Procjena smjera pogleda iz slika lica korištenjem dubokog učenja

Gracin, Jakov

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 417

PROCJENA SMJERA POGLEDA IZ SLIKA LICA KORIŠTENJEM DUBOKOG UČENJA

Jakov Gracin

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 417

PROCJENA SMJERA POGLEDA IZ SLIKA LICA KORIŠTENJEM DUBOKOG UČENJA

Jakov Gracin

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

DIPLOMSKI ZADATAK br. 417

Pristupnik:	**Jakov Gracin (0036515226)**
Studij:	Računarstvo
Profil:	Računarska znanost
Mentor:	akademik prof. dr. sc. Sven Lončarić

Zadatak: **Procjena smjera pogleda iz slika lica korištenjem dubokog učenja**

Opis zadatka:

Smjer pogleda osobe je izrazito bitan oblik neverbalne komunikacije koji signalizira pozornost i zanimanje. Posljednih godina, metode za procjenu smjera pogleda iz slika lica su sve više u središtu interesa brojnih istraživačkih laboratorija i industrije. U sklopu diplomskog rada potrebno je proučiti metode za procjenu trodimenzionalnog vektora smjera pogleda iz slika lica. Pri tome, posebna pozornost treba biti obraćena na metode koje primjenuju duboko učenje. Potrebno je odabrati neku od metoda procjene smjera pogleda iz literature i primjeniti ju na neki od javno dostupnih skupova podataka. Naposlijetku, generalizacijska moć odabrane metode treba biti ispitana na skupu slika koje model nije vidio prilikom učenja koristeći metriku kutne razlike između vektora predikcije i točnog vektora smjera pogleda.

Rok za predaju rada: 28. lipnja 2024.

Duboke i iskrene zahvale mome Bogu i Spasitelju Isusu Kristu iz Nazareta što mi je obnovio život i sav njegov sadržaj.

Duboke i iskrene zahvale mojim roditeljima Višnji i Mladenu Gracinu na podršci i potpori tijekom života, studija i izrade ovog Diplomskog rada. Iskrene zahvale akademiku prof. dr. sc. Svenu Lončariću, Franku Šikiću, mag. ing. i dr. sc. Doniku Vršnaku na mentoriranju i podršci pri izradi ovog Diplomskog rada te tijekom studija.

Sadržaj

1. Uvod 3

1.0.1. Općenito o metodama procjene i praćenja pogleda 3

1.0.2. Procjena i praćenje pogleda pomoću dubokog učenja 6

2. Eksperiment 8

2.1. Metoda *MultiZoomGaze* 8

2.1.1. Opis 8

2.1.2. Arhitektura - model MSA (eng. *Multiple Scale Aggregation*) 9

2.1.3. Kutevi *yaw* (θ) i *pitch* (ϕ) 12

2.1.4. Funkcija gubitka 15

2.2. Skup podataka "Look Both Ways" 16

2.2.1. Opće informacije 16

2.2.2. Podaci o pogledu vozača 17

2.3. Hardverska podrška 19

2.4. Treniranje 19

2.4.1. Detaljnije o podacima 19

2.4.2. Podaci o pogledu 20

2.4.3. Problemi tijekom treniranja 20

2.4.4. Hiperparametri 22

3. Rezultati 23

3.1. Vizualizacija smjera pogleda 23

3.1.1. Kvaliteta rezultata 25

3.2. Metrike treniranja 25

4. Diskusija 27

4.1. Usporedba s predviđanjem pogleda iz [1] 27
4.2. Nedostaci provedene metode 28

5. Budući rad 30

6. Zaključak 31
6.1. Sumacija metode 31
6.2. Rezultati 31

Sažetak 32

Abstract 33

A: Programski kod 34
A1. Generiranje slike strelice pomoću *pyplot* paketa 34

Literatura 40

1. Uvod

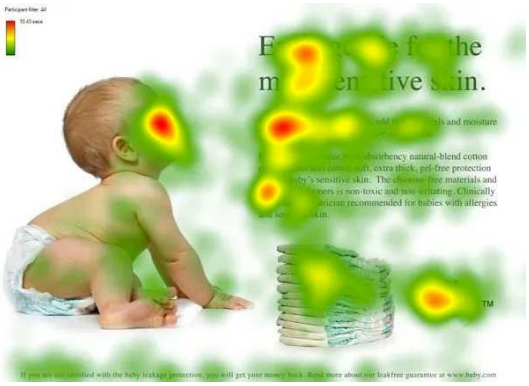
1.0.1. Općenito o metodama procjene i praćenja pogleda

Područje istraživanja vezano za praćenje ljudskog pogleda bilo je aktivno već u drugoj polovici prošlog vijeka, znatno prije najsuvremenijih primjena u autonomnoj vožnji, VR/AR i *gaming* kontekstima, mobilnim aplikacijama te ostalih područja endemskima proteklim dvama desetljećima. Iz literature i povijesti industrije jasno je vidljivo da su razvijene raznolike metode [5] temeljene na potpuno različitim principima rada.

Prije razdoblja ubrzanog razvoja računalnog vida temeljenog prvenstveno na dubokom učenju u proteklih nekolicinu godina, istraživanja iz prošlog stoljeća oslanjala su se većinom na mehaničke i optičke metode praćenja. Primjerice, umetanje sitne zavojnice u oko kojim se pomoću promjene magnetskog polja prate kretnje oka razvijeno je već u 60-im godinama prošlog stoljeća [10].

Nešto manje invazivan ali i dalje nezgrapan način praćenja pogleda koristi činjenicu da okom upravljaju mišići te se mjerenjem aktivacije mišića koji okružuju oko procjenjivala promjena smjera subjektova pogleda [11]. Taj pristup, zvan elektromiografija (EMG), pojavio se također u 60-im godinama prošlog stoljeća, a sve do danas se koristi u svrhe medicinskih dijagnostika, neuroloških istraživanja i slično.

Slika 1.1. Suvremeno praćenje pogleda pomoću EMG-a i VR-a [17]



Slika 1.2. *Heatmap* - praćenje pogleda u istraživanju korisničkog iskustva *web* stranice

Procjenjivanje ljudskog pogleda ima značajnu sadašnju i buduću ekonomsku važnost, te je postojala inicijativa da se metode razviju na čim jeftiniji, jednostavniji i brži način te po mogućnosti neinvazivno po subjekta. Razvoj kamera visoke rezolucije i dubokih neuronskih mreža u prošla dva desetljeća omogućili su jednostavan i praktičan način procjene pogleda, za razliku od hardverski kompliciranih i nezgrapnih medota 20. stoljeća. U najnovije vrijeme moguće je pomoću obične kamere *smartphone*-a i duboke neuronske mreže razumne veličine dobiti vrlo prihvatljive rezultate za velik broj primjena - bilo da se radi o već zreloj industriji, za privatne hobističke projekte ili bilo što između.

Primjene procjene smjera pogleda

U najpćenitijem kontekstu, procjena smjera pogleda koristi se za modeliranje i razumijevanje ljudske namjere i pozornosti te za predviđanje sljedećeg poteza u raznim scenarijima gdje je potrebno reagirati na podražaje iz okoline. U ostalome, praćenje pogleda koristi se u istraživanju interakciji čovjeka i računala (eng. *human-computer interaction*, HCI) što obuhvaća AR/VR, razvoj korisničkih sučelja (UX/UI), te analizu korisničkog iskustva (eng. *user research*) kod *smartphone*-a a i PC-a.

