

Prepoznavanje gesta u ozbiljnim igrama

Dragošević, Marija

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:168:583354>

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

Download date / Datum preuzimanja: **2025-03-14**



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 521

PREPOZNAVANJE GESTA U OZBILJNIM IGRAMA

Marija Dragošević

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 521

PREPOZNAVANJE GESTA U OZBILJNIM IGRAMA

Marija Dragošević

Zagreb, lipanj 2024.

DIPLOMSKI ZADATAK br. 521

Pristupnica: **Marija Dragošević (0036514778)**

Studij: Računarstvo

Profil: Računarska znanost

Mentor: izv. prof. dr. sc. Jurica Babić

Zadatak: **Prepoznavanje gesta u ozbiljnim igrama**

Opis zadatka:

Prepoznavanje gesta u ozbiljnim igrama omogućuje imerzivno iskustvo interakcije čovjeka i računala čime se može unaprijediti korisničko iskustvo igrača, posebice kroz intuitivne mehanizme upravljanja igrom temeljene na prirodnim tjelesnim pokretima. Vaš je zadatak ostvariti rješenje za prepoznavanje gesti u ozbiljnim igrama. U okviru navedenog zadatka potrebno je istražiti postojeća rješenja za prepoznavanje gesti te njihovu uporabu u ozbiljnim igrama. Na temelju analize predložiti model rješenja koji se sastoji od minimalno dva dijela: (i) modula za prepoznavanje gesti na temelju video ulaza iz kamere uređaja; te (ii) modula za upravljanje igrom na temelju prepoznatih gesti. Modul za prepoznavanje gesti mora se oslanjati na prijenosno učenje pomoću kojeg je moguće prepoznati minimalno tri nove geste u odnosu na polazišni model nad kojim je obavljeno prijenosno učenje.

Rok za predaju rada: 28. lipnja 2024.

Hvala svima koji su vjerovali u mene na ovom putu, Bogu, obitelji i prijateljima, a posebno svojim bakama Milicama kojima ovaj rad i posvećujem. Isto tako, zahvaljujem svom mentoru izv. prof. dr. sc. Jurici Babiću i asistentici Ani Radović, mag. ing. koji su mi pomogli pri izradi diplomskog rada svojim savjetima i konstruktivnim kritikama, pružajući nesebičnu pomoć u svakom trenutku kada je to bilo potrebno.

Sadržaj

1. Uvod	3
2. Teorijska podloga	4
2.1. Interakcija čovjeka i računala	4
2.2. Prepoznavanje gesta	5
2.2.1. OpenPose	5
2.2.2. YOLOv8 Pose	6
2.2.3. MediaPipe Solutions	6
2.3. Ozbiljne igre	7
3. Prepoznavanje gesta koristeći MediaPipe Solutions	8
3.1. Model za prepoznavanje karakterističnih točaka ruke	9
3.2. Model za prepoznavanje gesta	10
4. Skup podataka	11
4.1. Metodologija	11
4.2. Opća uredba o zaštiti podataka (GDPR)	12
4.3. Specifikacija gesta	12
4.3.1. Postojeće geste	12
4.3.2. Nove geste	13
4.4. Aplikacija GestureCapture	14
4.5. Prikupljanje podataka	16
4.6. Čišćenje, validacija i analiza podataka	16
5. Prilagođeni model za prepoznavanje gesta	17
5.1. MediaPipe Model Maker	17

5.2. Klasifikacija	18
5.2.1. Učitavanje podataka	18
5.2.2. Treniranje	19
5.2.3. Rezultati	19
6. Studijski slučaj korištenja modela u ozbiljnoj igri	23
6.1. Ozbiljna igra "Pogodi gestu!"	23
6.1.1. Opis	23
6.1.2. Arhitektura	25
6.1.3. Svrha	26
6.1.4. Evaluacija	27
7. Zaključak	28
Popis slika	29
Popis tablica	30
Literatura	31
Sažetak	36
Abstract	37
A: Privola	38
B: Upute	40

1. Uvod

Prepoznavanje gesta predstavlja ključnu komponentu u razvoju intuitivnih interakcija između čovjeka i računala. U posljednjim godinama, značajan napredak je ostvaren u ovom području zahvaljujući razvoju naprednih tehnologija računalnog vida i strojnog učenja. U ovom radu istražuje se primjena tehnologija za prepoznavanje gesta u ozbiljnim igrama, s ciljem poboljšanja korisničkog iskustva kroz prirodne metode interakcije. Predstavljanjem različitih biblioteka i alata, poput OpenPose, YOLOv8 Pose i korištenjem MediaPipe Solutions, razvit će se sustav za prepoznavanje gesta koji može integritati geste kao način upravljanja igrom. Ozbiljne igre, koje se koriste u edukativne, rehabilitacijske i simulacijske svrhe, pružaju idealno okruženje za primjenu ove tehnologije, omogućujući korisnicima da se na interaktivan i zabavan način angažiraju. Ova tehnologija može poboljšati pristupačnost igara za osobe s različitim fizičkim sposobnostima i pružiti nove mogućnosti za edukaciju i terapiju.

U prvom dijelu rada, predstaviti će se teorijska podloga koja uključuje pregled tehnologija za prepoznavanje gesta. Opisati će se pojam ozbiljnih igara i njihova važnost u različitim industrijama. U drugom dijelu, detaljno će se opisati razvoj prilagođenog modela za prepoznavanje gesta, uključujući prikupljanje podataka, treniranje modela i evaluaciju performansi. Na kraju, prikazati će se konkretan primjer primjene razvijenog modela u ozbiljnoj igri, čime će se demonstrirati praktična upotreba ove tehnologije. Cilj ovog rada je doprinijeti boljem razumijevanju i primjeni tehnologije prepoznavanja gesta u kontekstu ozbiljnih igara, te pružiti smjernice za buduća istraživanja i razvoj u ovom području.

2. Teorijska podloga

Ovo poglavlje pružit će razumijevanje ključnih koncepata i tehnologija koje se koriste u radu. Predstavlja pregled područja interakcije čovjeka i računala (engl. *Human-Computer Interaction, HCI*), s naglaskom na računalni vid i prepoznavanje gesta. Zatim će se predstaviti nekoliko ključnih biblioteka i alata za prepoznavanje gesta, OpenPose, YOLOv8 Pose i MediaPipe Solutions. Na kraju će se predstaviti koncept ozbiljnih igara.

2.1. Interakcija čovjeka i računala

Interakcija čovjek-računalo (engl. *Human-Computer Interaction, HCI*) je multidisciplinarno područje koje proučava interakciju čovjeka (i.e. korisnika) i računala [1]. Ovo široko područje nastoji istražiti i poboljšati načine na koje ljudi komuniciraju s tehnologijom kroz dizajn računalnih sustava i sučelja koja su intuitivna i ugodna za korištenje. Fokus je na prilagođavanju tehnologije fizičkim i psihološkim potrebama korisnika kako bi se omogućila što veća učinkovitost i zadovoljstvo tijekom korištenja[2].

Računalni vid (engl. *Computer Vision*) područje je umjetne inteligencije u kojem se koriste tehnike strojnog učenja kako bi računala imala sposobnost izvući i interpretirati informacije iz slika i videa na način na koji to čine ljudi [3]. Koristi se za razne svrhe kao što je prepoznavanje objekata i lica, analiza slika, prepoznavanje aktivnosti u stvarnom vremenu i mnoge druge.

Integracija računalnog vida u interakciju čovjeka i računala predstavlja napredan i inovativan način korištenja tehnologije koje korisnicima omogućava prirodnije i intuitivnije iskustvo. Korisnici ne moraju učiti složene kontrole i sučelja već se mogu osloniti na način komuniciranja s tehnologijom koji im je prirodni, prepoznavanjem gesta ili

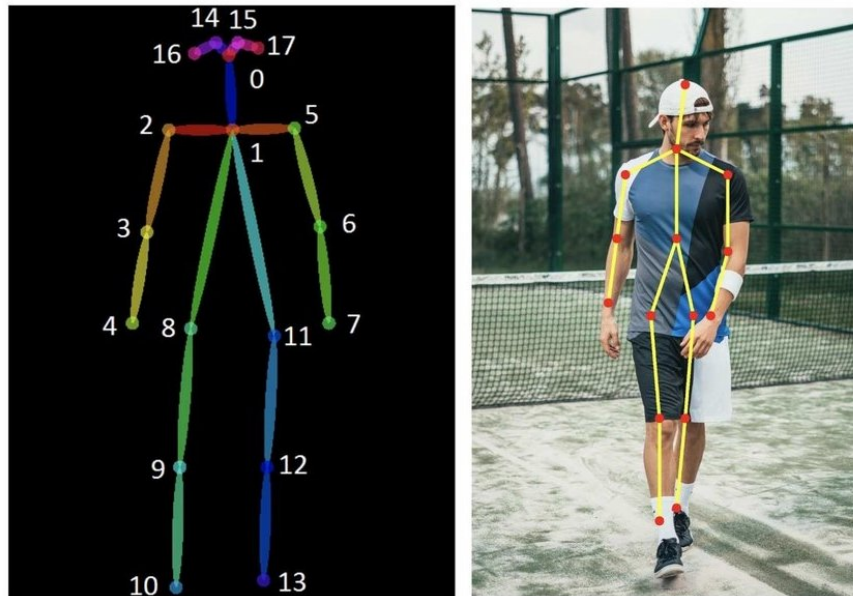
praćenjem pokreta. Međutim, postoje brojni izazovi koje treba prevladati kako bi se osigurala učinkovita implementacija. Prepoznavanje mora biti točno i pouzdano, sustav se mora moći prilagoditi različitim uvjetima i korisnicima, obrada podataka mora biti brza i pritom se moraju poštivati stroge politike privatnosti i sigurnosti. Zato je rješavanje ovih izazova ključno za daljnji razvoj i primjenu računalnog vida u interakciji čovjeka i računala [4].

2.2. Prepoznavanje gesta

Prepoznavanje gesta jedno je od glavnih problema kojim se bavi računalni vid. U širem smislu, to je tehnologija koja pokušava interpretirati ljudske geste kao što su pokreti ruke, izrazi lica ili govor tijela [5]. Fokus u ovom radu bit će na prepoznavanju položaja i pokreta ruke. Na temelju prepoznatih gesta, računala pokreću određene akcije što omogućava razne primjene. Neke od njih su korištenje sučelja bez dodira, stvaranje interaktivnih igara, korištenje u rehabilitaciji i asistivnim tehnologijama kao što je npr. potpomognuta komunikacija [6]. Prepoznavanje gesta najviše se oslanja na konvolucijske neuronske mreže (engl. *Convolutional Neural Networks, CNNs*) koje su pokazale sposobnost da učinkovito analiziraju vizualne podatke i daju točne i pouzdane rezultate [7]. U nastavku će biti dani neki od najpoznatijih biblioteka koji uključuju modele za prepoznavanje gesta.

