU-a
```