Mobilne aplikacije u svrhu zabave isto tako imaju određenu primjenu (npr. razvoj novih *Snapchat* filtera).

Nadalje u području marketinga te dizajna i estetike proizvoda izvan područja IT-a, praćenje pogleda može veoma pridonijeti kvalitetnom i efikasnom razvoju dizajna [4] i osmišljavanju optimalne marketinške kampanje [12]. Na kraju bitno je istaknuti i istraživanje u području autonomne vožnje [6] kao jedne od najznačajnijih primjena koja će

uvelike utjecati na kvalitetu i stil života u bliskoj budućnosti.

Metode

Postoji osnovna podjela metoda praćenja pogleda s obzirom na krajnji rezultat eksperimenta, tj. dimenzionalnost rezultatnog vektora. Metode se dijele na:

- dvodimenzionalnu (2D) procjenu smjera pogleda - PoG (eng. *Point of gaze*) vektor ograničen na dvije dimenzije prisutan na samoj slici gdje se nalazi i subjekt čiji se pogled procjenjuje.
- trodimenzionalnu (3D) procjenu smjera pogleda - trodimenzionalni vektor u prostoru, gdje se prostor aproksimira na temelju informacija iz dvodimenzionalne slike. U ovom slučaju postoje varijante vektora u polarnim ili kartezijevim koordinatama.

Hardverska podrška

Kroz proteklih pola stoljeća više su se različitih vrsta hardverskih komponenata koristile u istraživanju i procjeni smjera pogleda. U počecima istraživanja u 60-im godinama prošlog stoljeća su kod određenih eksperimenata [10] bili prisutni kirurški implantirana bakrena zavojnica u oko (invazivan način procjene smjera gledanja), generator magnetskog polja, pojačivač signala, stabilizator glave, i još nekolicina manjih komponenata. U drugim slučajevima , na primjer kod elektromiografije (EMG) [11], koristile su se elektrode, žice koje spajaju elektrode s elektroničkom opremom, pojačivač signala, oprema za procesuiranje električnog signala te osciloskop za snimanje električnog signala, uz nekolicinu pomoćnih komponenata.

Jasno je da su te metode poprilično nezgrapne i da zahtijevaju mnogo pripreme, kalibracije i velik broj skupocjenih i specijaliziranih hardverskih komponenata.

S druge strane, u protekla se dva desetljeća razvoj istraživanja smjera ljudskog pogleda ubrzao između ostalog i zbog toga što je relativno trivijalno postaviti hardversku opremu. Kako su metode evoluirale iz onih koje se oslanjaju na fizikalne tjelesne parametre koje proizvodi ljudsko oko u pokretu, u novije se vrijeme smjer pogleda može (za praktične svrhe dovoljno) vjerno procjeniti isključivo na temelju slike subjekta čiji smjer pogleda procjenjujemo. Budući da su kamere postale sve jefinije, dostupnije i imaju sve

Slika 1.3. *Multiple camera setup* [18] - može se koristiti za praćenje pogleda

veću rezoluciju, s vremenom su glavno usko grlo u istraživanju umjesto hardvera za snimanje podataka postale arhitekture neuronskih mreža te pravna problematika javnih skupova podataka za treniranje.

U općenitom smislu, kamere koje su korištene u istraživanju uključuju RGB-D kamere (kamere koje uz boju i izgled snimaju i dubinu prostora), infracrvene kamere te *multicam* postavka - snimanje s više kamera paralelno. Uz to, nalazimo i zaslone nošene na glavi (eng. *head-mounted display*), *web*-kamere i napokon kamere *smartphone*-a koje su dovoljno visoke rezolucije da budu vrlo kompetitivne ostalim vrstama kamera koje se koriste za istraživanje smjera pogleda. Doduše, jedno je ograničenje kod *smartphone*-a to što se procjenjuju 2D PoG vektori unutar mobilnih aplikacija koje predviđaju pogled - zbog manjka računalne moći.

1.0.2. Procjena i praćenje pogleda pomoću dubokog učenja

Relevantni javno dostupni skupovi podataka

Od popularnih skupova podataka bitno je spomenuti MPIIGaze, Gaze360 [7], RT-GENE, ETHXgaze te GazeCapture, između ostalih. Naposljetku pojavljuje se i skup podataka *Look Both Ways* [2], koji se koristio u ovom radu.

Problemi sa skupovima podataka

Neke od poteškoća kod skupova podataka su složena pravna situacija, sigurnost i javna dostupnost skupa podataka. Budući da se na slikama radi o stvarnim osobama, postoje

pravne zaštite poput GDPR-a u Europskoj uniji koje štite slikane subjekte.

Također, nije lako naći kvalitetne i dobro označene skupove podataka. Nije velik broj lako dostupnih i dobro označenih skupova podataka koji imaju dovoljno slika da bi se moglo provesti treniranje. U svakom je slučaju lakše pronaći neoznačene slike za testiranje nego označenih slika za treniranje, (primjerice iz javno dostupnih filmova ili serija) te onda jedina mogućnost ostaje vizualno provoditi provjeru.

U idealnom slučaju bi model trebao moći točno procijeniti smjer pogleda na slikama u raznim uvjetima: niske rezolucije, različite razine svjetline, uz raznolike pozicije glave te unatoč obstrukcijama (gdje se oči ne vide direktno, npr. zbog naočala, kapa i sličnih objekata). Ipak, izazovno je naći skup podataka koji bi mogao imao i takve slike za treniranje, a da su pritom označene.

Podaci - Video i slika

Traženi model može procjeniti smjer pogleda na temelju slike. Ako se pak podaci sastoje od videa, onda je potrebno napraviti korak predobrade podataka da se dobiju pojedine sličice videa (eng. *frame*-ovi), te se onda problem tretira na isti način kao i kod zasebnih slika.

Ipak, bitno je uočiti problematiku nalaženja odgovarajuće metode za analizu *frame*-ova videa. U nastavku slijede neki od pristupa ekstrahiranja informacije o pogledu iz *frame*-ova:

- Ekstrahiranje informacije o pogledu zasebno u svakom *frame*-u - sliku po sliku
- Korištenje *frame*-ova prije i nakon traženog *frame*-a da bi se analizirao pogled pristuan na njemu, kao što je to slučaj kod [1] i [7].