2.2.1. OpenPose

OpenPose je biblioteka za prepoznavanje poze tijela više osoba u stvarnom vremenu. Može detektirati 135 karakterističnih točaka ljudskog tijela, stopala, ruku i lica na pojedinačnim slikama kao što se može vidjeti na slici 2.1. Ova biblioteka je popularna upravo zbog svoje kvalitete i robusnosti u okruženjima s više osoba. Modeli za prepoznavanje poze su veliki i zahtjevaju značajne računalne resurse što ih čini neprikladnima za video analizu u stvarnom vremenu [8]. Kako bi se poboljšale performanse, smanjili računalni resursi i omogućilo prepoznavanje u stvarnom vremenu predstavljen je Lightweight OpenPose algoritam koji učinkovito radi na CPU uz minimalni gubitak točnosti [9].



Slika 2.1. Prepoznavanje poze tijela koristeći OpenPose [8].

2.2.2. YOLOv8 Pose

YOLOv8 (*You Only Look Once version 8*) je najnovija iteracija popularnog YOLO modela za prepoznavanje objekata. Svi YOLO modeli se temelje na algoritmu jednog prolaska kroz sliku (*engl. Single-Shot Detector, SSD*) i koriste konvolucijsku neuronsku mrežu (CNN). Prepoznavanje objekata jednim prolazom koristi jedan prolaz ulazne slike kako bi se napravile predikcije o prisutnosti i lokaciji objekata na slici. Ova metoda obrađuje cijelu sliku u jednom prolazu, što je čini računalno učinkovitom za rad u stvarnom vremenu [10]. YOLOv8 Pose model je specijalizirana verzija koja se fokusira na prepoznavanje poze i karakterističnih točaka tijela, uključujući zglobove, laktove, koljena, gležnjeve, kao i ključne točke na rukama što čini ovaj model prikladnim za prepoznavanje gesta [11].

2.2.3. MediaPipe Solutions

MediaPipe Solutions je platforma otvorenog koda (*engl. open-source framework*) koju je Google predstavio 2023. godine [12]. Razvijena je na temelju MediaPipe platforme čija je prva javna verzija dostupna od 2019. MediaPipe Solutions sadržava skup raznih biblioteka i alata kojima programerima olakšava i ubrzava ugradnju umjetne inteligencije i strojnog učenja u njihove aplikacije. Sadržava predtrenirane i spremne za pokretanje modele za prepoznavanje i praćenje objekata, prepoznavanje i analizu lica, prepoznavanje

nje gesta, analizu pokreta, praćenja tijela, obradu zvuka i još mnogo njih.

Modeli se mogu koristiti na raznim platformama i operacijskim sustavima (e.g. Python, Web, Android, iOS) kao i prilagoditi vlastitim potrebama. Jedna od glavnih prednosti je ta što su modeli i alati u MediaPipe Solutions optimizirani za rad u stvarnom vremenu. To znači da aplikacije mogu obrađivati i analizirati videozapise ili zvučne zapise u stvarnom vremenu s visokom brzinom i efikasnošću [13]. Rješenje koje nudi MediaPipe za prepoznavanje gesta bit će detaljnije obrađeno u poglavlju 3., gdje će se dati detaljna specifikacija modela.

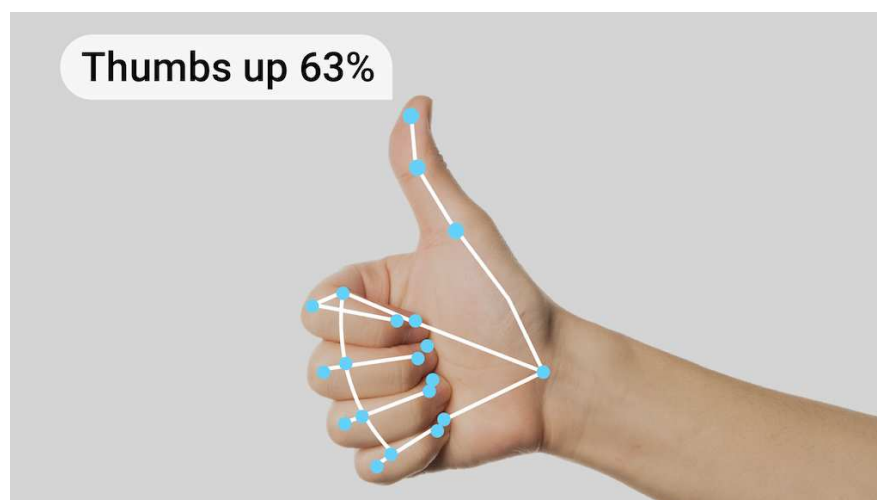
2.3. Ozbiljne igre

Ozbiljne igre su vrste video igara koje nisu dizajnirane prvenstveno za zabavu, već imaju obrazovne, terapijske, istraživačke ili neke druge ozbiljne ciljeve [14]. Među njima može se istaknuti inovativni projekt Play2Green u okviru programa ERASMUS+. Fokusira se na integraciju ozbiljnih igara temeljenih na umjetnoj inteligenciji i proširenoj stvarnosti (engl. *Augmented Reality, AR*) u obrazovanje čiji je cilj podizanje svijesti o okolišu i borbi protiv klimatskih promjena [15]. Još neki od najpoznatijih primjera ozbiljnih igara su: *Microsoft Flight Simulator*, komercijalni simulator letenja, koji se smatra početkom ozbiljnih igara, *DuoLingo*, edukativna igra za učenje stranih jezika i *Little Learning Machines*, igra o podržanom učenju (engl. *reinforcement learning*) u kojoj igrači treniraju svoje male strojeve koristeći različite nagrade te na taj način uče o umjetnoj inteligenciji kako napreduju kroz igru [16].

Kako bi se igračko iskustvo poboljšalo, umjesto korištenja tipkovnice, miša ili upravljača može se koristiti sustav za prepoznavanje gesta koje kada se gesta prepozna, okida neku akciju u igri. Prepoznavanje gesta omogućava igračima da koriste prirodne pokrete ruku za kontrolu igre što omogućava bržu, dinamičnu i direktnu interakciju [17]. Napredak u računalnom vidu i prepoznavanju gesta čini ovu tehnologiju sve više dostupnom i primjenjivom u raznim edukacijskim, rehabilitacijskim i simulacijskim igrama [18]. U poglavlju 6. prikazat će se integracija tehnologije prepoznavanja gesta u jednostavnoj edukativnoj ozbiljnoj igri.

3. Prepoznavanje gesta koristeći MediaPipe Solutions

Gesture Recognizer je rješenje koje Google nudi za prepoznavanje gesta u stvarnom vremenu u sklopu svoje MediaPipe Solutions platforme [19]. Ovo rješenje može se koristiti za prepoznavanje specifičnih gesta ruku korisnika i aktiviranje značajki aplikacije koje odgovaraju tim gestama. Već istrenirani model za prepoznavanje prilagođen je za korištenje u Python, Android, iOS i web aplikacijama te prihvaća statične podatke (i.e. slike) ili kontinuirani tok (i.e. isječke iz videa). Nudi mogućnost prilagodbe modela i treniranje na vlastitom skupu podataka što će biti detaljno predstavljeno u poglavlju 5. Temelji se sustavu Hand Gesture Classification koji se sastoji od dva modela, za prepoznavanje karakterističnih točaka ruke Hand Landmarker i klasifikaciju gesta Hand Gesture Classifier čija će specifikacija biti dana u nastavku. Rezultat prepoznavanja koji kombinacija ta dva modela daje može se vidjeti na slici 3.1.

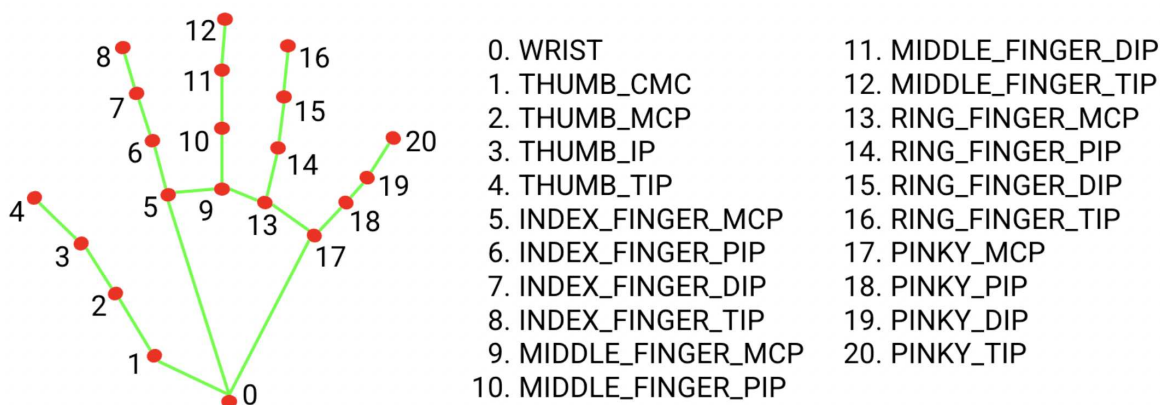


Slika 3.1. Prepoznavanje geste palac gore, s točnošću 63% koristeći Gesture Recognizer [19].

3.1. Model za prepoznavanje karakterističnih točaka ruke

Prepoznavanje karakterističnih točaka ruke odvija se pomoću paketa modela Hand Landmarker u kojem su ugrađena 2 modela: model za prepoznavanje ruke (i.e. dlan i prsti) koji locira ruku na ulaznoj slici i model za prepoznavanje karakterističnih točaka na ruci koji ih prepoznaje na izrezanoj slici ruke koju smo prethodno locirali [20]. Detaljna specifikacija svakog modela dana je u tablici 3.1.

Glavni rezultat prepoznavanja je 21 koordinata (x, y, z) zglobova ruke kao što se može vidjeti na slici 3.2. Hand Landmarker je treniran na oko 30 tisuća stvarnih slika i nekoliko renderiranih modela ruku na različitim pozadinama [20].



Slika 3.2. Karakteristične točke ruke na izlazu iz Hand Landmarker paketa modela [20].

Naziv	Tip	Arhitektura	Ulaz	Izlaz
Detector model	Konvolucijska neuronska mreža	SSD	192 x 192 x 3 tenzor ¹	2016 x 18 tenzor ²
Tracker model	Konvolucijska neuronska mreža	Regresijski model	224 x 224 x 3 tenzor ⁴	realni skalar ³ 1 x 63 tenzor ⁵ realni skalar ⁶ 1 x 63 tenzor ⁷

Tablica 3.1. Specifikacija Hand Landmarker paketa modela [21].

¹Tensor dimenzije 192 x 192 x 3 koji predstavlja jedan okvir videa ili sliku.

²Realni tenzor dimenzije 2016 x 18 koji predstavlja referentne okvire (engl. *Anchor Boxes*).

³Realni skalar koji predstavlja vjerojatnost postojanja ruke na ulaznoj slici.

⁴Tensor dimenzije 224 x 224 x 3 koji predstavlja dio početne slike u kojoj je detektor pronašao ruku.

⁵Tensor koji predstavlja 21 (x, y, z) karakterističnu točku normaliziranu veličinom slike.

⁶Realni skalar koji predstavlja vjerojatnost da se na slici nalazi desna ruka.

⁷Tensor koji predstavlja 21 (x, y, z) karakterističnu točku na metričkoj skali u stvarnim dimenzijama.

3.2. Model za prepoznavanje gesta

Prepoznavanje gesta odvija se pomoću *Hand Gesture Classification* paketa modela koji isto sadrži 2 modela. *Embedding model* služi za ekstrakciju značajki iz 21 karakteristične točke ruke koje onda klasificira *Classification model*. Ulaz Embedding modela nisu slike (i.e. RGB podaci o pikselima), već su izlazi iz Hand Landmarker paketa modela. Izlaz klasifikacijskog modela je 8-dimenzionalni vektor koji predviđa vjerojatnost za svaku pojedinu klasu. Element na poziciji 0 predstavlja vjerojatnost da gesta nije poznata modelu, a elementi na pozicijama 1 do 8 predstavljaju vjerojatnosti za neku od 7 gesti koje su objašnjene u potpoglavlju 4.3.1. Detaljna specifikacija dana *Hand Gesture Classification* paketa modela dana je u tablici 3.2.

U specifikaciji modela navedene su primjene koje su izvan njegovog opsega. Nije prikladno koristiti ga za prepoznavanje gesta koje uključuju dvije ruke, za dinamičke geste, prevođenje znakovnog jezika kao i za bilo koji oblik nadzora ili prepoznavanja identiteta. U nastavku ovog rada ispitat će se je li ipak moguće koristiti model za geste s dvije ruke i kakve rezultate pri tome daje.

Naziv	Tip	Arhitektura	Ulaz	Izlaz
Embedding model	Potpuno povezana neuronska mreža s rezidualnim blokovima	Regresijski model	1 x 63 tenzor ⁵ realni skalar ⁶ 1 x 63 tenzor ⁷	128 x 1 tenzor ⁸
Classification model	Potpuno povezana neuronska mreža	Klasifikacijski model	128 x 1 tenzor ⁸	8-dimenzionalni vektor ⁸

Tablica 3.2. Specifikacija Hand Gesture Classification modela [22]

⁵Tenzor koji predstavlja 21 (x, y, z) karakterističnu točku normaliziranu veličinom slike.

⁶Realni tenzor dimenzije predviđenih značajki koji predstavlja karakteristične točke ruke.

⁷Realni skalar koji predstavlja vjerojatnost da se na slici nalazi desna ruka.

⁸Tenzor koji predstavlja 21 (x, y, z) karakterističnu točku na metričkoj skali u stvarnim dimenzijama.

⁸Realni tenzor predviđenih značajki koji predstavlja karakteristične točke ruke.

⁸Vektor s 8 elemenata koji predstavlja klase za 8 gesta.

4. Skup podataka

U ovom poglavlju opisat će se postupak prikupljanja skupa podataka. Obuhvaćat će specifikaciju gesta i opis programskog rješenja GestureCapture, pregled pravnih okvira i etičkih smjernica koji su se poštovali kako bi se osigurala prava volontera čije će slike biti naknadno korištene za treniranje modela. Na kraju će biti predstavljen pregled konačnog skupa podataka uključujući njegovu analizu.

4.1. Metodologija

Skup podataka (engl. *dataset*) je strukturirana kolekcija podataka koja je pohranjena u formatu koji se koristi za treniranje modela strojnog učenja [23]. Za potrebe ovog rada, osmišljen je skup podataka HANDS (*Hand Actions for Navigation and Direction in Serious games*) po uzoru na HaGRID (*HAnd Gesture Recognition Image Dataset*). On sadržava 552,992 slika veličine 1080p i namijenjen je za izgradnju sustava za prepoznavanje gesti ruke. [24]. U nastavku ovog rada koristit će se naziv "HANDS" kao skraćenicu za skup podataka za prepoznavanje gesti ruke koji će biti detaljno objašnjen u poglavlju 4.3.

Opći postupak izrade skupa podataka je definiranje, prikupljanje, čišćenje, transformacija i validacija podataka [25]. Kako bi se stvorio pouzdan, sveobuhvatan i raznovrstan skup podataka HANDS, praćen je opći postupak izrade koji je prilagođen specifičnim zahtjevima:

1. Definiranje gesti i opsega skupa podataka - detaljan pregled ovog koraka opisan je u poglavlju 4.3.
2. Razvoj aplikacije za snimanje i označavanje slika - GestureCapture.py je programsko rješenje koje je omogućilo jednostavno prikupljanje podataka od volontera koji

su pristali sudjelovati u izradi HANDS-a. Detaljan opis i arhitektura dani su u poglavlju 4.4.

3. Pravni okvir i sigurnost podataka - Pregled će biti dan u poglavlju 4.2.
4. Prikupljanje podataka
5. Čišćenje podataka
6. Validacija i Analiza

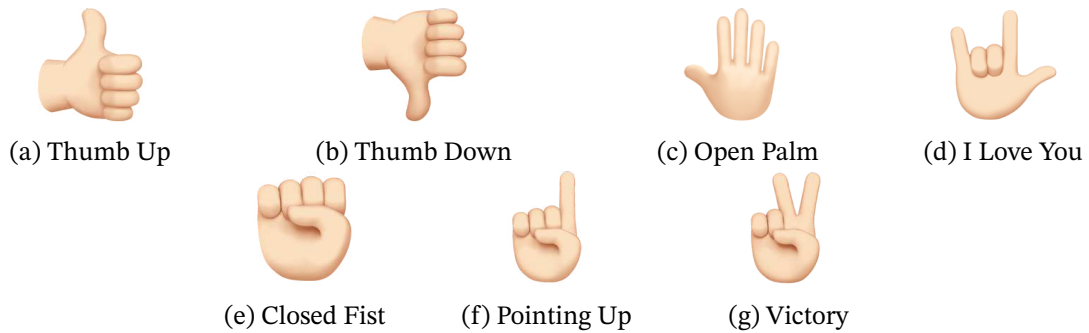
4.2. Opća uredba o zaštiti podataka (GDPR)

Kako HANDS sadrži slike ljudi, bilo je važno poštovati niz pravnih okvira i etičkih smjernica kako bi se osigurala zaštita privatnosti i prava osoba koje sudjeluju u prikupljanju podataka. S obzirom da je Hrvatska dio Europske unije te da su svi sudionici njezini građani, bilo je potrebno poštovati Opću uredbu o zaštiti podataka (GDPR) koja propisuje stroge zahtjeve za prikupljanje, obradu i pohranu osobnih podataka [26]. Oni uključuju privolu, transparentnost, pravo na pristup i brisanje te sigurnost podataka. Prva tri zahtjeva ostvarena su na način da su sudionici morali potpisati obrazac u kojem su bili obaviješteni o svrsi prikupljanja podataka, načinu njihove obrade i pravu na povlačenje privole koja za posljedicu ima brisanje svih podataka. Isto tako, sudionici u svakom trenutku trajanja prikupljanja i treniranja mogu tražiti pristup tim podacima. Obrazac se može pročitati u prilogu A Sigurnost podataka ostvarena je na način da su svi podaci pohranjeni lokalno i na privatnu repozitorij na Google Drive platformi, anonimizirani te će nakon treniranja modela biti obrisani. Osim što su se na ovaj način zadovoljili pravni i etički okviri osigurano je povjerenje i integritet prikupljanja podataka.

4.3. Specifikacija gesta

4.3.1. Postojeće geste

Hand Gesture Classification model prepoznaje 7 gesta koje su prikazane na slici 4.1. Model sadržava 8 klasa jer sve geste koje ne prepozna, klasificira kao klasu None.



Slika 4.1. Geste koje prepoznaje Hand Gesture Classification model [27]

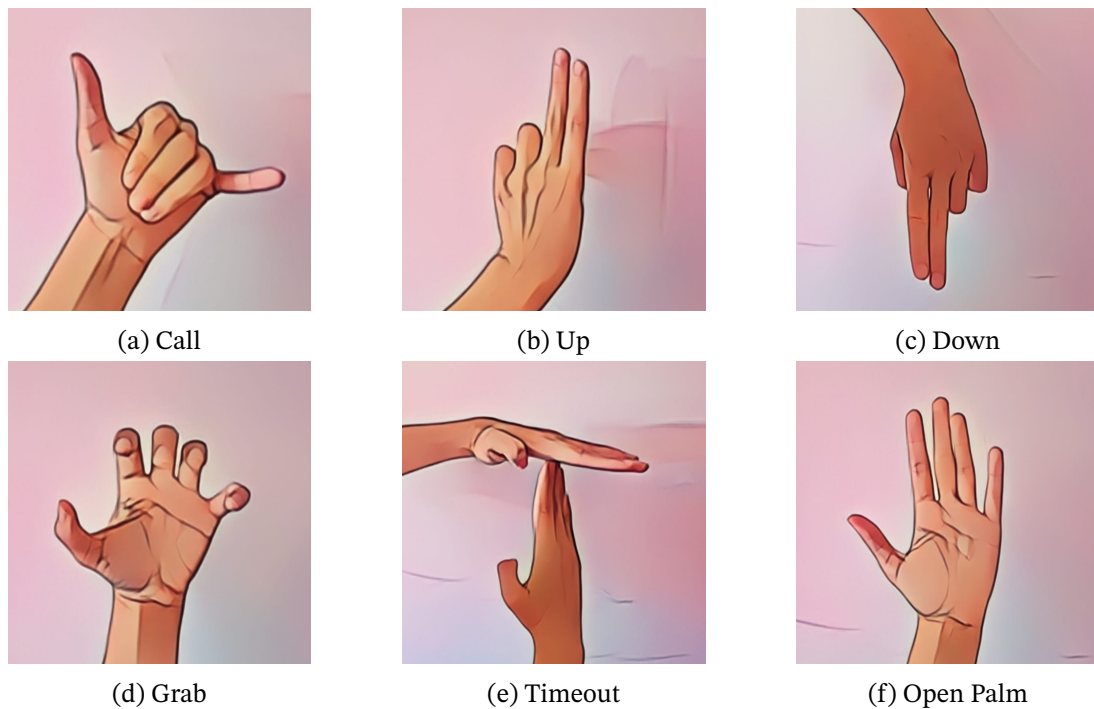
4.3.2. Nove geste

Cilj ovog rada je ostvariti prepoznavanje minimalno tri nove geste u odnosu na polazišni model. Odlučeno je da prilagođeni (i.e. treniran na HANDS skupu podataka) Hand Gesture Classification model prepoznaje 5 novih gesta i 1 već postojeću Open Palm. Sadrži 7 klasa, jer osim 6 za geste, mora sadržavati i None za one geste koje ne uspije prepoznati. Tablica 4.1. detaljno prikazuje za svaku gestu oznaku klase, opis i referencu na sliku koja je dio prikaza svih gesti na slici 4.2.