2. Eksperiment

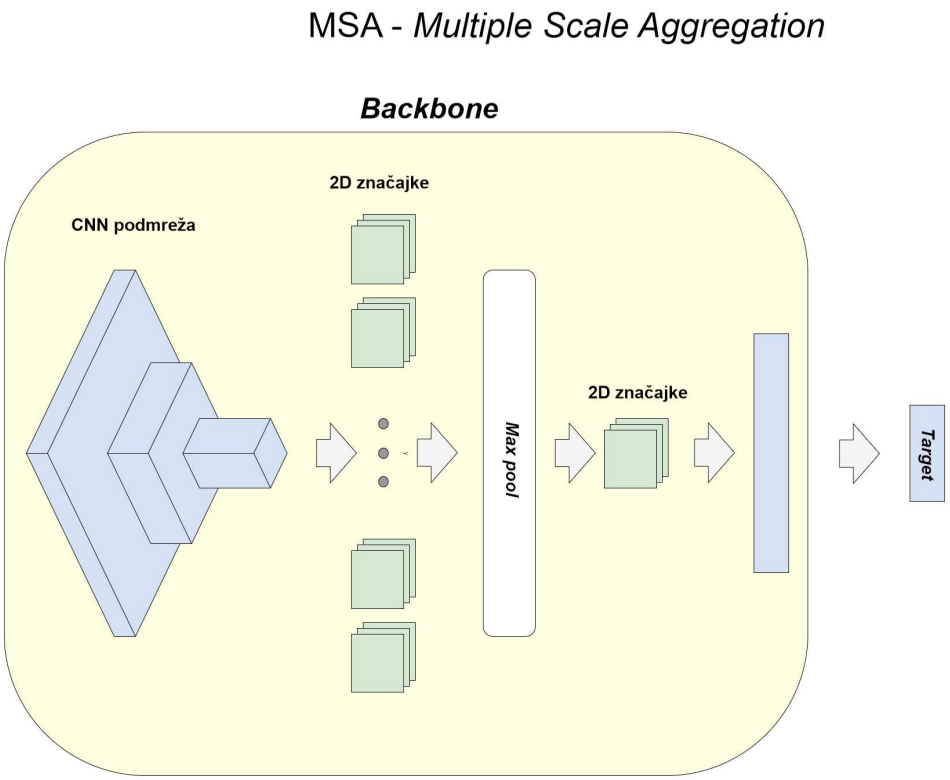
2.1. Metoda *MultiZoomGaze*

2.1.1. Opis

Model korišten u okviru ovog rada koristi agregiran način obrade slike osobe da bi se iz nje ekstrahirala informacija o pogledu. Konkretno, uzima se više različitih razina povećanja (*zoom-a*) jedno te iste slike te se paralelno postavljaju na ulaz mreže. Ta metoda je specifična za ovaj model robusnija i preciznija je od mnogih drugih metoda procjene pogleda jer za nju nije nužno potrebno da se na slici nalaze jasni nesakriveni dijelovi očiju i lica. Štoviše, oči mogu biti obstruirane (sakrivene nekim drugim predmetom) ili pak cijelo lice ne mora uopće biti vidljivo, nego osoba može biti okrenuta od kamere.

Zbog te činjenice, ova metoda agregiranja različitih razina povećanja slike stabilno procjenjuje smjer pogleda i u tim ekstremnijim slučajevima koji su kod primjene drugih metoda problematični.

Slika 2.1. *Multiple zoom* tehnika za obradu videa opisana u [1]



Slika 2.2. Arhitektura MSA (*Multiple Scale Aggregation*)

2.1.2. Arhitektura - model MSA (eng. *Multiple Scale Aggregation*)

U ovome radu korišten je model MSA (*Multiple Scale Aggregation*) koji se temelji na paralelnom učitavanju više slika i procjenjivanja smjera pogleda na temelju njih, a preuzeta je iz [1] i modificirana za primjenu na drugačijem skupu podataka. U nastavku slijedi detaljniji pregled arhitekture tog modela.

Backbone mreže

Na ulaz modela postavljamo slike te se one obrađuju u prvom dijelu mreže, takozvanom *backbone*-u. *Backbone* jest, u općenitom kontekstu terminologije neuronskih mreža, naziv za dio modela koji je odgovoran za ekstrakciju značajki iz ulaznih podataka - u ovom slučaju slika lica. Vrlo se često pojavljuje u primjenama računalnog vida gdje postoji detekcija ili klasifikacija predmeta.

U slučaju ovog modela, *backbone* je CNN (konvolucijska neuronska mreža) koja procesuira slijed povećanih slika te iz njih ekstrahira dvodimenzionalne značajke potrebne za daljnju obradu u ostatku modela.

Neke od varijanti CNN mreža koje su se koristile kao *backbone* u originalnom radu [1] su: *Squeezenet*, *Shufflenet_V2*, *Mobilenet_V2*,*Resnet18* [9] i *Hardnet68*.

Budući da je ekspresivnost mreže (koja je općenito proporcionalna broju parametara) u ovom slučaju bitan faktor pri treniranju i evaluaciji, *backbone* mreža mora biti pažljivo odabrana da bi svi detalji prisutni u višestrukim razinama povećanja ulaznih slika mogli biti obuhvaćeni i predstavljeni u značajkama na izlazu iz mreže.

Iz perspektive ekspresivnosti mreže, u evaluaciji iz originalnog rada je *Squeezenet* imao najveću kutnu pogrešku (eng. *angular error*) procijenjenog vektora pogleda, a *Hardnet68* najmanju. Ta pojava proizlazi iz činjenice da je *Hardnet68* kompleksniji model od *Squeezenet*-a, pa je mogao obuhvatiti više informacija iz ulaznih slika.

U ovome radu jest korišten *Resnet18* kao *backbone*.

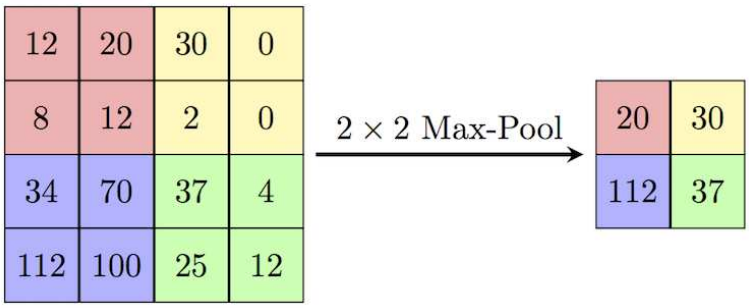
Feature mape (mape značajki)

Mape značajki (eng. *feature maps*) su dvodimenzionalne matrice neurona koje kvalitativno (brojevima) i prostorno (svojom pozicijom u arhitekturi) definiraju informacije o ciljanoj izlaznoj vrijednosti mreže. Imaju nezaobilaznu ulogu u neuronskim mrežama koje se primjenjuju na probleme računalnog vida.

Kod modela u ovome radu, svaka mapa značajki sakuplja informacije o slici na različitim razinama povećanja. Tim značajkama su paralelno sadržane informacije relevantne za procjenu pogleda iz slike (tj. slika), te se u narednom sloju daljnje sažimlju.

Ovaj dio arhitekture mreže jest neophodan za ekstrahiranje i sažimanje informacija o slici (primjerice lokacija i orijentacija očiju u odnosu na lice, gleda li osoba direktno prema kameri ili iz kuta, itd.). Nakon što ovaj dio obrade završi, informacije koje odgovaraju ključnim dijelovima slike su sažeto i informacijski gusto prikazani u izlazima neurona, tako da se na temelju njih može nesmetano zaključivati o smjeru pogleda u narednim (arhitekturalno gušćim) slojevima mreže.