Prilikom odabira gesta, vodilo se računa da one budu jednostavne, intuitivne i da se mogu primijeniti na ozbiljnu igru. Isto tako, bilo je važno da tranzicija između njih bude fluidna, odnosno da korisnik lako može prebaciti jednu gestu u drugu.

Klasa	Opis	Slika
Call	Ispruženi palac i mali prst, simulira držanje telefonske slušalice između uha i usta	(a)
Up	Ispruženi i spojeni kažiprst i srednji prst koji pokazuju prema gore	(b)
Down	Ispruženi i spojeni kažiprst i srednji prst koji pokazuju prema dole	(c)
Grab	Poluzatvaranje ruke u šaku, kao da hvatate ili držite objekt	(d)
Timeout	Dlanovi okomito postavljeni jedan na drugi stvarajući slovo T	(e)
Open Palm	Ispruženi dlan sa skupljenim ili blago raširenim prstima	(f)
None	Sve ostale geste koje nisu opisane u tablici	/

Tablica 4.1. Specifikacija novih gesta

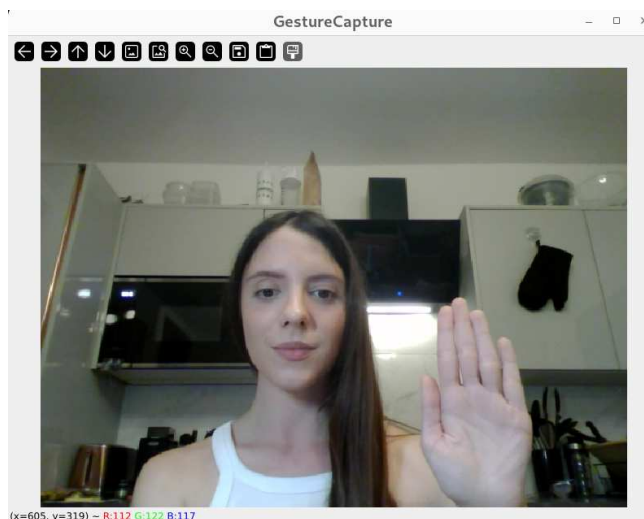


Slika 4.2. Nove geste za prilagođeni model.

4.4. Aplikacija GestureCapture

Programsko rješenje prikupljanja i označavanja skupa podataka je Python skripta GestureCapture koja koristi biblioteku OpenCV [28] za snimanje 6 različitih gesti ruke putem kamere ugrađene u računalo na kojem se pokreće. Omogućava korisniku, volonteru koji se prijavio za sudjelovanje u prikupljanju skupa podataka, da snima slike različitih gesta koje će onda biti spremljene i organizirane u odgovarajuće direktorije s nazivom oznake klase.

Kada se skripta pokrene, otvara se video prijenos. Ako kamera nije dostupna, na konzoli se ispisuje poruka o grešci i program se prekida. Ako je kamera dostupna, na konzoli se isto tako ispisuju upute na koji način koristiti aplikaciju. Korištenje je maksimalno pojednostavljeno na način da se korisnik kreće kroz aplikaciju pomoću tipki s tipkovnice, čijim klikom okida sliku koja se onda sprema u direktorij ovisno o nazivu geste za pritisnutu tipku. Npr. kada korisnik pritisne tipku "u" stvara se direktorij "Up" i svakim pritiskom na tu tipku snima se slika i sprema u taj direktorij. Tablica 4.2. prikazuje mapiranje svake tipke na jednu od 6 gesta ili na upute za program.



Slika 4.3. Prikaz aplikacije GestureCapture.

Tipka	Radnja
"u" ili "U"	Up
"d" ili "D"	Down
"g" ili "G"	Grab
"p" ili "P"	Open Palm
"c" ili "C"	Call
"t" ili "T"	Timeout
"s" ili "S"	zaustavljanje Timeouta
"q" ili "Q"	prekid aplikacije

Tablica 4.2. Tipke za različite radnje u GestureCapture aplikaciji.

Geste Up, Down, Grab, Open Palm i Call zahtjevaju samo jednu ruku te je druga slobodna za pritisak tipke. Gesta Timeout zahtjeva obje ruke pa je funkcionalnost za snimanje slika morala biti modificirana. Kada korisnik pritisne tipku "t" pokreće se režim automatskog snimanja slike svake sekunde, ali prvo se čeka 3 sekunde kako bi se mogla postaviti poza. Na ovaj način je spriječeno spremanje slika koje ne predstavljaju Timeout gestu. Kada korisnik želi zaustaviti snimanje ove geste, to može učiniti pritiskom na tipku "s".

Na konzoli se također ispisuju redni brojevi snimljenih slika kako bi korisnik imao povratnu informaciju o tome koliko je slika već snimljeno. Ovo je bilo jako korisno jer su sudionici bili zamoljeni da za svaku gestu snime jednak broj slika kako bi HANDS bio balansiran. U prilogu B, nalaze se detaljne upute koje su poslone svakom sudioniku. Tu su opisani potrebni resursi, opis gesti, upute za korištenje aplikacije i način slanja podataka koji su se na kraju objedinili u jedinstveni skup podataka HANDS.

4.5. Prikupljanje podataka

Nakon specifikacije gesta, izrade aplikacije GestureCapture i uputa za korisnike, slijedi prikupljanje podataka. Ciljna skupina za ovaj zadatak bili su kolege, prijatelji i članovi obitelji koji već posjeduju neke ili sve potrebne resurse i informatičku podlogu, poput instaliranog Pythona, integriranog razvojnog okruženja kao što je Visual Studio Code ili nekog drugog, te su upoznati s pokretanjem koda. Program, uputstva i obrazac za priloženi poslani su im na vlastitu e-mail adresu ili putem drugih komunikacijskih kanala (e.g. Whatsapp, Messenger) kako bi ih mogli preuzeti i isprobati.

U prikupljanju je sudjelovalo 10 ljudi. Najčešći prijavljeni problemi bili su nekompatibilnost Pythona i traženih biblioteka, postavke privatnosti koje su onemogućavale pokretanje kamere kao i neke nejasnoće vezane uz uputstva. Generalno je povratna informacija za aplikaciju da je intuitivna i jednostavna za korištenje.

4.6. Čišćenje, validacija i analiza podataka

S obzirom da HANDS nije velik skup podataka te da je važna kvaliteta i točnost podataka odlučeno je da se čišćenje podataka obavi ručno čime se istovremeno mogla izvršiti i njihova validacija. Ručno čišćenje podataka je postupak provjere, filtriranja i uređivanja podataka koje obavlja čovjek, umjesto da se koristi automatizirani ili poluautomatizirani proces [29]. Na ovaj način osigurava se bolja točnost i pouzdanost jer su ljudi bolji u prepoznavanju suptilnih nedostataka od automatiziranih algoritama [30].

Skup podataka na početku je sadržavao 1650 slika. Nakon brisanja i balansiranja sadržava 1610 slika, po 230 za svaku kategoriju. Ulazni podaci bili su većinom ispravni, što nije zahtijevalo puno manualnog čišćenja. Ovo signalizira da su sudionici ispravno slijedili upute te su možda i sami obrisali slike za koje su smatrali da nisu zadovoljavajuće. Najčešći razlozi brisanja bili su: niska rezolucija, mutna slika zbog pomicanja ruke prilikom slikanja, netočna gesta ili balansiranje skupa podataka ako je neki direktorij sadržavao previše slika.

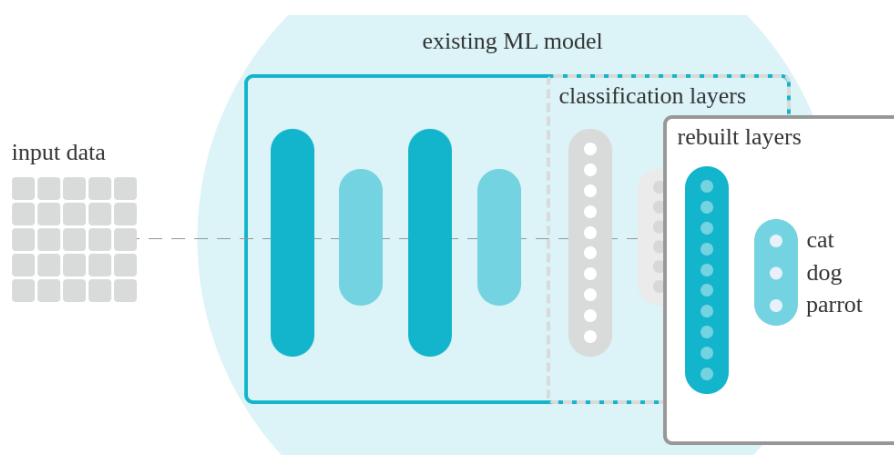
5. Prilagođeni model za prepoznavanje gesta

U ovom poglavlju predstavlja se prilagođeni model za prepoznavanje gesta, razvijen pomoću MediaPipe Model Makersa. U nastavku se detaljno opisuje postupak prilagodbe modela, učitavanje podataka, treniranje i rezultate koji su postignuti.

5.1. MediaPipe Model Maker

MediaPipe Solutions u paleti svojih mogućnosti nudi alat MediaPipe Model Maker za prilagođavanje postojećih modela strojnog učenja koji se mogu trenirati na vlastitim skupovima podataka. Model Maker koristi tehniku treniranja modela koja se zove prijenosno učenje (engl. *transfer learning*), kojom se postojeći modeli ponovno treniraju na novim podacima. Ova tehnika koristi značajan dio postojeće logike modela što znači da treniranje traje manje vremena nego treniranje potpuno novog modela i može se obaviti s manje podataka. Na primjer, kod prilagodbe modela za klasifikaciju, dovoljno je imati oko 100 uzoraka podataka za svaku klasu kako bi model bio učinkovit, što značajno štedi memoriju i ubrzava proces treniranja.

Kada se pripremi vlastiti skup podataka i preda modelu na treniranje, alat ponovno trenira model uklanjanjem nekoliko posljednjih slojeva modela koji klasificiraju podatke te izgrađuje nove slojeve za klasifikaciju pomoću skupa podataka koji mu je predan što se može vidjeti na slici 5.1. To je razlog zašto rezultirajući model može prepoznati samo klase navedene u novim podacima što predstavlja ograničenje prilagodbe modela. Ako je originalni model bio treniran za 10 klasa, a ponovno se trenira za 5, model će prepoznati samo tih 5 novih klasa. Model Maker također podržava neke opcije za fino podešavanje slojeva modela (engl. *fine tuning*) kako bi se poboljšala točnost i performanse.



Slika 5.1. Način rada Model Makera za klasifikacijski model

5.2. Klasifikacija

Nakon oblikovanja HANDS skupa podataka, želimo ga proslijediti prilagođenom Hand Gesture Classification modelu kako bi prepoznao nove specificirane geste iz tog skupa podataka, a postupak je detaljno objašnjen u vodiču za prilagodbu modela za prepoznavanje gesta [31]. Kako bi se ostvarila klasifikacija, korištena je već pripremljena `MediaPipe gesture_recognizer.ipynb` Jupyter bilježnica koja se može dodatno uređivati [32]. Dodani su dijelovi koda za analizu i vizualizaciju skupa podataka. Bilježnica je pokrenuta u Google Colaboratory (i.e. Colab) okruženju koji ne zahtijeva nikakvo postavljanje za korištenje osim posjedovanja Google računa te pruža besplatan pristup računalnim resursima za obradu podataka i strojno učenje, uključujući grafičke procesorske jedinice (engl. *Graphics Processing Unit, GPU*) i jedinice za obradu tenzora (engl. *Tensor Processing Unit, TPU*) [33]. Colab je popularan za korištenje jer omogućava korisnicima pisanje i izvršavanje Python koda, suradnju, dijeljenje te spajanje na Google Drive [34].

5.2.1. Učitavanje podataka

HANDS je učitana na Google Drive kojeg smo onda povezali s bilježnicom kako bi dobila pristup tim podacima. Traženi format skupa podataka je

```
<putanja_skupa_podataka>/<naziv_oznake>/<ime_slike>.*
```

gdje jedan direktorij naziva oznake mora biti "None". Ta oznaka predstavlja gestu koja nije klasificirana kao nijedna druga gesta iz skupa podataka. Prilikom kreiranja HANDSa, o tome se vodilo računa te je prilikom provjere potvrđeno da zadovoljava taj format.

Nakon učitavanja podataka, poziva se model za prepoznavanje karakterističnih točaka ruke iz slika, a slike se prije toga promiješaju (engl. *shuffle*) (i.e. nasumično se preuredi redoslijed primjera u skupu podataka). HANDS sadržava 1610 slika. Sve slike na kojima prepoznavanje nije bilo moguće su izbačene iz skupa podataka te ih je na kraju ostalo 1464. Skup podataka s kojim će se vršiti klasifikacija više nisu slike, već su to ekstrahirane značajke karakterističnih točaka ruke iz svake slike. Skup podataka je onda podijeljen na skup za treniranje, skup za validaciju i skup za testiranje u omjeru 80:10:10. Broj primjera za svaki skup možemo vidjeti u tablici 5.1.

Skup	Broj primjera
Treniranje	1171
Validacija	146
Testiranje	147

Tablica 5.1. Podjela skupa podataka HANDS.

5.2.2. Treniranje

Treniranje prilagođenog modela započinje nakon što metodi *create* prosljedimo skup za treniranje, skup za validaciju te opcije koje predstavljaju hiperparametre modela koji kontroliraju proces učenja, omogućuju fino podešavanje i poboljšanje performansi modela. Kako bismo našli najbolju kombinaciju hiperparametara (i.e. onu koja daje najveću točnost modela), treniranje je izvršeno za više kombinacija. Treniranje je bilo jako brzo, za svaku epohu prosječno oko 3 sekunde. Razlog je što je skup podataka bio malen, a polazišni model je dobro optimiran i predtreniran. Takav ishod je bio očekivan obzirom na svojstva MediaPipe Model Maker rješenja. Rezultati treniranja bit će predstavljani u potpoglavlju 5.2.3.

5.2.3. Rezultati

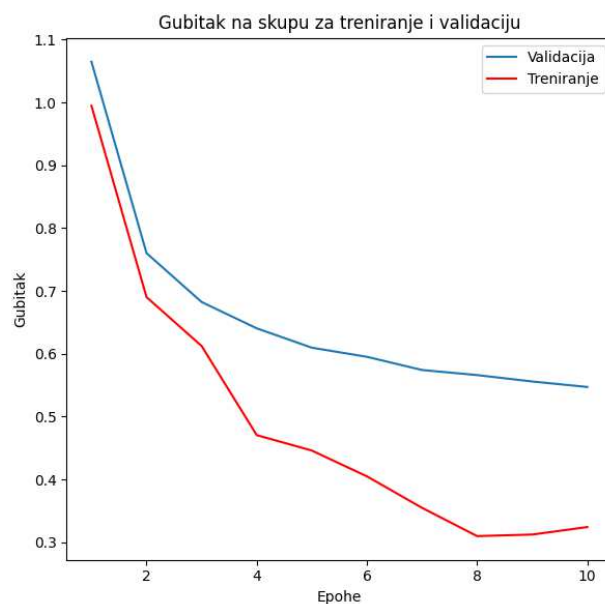
Nakon treniranja, na skupu podataka za testiranje evaluira se prilagođeni model za prepoznavanje geste ruke te se kao rezultat dobiva točnost i gubitak na skupu za testiranje. Rezultati uz konfiguraciju hiperparametara, stope učenja i broja epoha, prikazani su u tablici 5.2.

Naziv modela	Stopa učenja	Broj epoha	Gubitak	Točnost
bazni model	0.001	10	0.1658	0.9048
model 1	0.001	100	0.1595	0.8912
model 2	0.003	100	0.1799	0.8979

Tablica 5.2. Specifikacija Hand Gesture Classification modela [22]

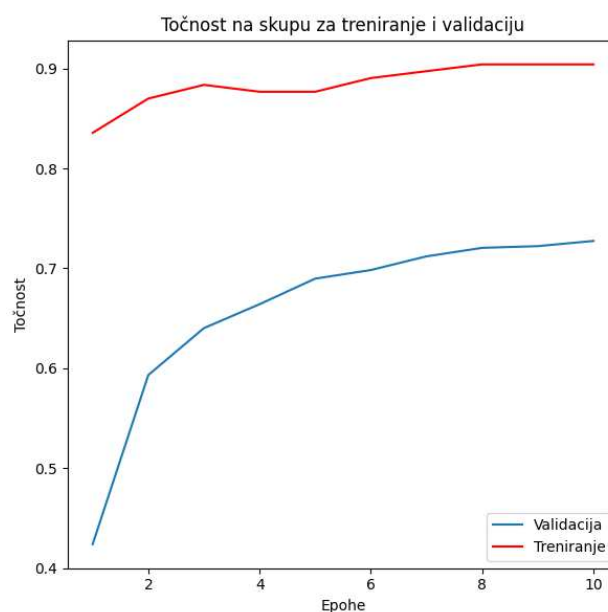
Najveću točnost na skupu za testiranje, 0.9048 dao je bazni model koji sadržava konfiguraciju parametara postavljenu od strane MediaPipe Model Makera te je on korišten za daljnju analizu i integraciju. Može se vidjeti da povećanjem epoha pada točnost na testnom skupu te se može zaključiti čak da je dovelo do laganog pretreniranja (engl. *overfitting*). Rezultati treniranja baznog modela za prepoznavanje gesta prikazani su na grafovima gubitka i točnosti za treniranje i validaciju kroz epohe. Analiza tih rezultata omogućava uvid u performanse modela i njegovo ponašanje tijekom treniranja.

Graf gubitka prikazan na slici 5.2. pokazuje da model na skupu za treniranje značajno smanjuje gubitak kroz prvih nekoliko epoha, nakon čega se smanjenje usporava i stabilizira. Ovo sugerira da model uspješno uči iz podataka za treniranje, što rezultira boljim prepoznavanjem gesta s manje pogrešaka. Gubitak na skupu za validaciju također opada, ali sporijim tempom u usporedbi s treniranjem, i stabilizira se u kasnijim epohama. Stabilizacija gubitka na skupu za validaciju sugerira da model postiže dobru generalizaciju na neviđenim podacima, što je ključan aspekt za stvarnu primjenu modela.



Slika 5.2. Prikaz grafa pogreške treniranja i validacije.

Graf točnosti prikazan na slici 5.3. pokazuje brzi rast točnosti na skupu za treniranje, koja doseže vrlo visoke vrijednosti, oko 90%, nakon nekoliko epoha. To ukazuje na sposobnost modela da brzo nauči prepoznavati geste iz podataka za treniranje i pokazuje odlične performanse predtreniranog modela. Točnost na skupu za validaciju također raste, iako sporijim tempom, i stabilizira se oko 80%. Ovaj rezultat pokazuje da model dobro prepoznaje geste na neviđenim podacima, što potvrđuje njegovu korisnost i učinkovitost prilikom primjene.

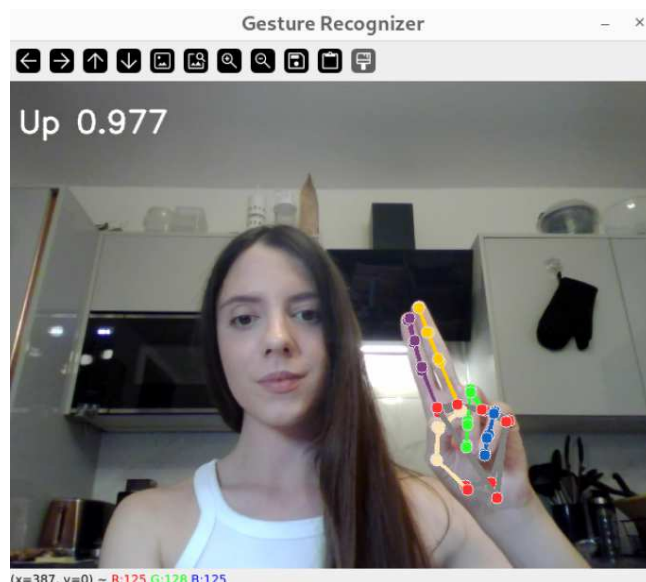


Slika 5.3. Prikaz grafa točnosti treniranja i validacije.

Na kraju, bazni model se pretvori u format Tensorflow Lite s ekstenzijom `.task` te je nazvan `gesture_recognizer.task`. Nakon što se preuzme, model se može implementirati i upotrebljavati za mobilne i desktop aplikacije razne namjene.

Kako bi se isprobao model, izrađena je jednostavna Python skripta `GestureRecognizer` po uzoru na MediaPipe vodič za prepoznavanje gesta u Python aplikacijama [35]. Programsko rješenje koristi biblioteke MediaPipe i openCV te kameru računala. Prilikom pokretanja, otvara se video prijenos gdje korisnik vidi svoj prikaz. Za svaki okvir videa (i.e. sliku), model pokušava prepoznati gestu. Kada model prepozna ruku, obojaju se povezane karakteristične točke ruke te ako model uspješno prepozna gestu ispiše se njezina oznaka. Izlaz iz aplikacije ostvaruje se pritiskom tipke "Esc". Demonstracija aplikacije dana je na slici 5.4.

Prilikom isprobavanja modela u aplikaciji `GestureRecognizer` uočeno je da model nekada krivo klasificira gestu `Grab` kao `Open Palm` i obrnuto. Razlog je što te dvije geste, ovisno koliko osoba otvori ili zatvori dlan pokazuju sličnosti te je pogreška modela očekivana.



Slika 5.4. Prikaz aplikacije `GestureRecognizer`.