Max pool sloj



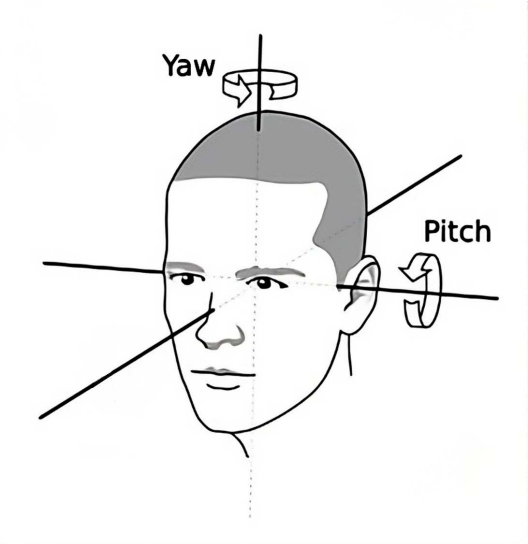
Slika 2.3. Primjer *Max pool* operacije

Pooling slojevi konvolucijskih neuronskih mreža služe za prostorno smanjenje dimenzija prijašnjih slojeva radi jednostavnije obrade u ostatku mreže, a čuvaju relevantne dijelove informacije i njihove međuodnose iz slojeva koje im prethode.

Mnoge vrste *pooling* slojeva postoje, primjerice *average pooling*, *max pooling*,*min pooling* itd. Na slici 2.3. je prikazan primjer *pooling* sloja koji prostorno sažima matricu 4 x 4 na matricu 2 x 2. To sažimanje se provodi tako da se svaka podmatrica (takozvana *jezgra*, eng. *kernel*) veličine 2 x 2 u ishodišnoj matrici preslikava na jednu jedinu vrijednost u odredišnoj matrici. Operacija koja se pritom koristi jest maksimum podmatrice (dok za druge *pooling* slojeve mogu biti primjenjene druge operacije, poput minimuma ili prosjeka vrijednosti podmatrice).

Pooling slojevi također imaju neophodnu ulogu u konvolucijskim neuronskim mrežama jer:

- čine postupak treniranja robusnijim na prostorne varijacije
- smanjuju komputacijski trošak treniranja mreže.



Slika 2.4. *Yaw* i *pitch* kutevi rotacije glave

Target sloj

Budući da je krajnji cilj treniranja mreže procjeniti kut gledanja subjekta, orijentaciju pogleda predstavljamo numerički, vektorom u prostoru. Orijentacija pogleda se može jednoznačno definirati pomoću polarnih koordinata, tj. kuteva eng. *yaw* (označeno s θ) i *pitch* (označeno s ϕ). U nastavku, na kraju poglavlja bit će riječ i o razlici između polarnih i kartezijevih koordinata u kontekstu treniranja mreže koja predviđa vrijednost vektora u tri dimenzije.

U zadnjem, ciljnom (eng. *target*) sloju na izlazu bismo u idealnom slučaju htjeli direktno dobiti procjene vrijednosti kuta *yaw* (θ_p, eng. *predicted yaw*) i *pitch* (ϕ_p, eng. *predicted pitch*), gdje se procjenjuju *ground truth* vrijednosti kuteva, θ_g i ϕ_g. Međutim, zbog poteškoća u treniranju mreže da procijeni te vrijednosti direktno, bilo je potrebno uvesti određene međukorake pri računanju te se umjesto iznosa kuteva *yaw* i *pitch* na izlazu zadnjeg sloja mreže procjenjuju vrijednosti **$\sin(\theta_p)$, $\cos(\theta_p)$ i $\sin(\phi_p)$**.

O detaljnim razlozima za uvođenje te substitucije za direktne vrijednosti kuteva *yaw* i *pitch* riječ je u nastavku.

2.1.3. Kutevi *yaw* (θ) i *pitch* (ϕ)

Budući da su se u treniranju mreže iz *Gaze360* [7] javljali problemi s velikim iznosima funkcije gubitka na dijelu domene kuta *yaw* (označen slovom θ) na granici između $\pi/2$

i $-\pi/2$ zbog nekontinuiteta u tom dijelu domene, u *Multi Zoom Gaze*-u [1] je razvijena metoda koja zaobilazi taj problem time što posredno evaluira θ, a ne neposredno.

Taj je problem riješen time što se ne predviđa direktno θ_p (*predicted yaw*), nego $sin(\theta_p)$ i $cos(\theta_p)$, i umjesto ϕ_p (*predicted pitch*) se predviđa $sin(\phi_p)$. Nakon toga se inverznim trigonometrijskim transformacijama (sin^{-1} i cos^{-1}) dobivaju θ_p i ϕ_p koje tražimo. Kao što je rečeno u prošlom poglavlju, podrazumijevamo da se na kraju mreže koju treniramo nalaze vrijednosti $sin(\theta_p)$, $cos(\theta_p)$ i $sin(\phi_p)$ koje potom koristimo da bismo izračunali θ_p i ϕ_p pomoću formula u nastavku.

Uvodimo oznake za dobivene vrijednosti sinusa i kosinusa kuta *yaw* te sinusa kuta *pitch* na izlazu iz mreže:

$$s_y = \sin(\theta_p) \tag{2.1}$$

$$c_y = \cos(\theta_p) \tag{2.2}$$

$$s_p = \sin(\phi_p) \tag{2.3}$$

kao eng. *sine of yaw*, *cosine of yaw*, i *sine of pitch*. *Yaw* (θ_p) možemo računati na dva načina.

- pomoću sinusa s_y i predznaka kosinusa c_y - što označavamo s θ_s
- pomoću kosinusa c_y i predznaka sinusa s_y - što označavamo s θ_c

Formule za θ_s i θ_c dane su u nastavku:

$$\theta_s = \begin{cases} \sin^{-1}(s_y) & \text{za } c_y \geq 0 \\ \pi - \sin^{-1}(s_y) & \text{za } c_y < 0 \end{cases} \tag{2.4}$$

$$\theta_c = \begin{cases} \cos^{-1}(c_y) & \text{za } s_y \geq 0 \\ -\cos^{-1}(c_y) & \text{za } s_y < 0 \end{cases} \tag{2.5}$$

Nakon što su θ_s i θ_p dobiveni, može ih se jedonstavno uprosječiti da bi se dobio traženi θ_p (*predicted yaw*), na sljedeći način:

$$\theta_p = (\theta_s + \theta_c)/2 \tag{2.6}$$

no to se u postupku treniranja pokazalo problematičnim pristupom za kuteve θ_p u iznosu od 0° i ± 90° [1]. Stoga se koristi nešto složenija shema uprosječivanja za računanje θ_p koje je robusnije s obzirom na vrijednost θ_g.

θ_p pomoću težinskog prosjeka θ_s i θ_c

Budući da je θ_s precizniji u području oko 0°, a θ_c oko ± 90° te da bi se izbjegli problemi u treniranju kada se θ_g nalazi u tim područjima, u radu [1] je predložena sljedeća težinska shema uprosječivanja:

$$\theta_p = w * \theta_s + (1 - w) * \theta_c \tag{2.7}$$

gdje je

$$w = |\cos((\theta_s + \theta_c)/2)| \tag{2.8}$$

Time se veća težina pridodaje θ_s kada je θ_g oko 0°, a θ_c kada je θ_g oko ± 90°. Za detaljniju diskusiju o odabiru w u jednadžbi 2.8 potrebno je pogledati [1].

Računanje kuta *pitch* (ϕ_p)

ϕ_p se trivijalno računa pomoću s_p inverznom trigonometrijskom funkcijom:

$$\phi_p = sin^{-1}(s_p) \tag{2.9}$$

budući da se slučaj kada je $\phi_p \in \{[-\pi, -\pi/2] \cup [\pi/2, \pi]\}$ zanemaruje zbog računanja orijentacije glave u polarnim koordinatama.