6. Studijski slučaj korištenja modela u ozbiljnoj igri

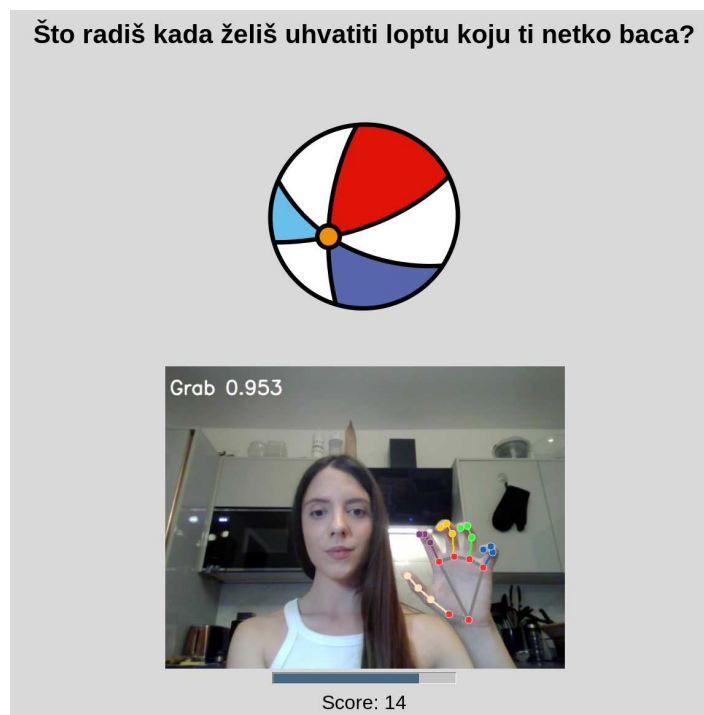
Kako bi se demonstrirale mogućnosti prilagođenog modela za prepoznavanje gesta, odlučeno je primijeniti ga na studijski slučaj ozbiljne igre. U nastavku ovog poglavlja bit će opisana ozbiljna igra "Pogodi gestu!".

6.1. Ozbiljna igra "Pogodi gestu!"

Koristeći tehnologiju prepoznavanja 6 gesta ruke, Call, Up, Down, Grab, Timeout, Open Palm, razvijena je jednostavna edukativna igra "Pogodi gestu!". Svrha ove aplikacije je pokazati primjenu modela za prepoznavanje gesta u interakciji čovjeka i računala kroz edukativnu igru. Cilj same igre je da igrači odgovore na postavljena pitanja ispravno izvršavajući odgovarajuću gestu i da se na taj način upoznaju s alternativnim načinom komunikacije.

6.1.1. Opis

Koncept igre temelji se na tome da korisnik odgovara na postavljena pitanja prikazivanjem odgovarajuće geste. Osim teksta pitanja, korisniku je prikazana i dodatna slika kao pomoć i asocijacija koja sugerira koja bi gesta mogla biti točan odgovor. Fond igračkih pitanja sadrži ukupno 18 pitanja, po tri za svaku gestu, a u jednoj rundi igre korisniku se prikazuje 15 nasumično odabranih pitanja koja se ne ponavljaju. Primjer jednog od pitanja je: "Što radiš kada želiš uhvatiti loptu koju ti netko baca?", na koje se očekuje gesta "Grab", što označava odgovor "uhvatiti loptu". Ovaj primjer može se vidjeti na slici 6.1.



Slika 6.1. Prikaz sučelja igre "Pogodi gestu!".

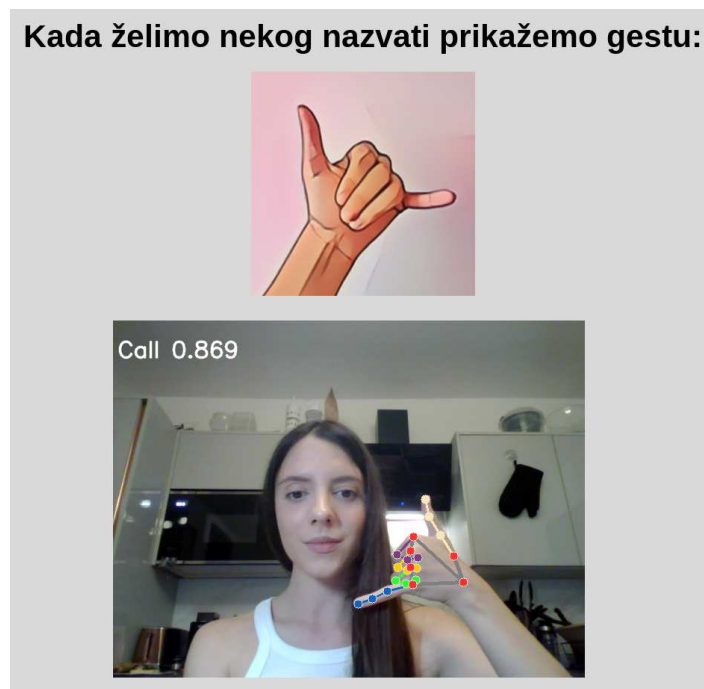
Kada korisnik pokaže ispravnu gestu koju model prepozna s minimalnom vjerojatnošću od 80%, prikazuje se novo pitanje sa slikom. Ispod video prijenosa na ekranu prikazuje se traka za napredak koja služi kao indikator preciznosti prikazane geste, a ispod nje nalazi se prikaz bodova. Nakon što korisnik točno odgovori na svih 15 pitanja, igra je gotova te se ispisuje rezultat i vrijeme potrebno za dovršetak runde.

Početni ekran igre sadržava dvije opcije, prva je za započeti igru i pritiskom te opcije otvara se ekran prikazan na slici 6.1., a druga je prikaz gesta koja služi za upoznavanje korisnika s gestama koje igra podržava i može se vidjeti na slici 6.2.



Slika 6.2. Prikaz početnog ekrana.

Na vrhu ekrana prikazuje se što gesta označava, npr. gesta Call u igri znači "pozovi pomoć". Ispod teksta se nalazi slika koja prikazuje kako gesta izgleda, a ispod slike se nalazi video prijenos prednje kamere u kojem korisnik vidi sebe te se njegove ruke bo-
jaju karakterističnim točkama povezane linijama i može isprobati prikazanu gestu što se može vidjeti na slici 6.3. Kada se gesta ispravno prikaže, odnosno model je prepozna s minimalnom vjerojatnošću od 80%, prikazuje se nova gesta koju je potrebno isprobati. Nakon što se isprobaju sve geste, korisnik dobije ispis poruke na ekran da je spreman za igru te je može započeti klikom na dugme "Započni igru".



Slika 6.3. Prikaz početnog ekrana.

6.1.2. Arhitektura

Igra "Pogodi gestu!" implementirana je koristeći Python uz korištenje različitih biblioteka i alata. Biblioteka Tkinter [36] koristila se za kreiranje korisničkog sučelja s grafičkim elementima koje uključuje naslov igre, gumb za započinjanje igre, prikaz pitanja, trake za napredak, rezultat igre te gumb za zatvaranje igre. Korišten je OpenCV [28] za prikaz video prijenosa u stvarnom vremenu te Pillow [37] za manipulaciju i prikaz slika povezanih s postavljenim pitanjima.

Za prepoznavanje gesta korištena je biblioteka Mediapipe [13] koja omogućava prepoznavanje karakterističnih točaka i interpretaciju gesta ruku s visokom točnošću. Pri-

lagodeni model za prepoznavanje gesta `gesture_recognizer.task`, predstavljen u poglavlju 5., učitani je i integriran u aplikaciju za obradu okvira videa u stvarnom vremenu pomoću stvaranja instance `GestureRecognizerOptions` u načinu rada `LIVE_STREAM` što se može vidjeti na isječku koda u nastavku.

```
options = GestureRecognizerOptions(  
    base_options=BaseOptions(  
        model_asset_path='gesture_recognizer.task'),  
        running_mode=VisionRunningMode.LIVE_STREAM,  
        result_callback=print_result)
```

Ova arhitektura omogućuje interaktivno iskustvo korisnika s igrom "Pogodi gestu!", istovremeno koristeći napredne tehnologije za obradu slika i prepoznavanje gesta kako bi korisnik imao što brže, preciznije i zabavnije iskustvo.

6.1.3. Svrha

Glavna svrha edukativne igre "Pogodi gestu!" je pokazati korisnicima na zabavan način mogućnost interakcije s računalom putem komunikacije gestama. Igra potiče igrače na razvoj motoričkih vještina putem izvođenja preciznih gesta kako bi odgovorili na postavljena pitanja. Pokazivanjem gesta, igrači poboljšavaju koordinaciju ruku te fine motoričke sposobnosti. Također, igra potiče logičko razmišljanje jer svaki odgovor u igri zahtijeva specifičnu gestu, što igrače potiče na pravilno tumačenje postavljenih pitanja i odabir odgovarajućeg gesta za ispravan rezultat. Integracija gesta kao načina odgovaranja na pitanja potiče igrače na aktivno povezivanje između vizualne percepcije i verbalnog odgovora, što potiče brže učenje i bolje razumijevanje konteksta. Igra dodatno stimulira kognitivne procese poput pažnje, memorije i brzine obrade informacija, potičući igrače na aktivno angažiranje njihovih kognitivnih sposobnosti. Kombinacija edukacije i zabave omogućuje učinkovit prijenos znanja o alternativnim oblicima komunikacije kroz praktično iskustvo, motivirajući igrače na sudjelovanje i učenje.

6.1.4. Evaluacija

Za evaluaciju igre "Pogodi gestu!" provedeno je testiranje s pet korisnika u dobi od 20 do 27 godina, koji su označili srednje visoku do visoku razinu informatičke pismenosti. Nakon što su korisnici odigrali igru, zamoljeni su da popune anketu izrađenu putem *Google Forms* obrasca kako bi izrazili svoje mišljenje o igri, njenoj interaktivnosti, jednostavnosti korištenja te općem zadovoljstvu. Rezultati ankete pokazali su da su korisnici općenito bili vrlo zadovoljni igrom. Svi su ocijenili igru kao zabavnu i poučnu, ističući da su uživali u novom načinu interakcije s računalom putem gesta. Korisnici su posebno pohvalili intuitivnost korisničkog sučelja te su naglasili kako je igra jednostavna za razumijevanje i korištenje.

Međutim, evaluacija je također otkrila nekoliko područja za poboljšanje. Korisnici su predložili dodavanje više različitih gesta kako bi igra bila još izazovnija i raznolikija. Također, neki su korisnici smatrali da bi igra mogla imati više vizualnih i zvučnih povratnih informacija kako bi bila još atraktivnija. Drugi prijedlog bio je uvođenje različitih razina težine kako bi igra bila prilagodljiva korisnicima različitih dobnih skupina i sposobnosti. Svi sudionici su se složili da je igra korisna za poboljšanje motoričkih vještina te da potiče logičko razmišljanje i brže procesuiranje informacija. Većina je izrazila želju da se igra dodatno razvija i proširi, te su pokazali interes za sudjelovanje u budućim verzijama igre. Evaluacija je pokazala da je igra "Pogodi gestu!" uspješno demonstrirala mogućnosti prilagođenog modela za prepoznavanje gesta u edukativnom i zabavnom kontekstu te će povratne informacije korisnika bit će korisne za daljnji razvoj i unapređenje igre.

7. Zaključak

U ovom radu istražena je integracija tehnologije prepoznavanja gesta u ozbiljne igre te je predstavljen model rješenja koji koristi MediaPipe Solutions za prepoznavanje gesta na temelju video ulaza u stvarnom vremenu. Tehnologija prepoznavanja gesta pokazala se kao značajan korak naprijed u interakciji čovjeka i računala, omogućujući prirodniju i intuitivniju komunikaciju. Razvoj modela prepoznavanja gesta uključivao je prikupljanje i obradu podataka, treniranje modela te evaluaciju njegovih performansi. Korištenje MediaPipe platforme pokazalo se kao efikasno rješenje zbog njezine optimizacije za rad u stvarnom vremenu i jednostavne prilagodbe postojećih modela za specifične potrebe. Primjena razvijenog modela prikazana je kroz studijski slučaj ozbiljne igre "Pogodi gestu!", koja koristi prepoznavanje gesta za interaktivnu edukaciju korisnika. Rezultati su pokazali da integracija prepoznavanja gesta može značajno poboljšati korisničko iskustvo, pružajući prirodniji način interakcije s igrom i omogućujući bolje motoričke i kognitivne sposobnosti igrača.

Iako su postignuti značajni rezultati, postoji još mnogo prostora za daljnje istraživanje i unapređenje. Buduća istraživanja mogla bi se fokusirati na prepoznavanje složenijih i dinamičnijih gesta, optimizaciju performansi modela te primjenu ove tehnologije u različitim kontekstima ozbiljnih igara. Na kraju, ovaj rad doprinosi boljem razumijevanju i primjeni tehnologije prepoznavanja gesta u ozbiljnim igrama, te pruža smjernice za daljnji razvoj i istraživanje. Integracija prepoznavanja gesta ima potencijal ne samo unaprijediti korisničko iskustvo u igrama, već i otvoriti nove mogućnosti za edukaciju, terapiju i druge primjene gdje je intuitivna interakcija s tehnologijom ključna.

Popis slika

2.1. Prepoznavanje poze tijela koristeći OpenPose [8].	6
3.1. Prepoznavanje geste palac gore, s točnošću 63% koristeći Gesture Recognizer [19].	8
3.2. Karakteristične točke ruke na izlazu iz Hand Landmarker paketa modela [20].	9
4.1. Geste koje prepoznaje Hand Gesture Classification model [27]	13
4.2. Nove geste za prilagođeni model.	14
4.3. Prikaz aplikacije GestureCapture.	15
5.1. Način rada Model Makera za klasifikacijski model	18
5.2. Prikaz grafa pogreške treniranja i validacije.	21
5.3. Prikaz grafa točnosti treniranja i validacije.	21
5.4. Prikaz aplikacije GestureRecognizer.	22
6.1. Prikaz sučelja igre "Pogodi gestu!".	24
6.2. Prikaz početnog ekrana.	24
6.3. Prikaz početnog ekrana.	25

Popis tablica

3.1. Specifikacija Hand Landmarker paketa modela [21].	9
3.2. Specifikacija Hand Gesture Classification modela [22]	10
4.1. Specifikacija novih gesta	13
4.2. Tipke za različite radnje u GestureCapture aplikaciji.	15
5.1. Podjela skupa podataka HANDS.	19
5.2. Specifikacija Hand Gesture Classification modela [22]	20

Literatura

- [1] J. M. Carroll, “Human computer interaction (hci)”, *Interaction Design Encyclopedia*. Retrieved on June, sv. 6, str. 2010, 2009.
- [2] Y. Liu, “Human-computer interface design based on design psychology”, u *2020 International Conference on Intelligent Computing and Human-Computer Interaction (ICHCI)*, 2020., str. 5–9. <https://doi.org/10.1109/ICHCI51889.2020.00009>
- [3] What is computer vision? Pristupljeno: lipanj 2024. [Mrežno]. Adresa: <https://www.ibm.com/topics/computer-vision>
- [4] Human-computer interaction with computer vision: Challenges and innovations. Pristupljeno: lipanj 2024. [Mrežno]. Adresa: <https://medium.com/@gmrethinagiri1507681/human-computer-interaction-with-computer-vision-challenges-and-innovations-b99c24c8e6ff#:~:text=This%20approach%20uses%20computer%20vision,%2C%20clicking%2C%20or%20dragging%20objects.&text=This%20approach%20uses%20computer%20vision%20to%20infer%20the,from%20their%20actions%20or%20gestures>.
- [5] S. Hussain, R. Saxena, X. Han, J. A. Khan, i H. Shin, “Hand gesture recognition using deep learning”, u *2017 International SoC Design Conference (ISOCC)*, 2017., str. 48–49. <https://doi.org/10.1109/ISOCC.2017.8368821>
- [6] A. Das, K. Maitra, S. Roy, B. Ganguly, M. Sengupta, i S. Biswas, “Development of a real time vision-based hand gesture recognition system for human-computer interaction”, u *2023 IEEE 3rd Applied Signal Processing Conference (ASPCON)*, 2023., str. 294–299. <https://doi.org/10.1109/ASPCON59071.2023.10396583>

- [7] F. Zhan, “Hand gesture recognition with convolution neural networks”, u *2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI)*, 2019., str. 295–298. <https://doi.org/10.1109/IRI.2019.00054>
- [8] The complete guide to openpose in 2024. Pristupljeno: lipanj 2024. [Mrežno]. Adresa: <https://viso.ai/deep-learning/openpose/>
- [9] H. Yang, Q. Liu, i J. Zhang, “Improved human-object interaction based lightweight openpose”, u *2022 International Conference on 3D Immersion, Interaction and Multi-sensory Experiences (ICDIIME)*, 2022., str. 104–107. <https://doi.org/10.1109/ICDIIME56946.2022.00030>
- [10] Yolo: Algorithm for object detection explained. Pristupljeno: lipanj 2024. [Mrežno]. Adresa: <https://www.v7labs.com/blog/yolo-object-detection>
- [11] Ultralytics yolo docs, pose estimation. Pristupljeno: lipanj 2024. [Mrežno]. Adresa: <https://docs.ultralytics.com/tasks/pose/#export>
- [12] (2023) Introducing mediapipe solutions for on-device machine learning. Pristupljeno: lipanj 2024. [Mrežno]. Adresa: <https://developers.googleblog.com/en/introducing-mediapipe-solutions-for-on-device-machine-learning/>
- [13] (2023) Mediapipe solutions guide. Pristupljeno: lipanj 2024. [Mrežno]. Adresa: <https://ai.google.dev/edge/mediapipe/solutions/guide>
- [14] F. Laamarti, M. Eid, i A. El Saddik, “An overview of serious games”, *International Journal of Computer Games Technology*, sv. 2014, 10 2014. <https://doi.org/10.1155/2014/358152>
- [15] Play2green. Pristupljeno: lipanj 2024. [Mrežno]. Adresa: <https://sociallab.fer.hr/play2green/>
- [16] Game-changers: 18 serious games that changed the world. Pristupljeno: lipanj 2024. [Mrežno]. Adresa: <https://www.growthengineering.co.uk/serious-games-that-changed-the-world/>

- [17] C. Rashmika, G. Varsha, i U. Sairam, “Gesture-based control of ping-pong: Developing an interactive game interface”, u *2023 4th International Conference on Smart Electronics and Communication (ICOSEC)*, 2023., str. 1085–1089. <https://doi.org/10.1109/ICOSEC58147.2023.10276168>
- [18] G. M. Kumar, V. Manohar, B. Ravi, S. Prasad, S. Paluvatla, i R. Sateesh, “Game controlling using hand gestures”, u *2022 International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC)*, 2022., str. 1–5. <https://doi.org/10.1109/ASSIC55218.2022.10088337>
- [19] Gesture recognition task guide. Pristupljeno: lipanj 2024. [Mrežno]. Adresa: https://ai.google.dev/edge/mediapipe/solutions/vision/gesture_recognizer
- [20] Hand landmarks detection guide. Pristupljeno: lipanj 2024. [Mrežno]. Adresa: https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker#models
- [21] Mediapipe model card hand tracking. Pristupljeno: lipanj 2024. [Mrežno]. Adresa: [https://storage.googleapis.com/mediapipe-assets/Model%20Card%20Hand%20Tracking%20\(Lite_Full\)%20with%20Fairness%20Oct%202021.pdf](https://storage.googleapis.com/mediapipe-assets/Model%20Card%20Hand%20Tracking%20(Lite_Full)%20with%20Fairness%20Oct%202021.pdf)
- [22] Mediapipe model card hand gesture classification. Pristupljeno: lipanj 2024. [Mrežno]. Adresa: https://storage.googleapis.com/mediapipe-assets/gesture_recognizer/model_card_hand_gesture_classification_with_faireness_2022.pdf
- [23] What is a dataset? Pristupljeno: lipanj 2024. [Mrežno]. Adresa: <https://www.databricks.com/glossary/what-is-dataset>
- [24] A. Kapitanov, K. Kvanchiani, A. Nagaev, R. Kraynov, i . A. Makhliarchuk, “Hagrid - hand gesture recognition image dataset”, *arXiv*, 2022. <https://doi.org/10.48550/arXiv.2206.08219>
- [25] H. Foxwell, *Creating Good Data: A Guide to Dataset Structure and Data Representation*, 01 2020. <https://doi.org/10.1007/978-1-4842-6103-3>
- [26] Complete guide to gdpr compliance. Pristupljeno: lipanj 2024. [Mrežno]. Adresa: <https://gdpr.eu/>

- [27] (2024) Emojis. Pristupljeno: lipanj 2024. [Mrežno]. Adresa: <https://emojipedia.org/apple>
- [28] Opencv. Pristupljeno: lipanj 2024. [Mrežno]. Adresa: <https://opencv.org/>
- [29] (2024) Data cleaning: Definition, techniques best practices for 2024. Pristupljeno: lipanj 2024. [Mrežno]. Adresa: <https://technologyadvice.com/blog/information-technology/data-cleaning/#:~:text=Manual%20data%20cleaning%20is%20needed,duplicate%20data%20from%20the%20dataset>.
- [30] E. Mosqueira-Rey, E. Hernández-Pereira, D. Alonso-Ríos, J. Bobes-Bascarán, i Fernández-Leal, “Human-in-the-loop machine learning: a state of the art”, *Artificial Intelligence Review*, sv. 56, 08 2022. <https://doi.org/10.1007/s10462-022-10246-w>
- [31] Hand gesture recognition model customization guide. Pristupljeno: lipanj 2024. [Mrežno]. Adresa: https://ai.google.dev/edge/mediapipe/solutions/customization/gesture_recognizer
- [32] Hand gesture recognition model customization jupyter notebook. Pristupljeno: lipanj 2024. [Mrežno]. Adresa: https://colab.research.google.com/github/googlesamples/mediapipe/blob/main/examples/customization/gesture_recognizer.ipynb#scrollTo=uqqRuzYN3yvt
- [33] Google colaboratory. Pristupljeno: lipanj 2024. [Mrežno]. Adresa: <https://colab.google/>
- [34] How to use google colab. Pristupljeno: lipanj 2024. [Mrežno]. Adresa: <https://www.geeksforgeeks.org/how-to-use-google-colab/>
- [35] Gesture recognition guide for python. Pristupljeno: lipanj 2024. [Mrežno]. Adresa: https://ai.google.dev/edge/mediapipe/solutions/vision/gesture_recognizer/python
- [36] Tkinter. Pristupljeno: lipanj 2024. [Mrežno]. Adresa: <https://docs.python.org/3/library/tkinter.html>