Polarne i kartezijeve koordinate

Iako i kartezijeve i polarne koordinate jednoznačno određuju vektor u prostoru, korištene su polarne koordinate za predstavljanje vektora pogleda između ostaloga radi pojednostavljivanja arhitekture mreže. Tako da se umjesto tri vrijednosti u kartezijevim koordinatama, (x,y,z) na kraju predviđaju samo dvije vrijednosti u polarnim koordinatama, (θ, ϕ).

S druge strane, budući da su *ground truth* vektori pogleda predstavljeni u kartezijevim koordinatama, za evaluaciju i računanje kutne pogreške (eng. *angular error*) je bilo

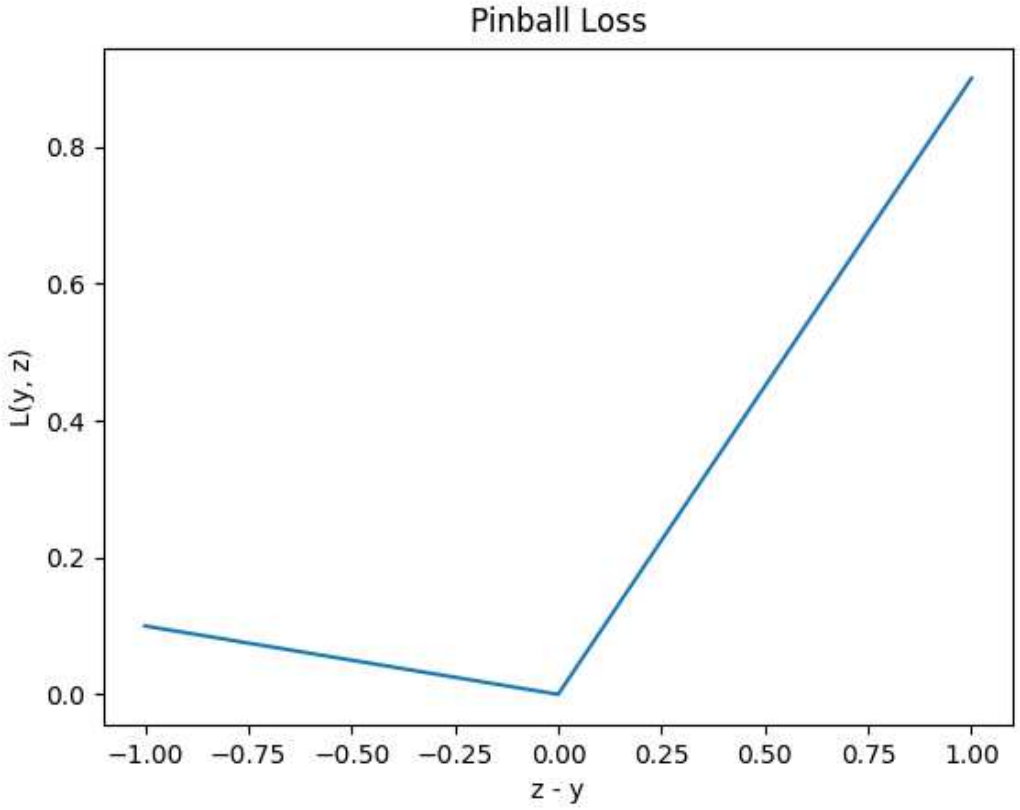
potrebno pretvoriti procjenjene polarne vektore u kartezijeve.

2.1.4. Funkcija gubitka

U [1] se koristila funkcija gubitka imena *pinball loss*. Ona ima karakteristiku da može različito vrednovati podcjenjivanje i precjenjivanje rezultata, ovisno o parametru funkcije τ.

$$L(y,z) = \begin{cases} (y-z)\cdot\tau, & \text{za } y \geq z \\ (z-y)(1-\tau), & \text{za } z > y \end{cases} \tag{2.10}$$

U slučaju treniranja modela je parametar funckije gubitka τ postavljen na vrijednost 0.1.



Slika 2.5. *Pinball loss* za $\tau = 0.1$

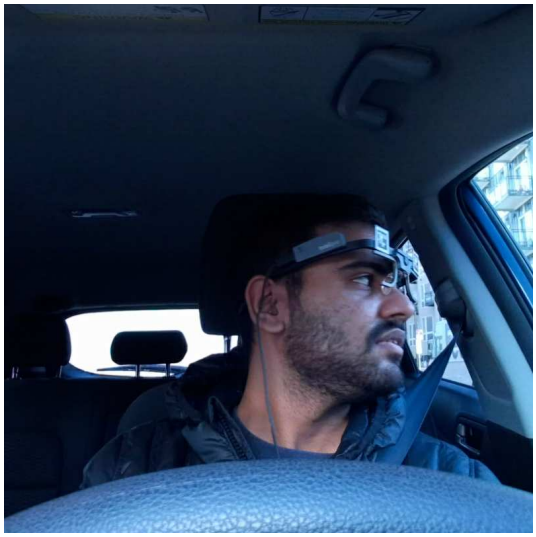
2.2. Skup podataka "Look Both Ways"

2.2.1. Opće informacije

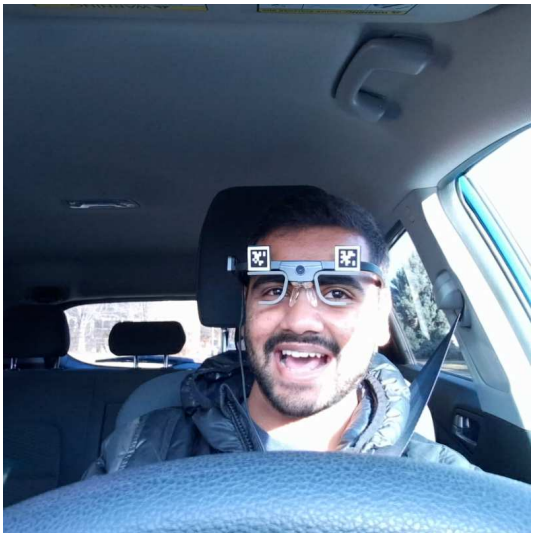
Skup podataka "Look Both Ways" je dobro strukturirani značeni skup slika vozača automobila iz 2022. godine. Sastoji se od videa, tj. sekvenci slika lica 28 različitih vozača u tijeku vožnje automobilom te *ground-truth* podacima o njihovom pogledu tijekom te slike, uz sliku i dubinsku mapu ceste na koju gledaju. Vožnja je snimana pri raznolikim uvjetima: doba dana, osvjetljenje, uvjeti na cesti itd., te ponekad i uz okluzije i različite izraze vozačeva lica.

Specifičnost ovog skupa podataka jest dvojaka:

- podaci su izravno javno dostupni bez bilo kakvih prijava i izdavanja dopusta za korištenje za razliku od mnogih ostalih (poput primjerice *Gaze360* [7], što u nekoj mjeri olakšava i ubrzava rad.
- Budući da skup podataka uključuje i slike prometa u kojemu se vozač nalazi tj. scena prometa u koju gleda, *in-domain ground truth* je vrlo konkretan i jasan. Kontekst vozačeva pogleda vrlo je važan i utoliko što olakšava proces eventualne ručne provjere točnosti.