[37] Pillow. Pristupljeno: lipanj 2024. [Mrežno]. Adresa: <https://pypi.org/project/pillow/>

Sažetak

Prepoznavanje gesta u ozbiljnim igrama

Marija Dragošević

Ovaj diplomski rad istražuje integraciju tehnologije prepoznavanja gesta u ozbiljnim igrama. Cilj je bio stvoriti skup podataka, razviti model prepoznavanja gesta koristeći MediaPipe Solutions te integrirati taj model u ozbiljnu igru kako bi se demonstrirale njegove mogućnosti. Rad uključuje teorijski pregled tehnologija prepoznavanja gesta, postupak stvaranja skupa podataka, razvoj modela prepoznavanja gesta te primjenu tog modela u ozbiljnoj igri "Pogodi gestu!". Rezultati su pokazali da integracija tehnologije prepoznavanja gesta značajno može poboljšati interakciju između igrača i igre, pružajući prirodnu i intuitivnu metodu upravljanja.

Ključne riječi: prepoznavanje gesta; ozbiljne igre; interakcija čovjeka i računala; MediaPipe rješenja

Abstract

Marija Dragošević

This MSc Thesis explores the integration of gesture recognition technology into serious games. The goal was to create a dataset and develop a gesture recognition model using MediaPipe Solutions and integrate this model into serious games to demonstrate its possibilities. It includes a theoretical review of gesture recognition technologies, procedure of creating the dataset, the development of a gesture recognition model, and the application of this model in the serious game "Guess the Gesture!". The results demonstrated that the integration of gesture recognition can significantly improve the interaction between players and the game, providing a more natural and intuitive control method.

Keywords: Gesture recognition; Serious Games; Human-Computer Interaction; MediaPipe Solutions

Privitak A: Privola

Suglasnost za sudjelovanje u prikupljanju podataka za prepoznavanje gesti

Ja sam Marija Dragošević, studentica diplomskog studija na Fakultetu elektrotehnike i računarstva u Zagrebu. Provodim istraživanje o prepoznavanju gesti i stvaram skup podataka za treniranje modela strojnog učenja. Vaše sudjelovanje uključuje fotografiranje dok izvodite razne geste. Prikupljene slike koristit će se isključivo u akademske istraživačke svrhe povezane s mojim diplomskom radom. Podaci neće biti dijeljeni s trećim stranama bez vašeg izričitog pristanka. Svi prikupljeni podaci bit će pohranjeni sigurno i obrađivani u skladu s Općom uredbom o zaštiti podataka (GDPR). Vaši osobni podaci bit će anonimizirani kad god je to moguće kako bi se zaštitila vaša privatnost. Vaše sudjelovanje u ovom istraživanju je dobrovoljno. U bilo kojem trenutku možete povući svoj pristanak za sudjelovanje bez ikakvih posljedica. Ako odlučite povući pristanak, vaši podaci bit će trajno izbrisani iz skupa podataka.

Ako imate bilo kakvih pitanja ili nedoumica u vezi s ovim istraživanjem ili vašim pravima kao sudionika, molimo vas da me kontaktirate na **marija.dragosevic@fer.hr**. Potpisom ovog obrasca potvrđujete da ste pročitali i razumjeli gore navedene informacije te da dobrovoljno pristajete sudjelovati u ovom istraživanju.

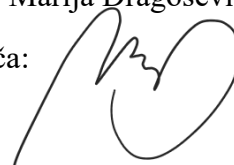
Ime sudionika:

Potpis sudionika:

Datum:

Ime istraživača: Marija Dragošević

Potpis istraživača:



Privitak B: Upute

GestureCapture.py upute za korištenje

GestureCapture aplikacija omogućava korisnicima da snimaju slike različitih gesti koristeći svoju web kameru. Ove slike će se koristiti za treniranje modela za prepoznavanje 6 gesti za moj diplomski rad. Procijenjeno vrijeme trajanja snimanja je 10ak minuta.

Instalacija

1. Preuzmite i instalirajte Python (pretpostavljam da ga već imate)

<https://www.python.org/>

2. Instalirajte potrebne biblioteke (Otvorite terminal -CMD na Windows-u ili terminal u VSC)

```
pip install opencv-python
```

```
pip install opencv-python-headless
```

```
pip install numpy
```

```
pip install pillow
```

Pokretanje i korištenje aplikacije

1. Preuzmite kod GestureCapture.py

2. Otvorite ga u VSC ili pokrenite iz terminala s `python gesture_capture.py`

3. Nakon što pokrenete aplikaciju, otvorit će se prozor sa prikazom web kamere.

Snimanje gesti

VAŽNO! Za svaku gestu potrebno je snimiti 10-20 slika, pod različitim kutem, bliže kameri, dalje kameri, nakrivljeno, **s desnom rukom i lijevom rukom (prednji dlan, stražni dlan)**, ali zadržavajući prirodan oblik geste s nekim uobičajenim varijacijama.

Broj koji se odlučite snimiti za jednu gestu molim vas da bude broj svake geste.

Npr. Odlučite snimiti 20 slika za jednu kategoriju, nastojte da u svakoj kategoriji bude 20 slika. Ako želite snimiti više slika naravno da je dopušteno. Isto tako, vaše lice se može vidjeti a i ne mora na slici, kako god mislite da vam je lakše.

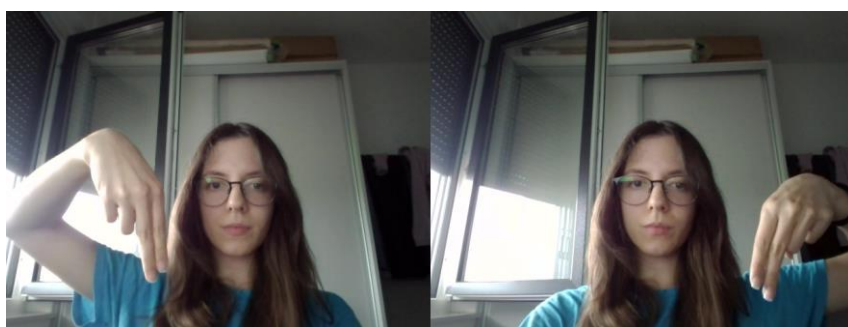
Slika se snima klikom na pojedini gumb za svaku gestu osim **geste Timeout** jer ta gesta zahtjeva **obje ruke** pa je snimanje **automatsko**.

1. Na ekranu će biti prikazane sljedeće upute:

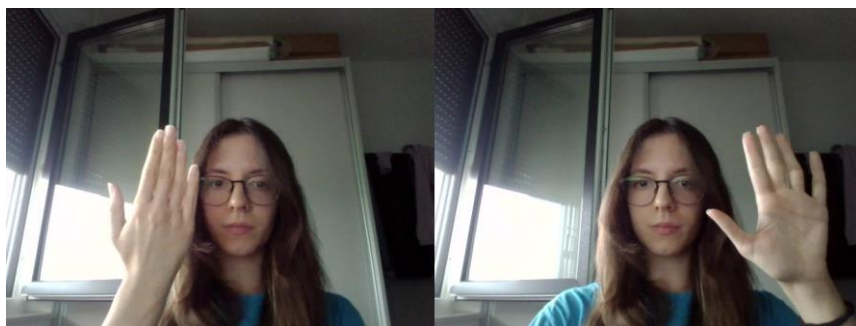
Pritisnite **'u'** za snimanje gesta **"Up"** (gore s dva prsta).



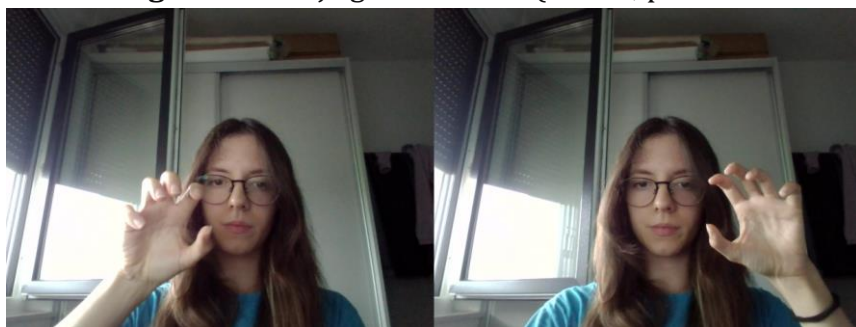
Pritisnite **'d'** za snimanje gesta **"Down"** (dole s dva prsta).



Pritisnite **'p'** za snimanje gesta **"Open Palm"** (otvoreni dlan skupljen ili blago raširen, prednja i stražnja strana).



Pritisnite **'g'** za snimanje gesta **"Grab"** (uhvati, pokret kao da ćete nešto ugrabiti).



Pritisnite 'c' za snimanje gesta "Call" (poziv).



Pritisnite 't' za snimanje gesta "Timeout" (pauza) sa kašnjenjem od 3 sekunde kako biste namjestili pozu i automatskim snimanjem svake sekunde.



Pritisnite 's' da zaustavite snimanje geste "Timeout".

Pritisnite 'q' za izlaz iz aplikacije.

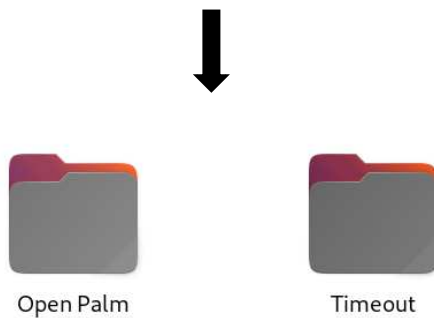
2. Kada pritisnete bilo koje od navedenih tipki za gestu, slika će biti snimljena i sačuvana u odgovarajućem podfolderu za svaku gestu unutar foldera **captured_images**.

Pr.

captured_images/Open Palm/

captured_images/Timeout/





3. Na kraju **zipajte captured_images folder** i proslijedite mi ga molim vas na whatsapp 0994258172 (preferirano) ili može i na mail marija.dragosevic@fer.hr. Ukoliko imate bilo kakvih pitanja ili problema sa aplikacijom, kontaktirajte me.

Hvala svima koji ćete sudjelovati <3