(a) Subjekt 1: Umjereno osvjetljene ,ekstreman kut gledanja, relativno malen kontrast



(b) Subjekt 1: Dobro osvjetljenje, velik kontrast, izraz lica različit od neutralnog

Slika 2.6. "Look Both Ways": različiti uvjeti snimanja za subjekta br. 1

(a) Subjekt 14: Djelomična okluzija lica, neneutralan izraz lica, približno izravan kut gledanja

(b) Subjekt 14: Slabije osvjetljenje,veliko odstupanje u kutu gledanja

Slika 2.7. "Look Both Ways": različiti uvjeti snimanja za subjekta br. 14

Za svakog od ukupno 28 vozača postoji otprilike 10GB podataka u obliku slika vozačeva lica, slike vozačeve okoline, dubinske mape vozačeve okoline i podataka o smjeru pogleda. To odgovara 500-2000 *frame-ova* videa koji se koriste za treniranje modela. Iako su slike vozačeve okoline i njena pripadajuća dubinska mapa korisne za ručnu provjeru procjene smjera pogleda, u okviru ovog rada su ključne bile slika vozača i pripadajući *ground truth* smjera pogleda.

Isto tako, u najvećoj većini slučajeva pri treniranju i testiranju nije potrebno raditi razliku između smjera pogleda lijevog i desnog oka (vozač rijetko, ako ikada gleda "u križ" zbog bliskih predmeta gledanja), tako da se ta distinkcija ignorira iako postoji u oznakama u skupu podataka.

Uz skup podataka su dostupni i modeli za procjenu pogleda koje su autori trenirali pomoću njega, no druga metoda procjene pogleda je korištena u sklopu ovoga rada.

2.2.2. Podaci o pogledu vozača

U datotekama s podacima o pogledu vozača ima nekoliko različitih podataka. Postoje 2D i 3D podaci - 2D podaci se odnose na dvodimenzionalne vektore lokacije unutar koordinatnog sustava piksela same slike vozača (u domeni [0,800] x [0,800]), dok se 3D podaci odnose na trodimenzionalne vektore u prostoru gdje su smješteni kamere za snimanje

Slika 2.8. "Look Both Ways" - slike ceste (vozačeve okoline)

```
Gaze_Loc_2D: [325, 366]
Gaze_Loc_3D: [  0.20710606  -1.31861856 -16.32339582]
Right_Gaze_Dir: [ 0.01294137 -0.07808459 -0.99686274]
Right_2D_Eye_Loc: [379.487 427.686]
Right_3D_Eye_Loc: [-0.01325403  0.01097194  0.65075006]
Left_Gaze_Dir: [ 0.0091915  -0.07786406 -0.99692161]
Left_2D_Eye_Loc: [472.42  420.765]
Left_3D_Eye_Loc: [0.05065491 0.00672849 0.64550006]
```

Slika 2.9. Podaci o pogledu vozača (2D i 3D informacije)

vozača i scene te vozač i scena.

Sljedeći podaci su navedeni:

- *Gaze_Loc_2D* - dvodimenzionalna lokacija na slici gdje se čini da vozač gleda
- *Gaze_Loc_3D* - trodimenzionalna lokacija u prostoru gdje završava vozačev pogled
- *Right_Gaze_Dir* - trodimenzionalna vektor smjera pogleda vozačevog desnog oka
- *Right_2D_Eye_Loc* - dvodimenzionalna lokacija na slici gdje se nalazi vozačevo desno oko
- *Right_3D_Eye_Loc* - trodimenzionalna lokacija u prostoru gdje se nalazi vozačevo desno oko
- *Left_Gaze_Dir* - trodimenzionalna vektor smjera pogleda vozačevog lijevog oka
- *Left_2D_Eye_Loc* - dvodimenzionalna lokacija na slici gdje se nalazi vozačevo lijevo oko

- *Left_3D_Eye_Loc* - trodimenzionalna lokacija u prostoru gdje se nalazi vozačevo lijevo oko

Netočnosti u izvornim podacima

U određenom broju slučajeva su izvorne datoteke imale netočne podatke ili su im neki podaci nedostajali. Od ukupno 121 345 slika, njih 7 026 nisu imale definiran *Right_Gaze_Dir* i *Left_Gaze_Dir* te se stoga nisu mogle koristiti. Također, neke slike su imale krivo označene smjerove pogleda te su stoga mogle imati štetan utjecaj na samo treniranje.

2.3. Hardverska podrška

Mreža se trenirala na dvije grafičke kartice na jednom od poslužitelja zavoda ZESOI na FERu u Zagrebu. Radi se o dvjema karticama NVIDIA TITAN Xp te je 12GB radne memorije bilo dovoljno da se provede treniranje.

Priprema i testiranje koda za treniranje i analizu rezultata se izvršavala djelomično na hardveru laptopa modela Lenovo T440, no veći dio izračuna i analize rezultata je izvršen na poslužitelju fakulteta. Proces treniranja je pokrenut i na platformi *Google Colab*, no to se pokazalo nepraktičnim zbog veličine podataka za treniranje koja je iznosila oko 300GB.

2.4. Treniranje

2.4.1. Detaljnije o podacima

Podjela na *train*, *test* i *validation* skupove

Slike izvorno nisu podijeljene u skupove za treniranje, testiranje i validaciju, nego je bilo potrebno to napraviti naknadno. Od ukupno 74 skupova podataka s 24 različita vozača, 48 skupova podataka (65%) je upotrebljeno za treniranje, 14 (19%) za testiranje i 12 (16%) za validaciju. Potrebno je da se slike od jednog te istog vozača ne dijele na različite skupove nego da ostanu cjeloviti - u suprotnome bi bio prisutan rizik od pristranosti.

2.4.2. Podaci o pogledu

Od ukupnih informacija o smjeru pogleda su za treniranje korišteni trodimenzionalni vektori smjera pogleda lijevog i desnog oka (*Right Gaze Dir* i *Left Gaze Dir*). Prvo je razmatrana mogućnost da se oba vektora iskoriste tako da budu uprosječeni i normalizirani te je tako dobiveni vektor trebao biti iskorišten kao *ground truth* tijekom treniranja.

$$a = (right\_gaze\_dir + left\_gaze\_dir)/2 \tag{2.11}$$

$$ground\_truth\_gaze = \frac{a}{||a||} \tag{2.12}$$

S druge strane, taj pristup je doprinuo numeričkim nestabilnostima u računanju rezultantnog *ground_truth_gaze* vektora, najvjerojatnije jer je norma vektora a ($||a||$) bila dovoljno malena da se pojavljivala vrijednost *nan* tijekom dijeljenja.

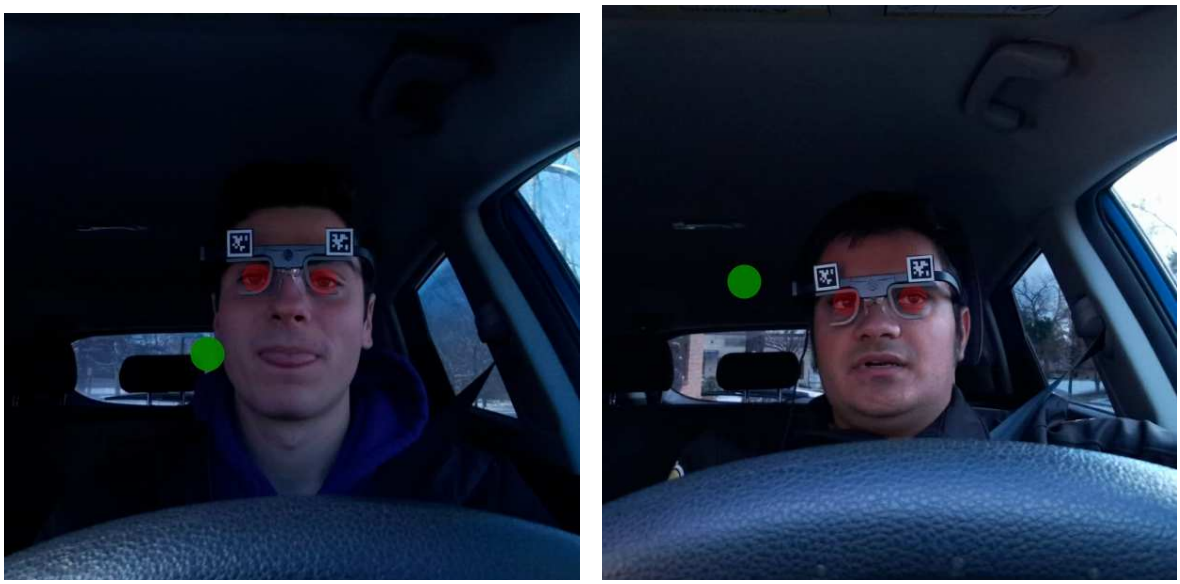
Na kraju je iskorišten najkraći i najjednostavniji pristup - koristio se direktno jedan od vektora smjera očiju: ***Right Gaze Dir***. Taj pristup je prihvatljiv jer su razlike između smjerova lijevog i desnog oka zanemarive pa nije narušena točnost podataka, a numeričke nestabilnosti se nisu pojavljivale kod tog pristupa.

S obzirom da se dimenzija slike u skupu podataka *Look Both Ways* (800 x 800 piksela) razlikuje u odnosu na skup podataka *Gaze360* (224 x 224 piksela), bilo je potrebno promijeniti dimenziju transformacije slika za treniranje i validaciju u odnosu na izvorni kod.

2.4.3. Problemi tijekom treniranja

Definicija *Ground truth*-a

Tijekom treniranja mreže su nastajali mnogobrojni problemi. Prvi od njih je bila definicija zadovoljavajućeg *ground_truth*-a koji bi točno odgovarao pogledu vozača na slici i ne bi dovodio do pojavljivanja *nan* vrijednosti u parametrima modela. Definicija koja je transformirala dvodimenzionalne podatke (*Right_2D_Eye_Loc*, *Left_2D_Eye_Loc* i *Gaze_Loc_2D*) u trodimenzionalne se pokazala vrlo nepreciznom jer *Gaze_Loc_2D* u velikom broju slu-

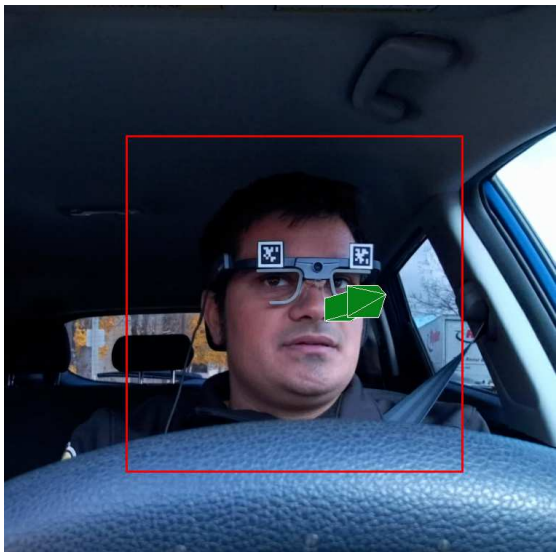


Slika 2.10. Vizualizacija nepreciznih 2D podataka iz skupa (crveno: oči vozača ; zeleno: netočni podatak o smjeru pogleda)

čajeva nije bio točan, tj. konzistentan sa smjerom pogleda vozača na slici.

Potom je slijedio pokušaj definicije *ground_truth*-a pomoću 2.11 i 2.12 koji su doveli do numeričkih nestabilnosti, tj. do pojavljivanja *nan* vrijednosti u težinama modela. Čim bi se u modelu počele pojavljivati *nan* težine, ubrzo bi model kao izlaz isto tako davao isključivo *nan* vrijednosti te je bilo potrebno naći alternativno rješenje.

Nadalje, budući da određen broj izvornih 3D podataka nije bio točan, proveden je hibridni pristup uspoređivanja 3D podataka iz dataseta i najboljeg modela iz rada [1] koji se pokazao relativno točnim u predviđanju stvarnog smjera pogleda u slikama iz ovog skupa podataka. U slučaju da je odstupanje *yaw* ili *pitch* kuteva između 3D podataka i predviđenog smjera pogleda bilo veće od 20%, te se slike nisu uzimale u obzir tijekom treniranja. Pokušaj uprosječivanja predviđenog smjera pogleda iz [1] rada i 3D podataka s ciljem korekcije tih netočnih 3d podataka na kraju nije dalo dobre rezultate, te je i taj pristup odbačen.



Slika 2.11. Primjer netočnog 3D podatka - smjer pogleda (zeleno) odgovara smjeru *Right_Gaze_Dir*

Na kraju se jednostavno rješenje uzimanja *Right_Gaze_Dir* kao *ground truth* smjera pogleda pokazalo najboljim.

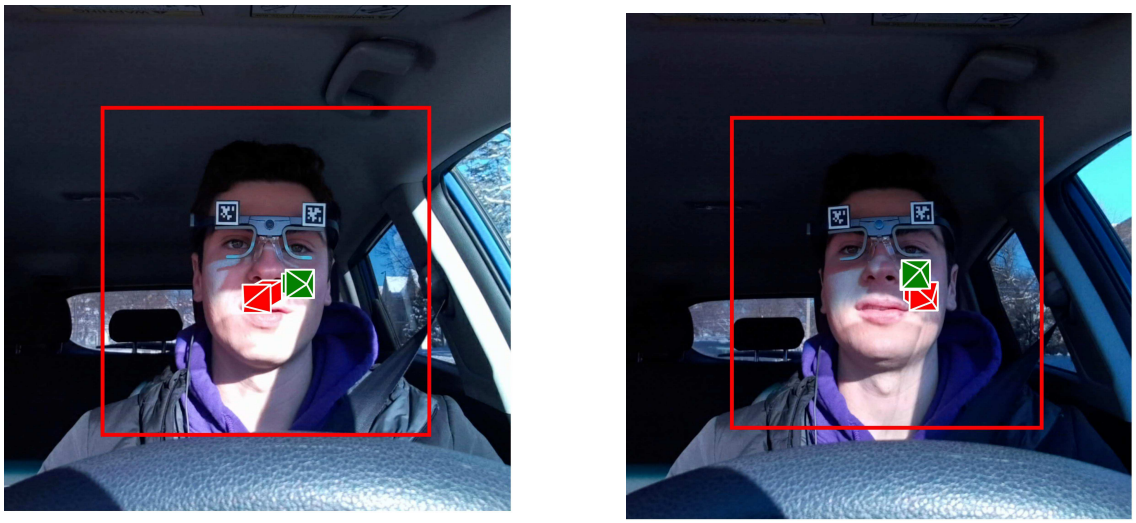
2.4.4. Hiperparametri

Tijekom treniranja je *batch size* ograničen na 4, broj epoha na 100 a stopa učenja na 0.0001. Veće brojke za *batch size* su rezultirale problemom manjka memorije, a broj epoha na kraju nije dostignuo brojku 100, nego je prekinut u epohi 14 jer su tada dobiveni zadovoljavajući rezultati. Varijacije u stopi učenja nisu imale značajniju ulogu u rezultatima treniranja.

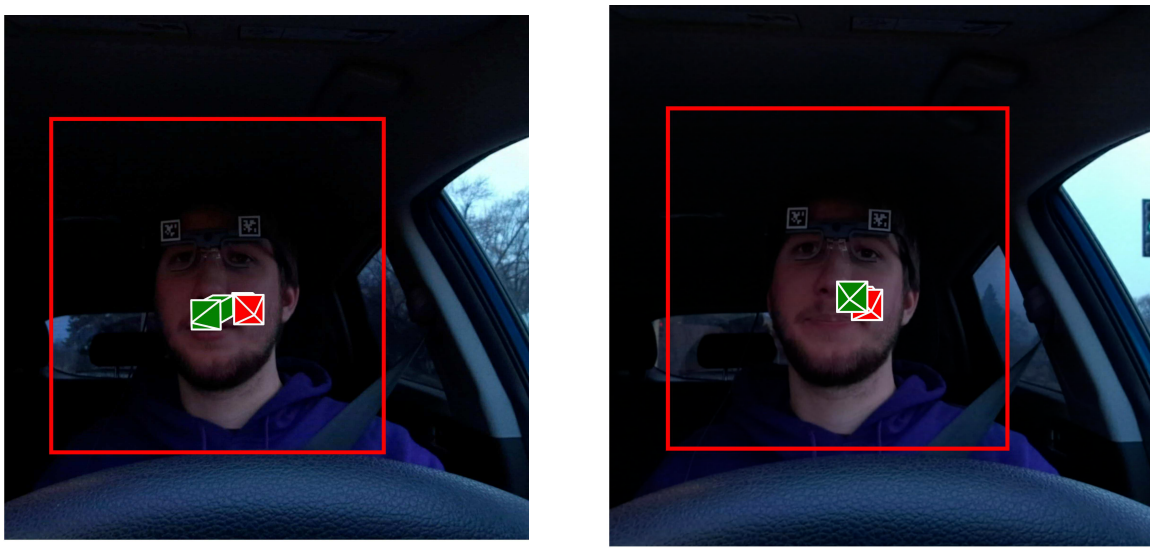
3. Rezultati

3.1. Vizualizacija smjera pogleda

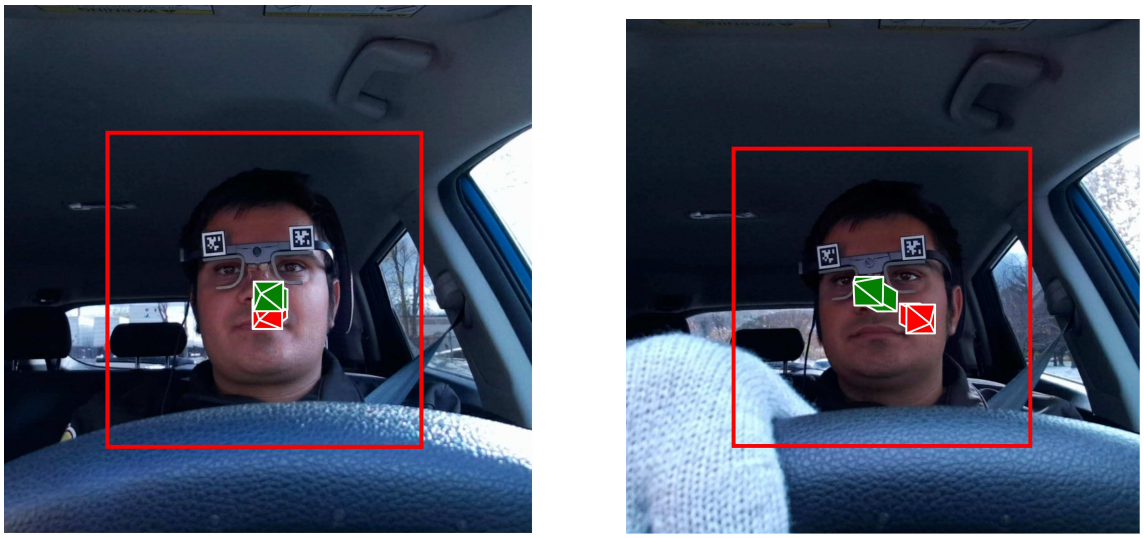
Slika 3.1. Rezultati treniranja (subjekt 19): crveno predviđeni smjer pogleda ; zeleno: *ground truth*



Slika 3.2. Rezultati treniranja (subjekt 19): crveno predviđeni smjer pogleda ; zeleno: *ground truth*



Slika 3.3. Rezultati treniranja (subjekt 22): crveno predviđeni smjer pogleda ; zeleno: *ground truth*



Slika 3.4. Rezultati treniranja (subjekt 23): crveno predviđeni smjer pogleda ; zeleno: *ground truth*

Oznake

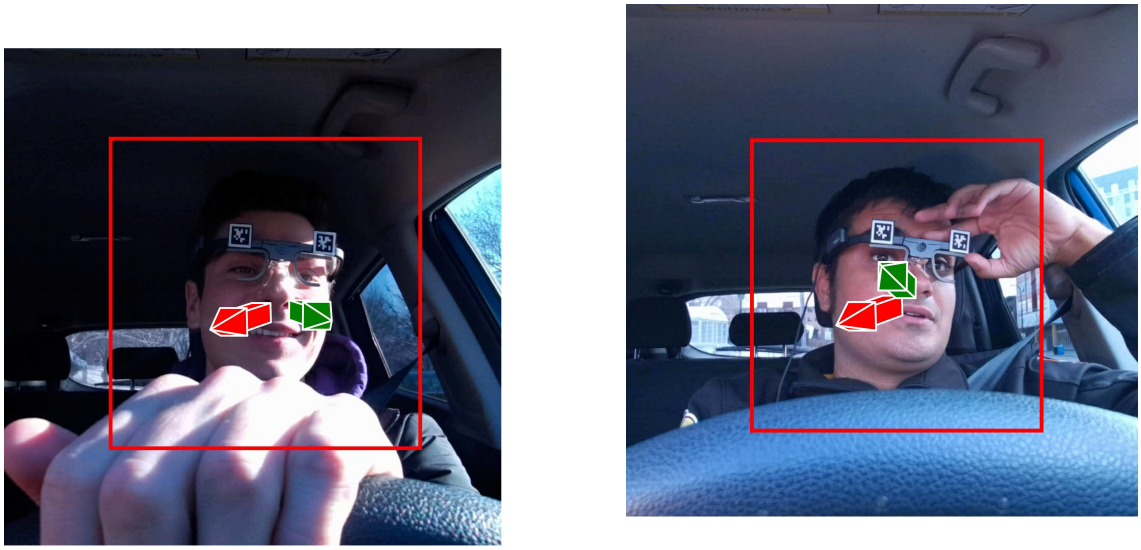
Zelena strelica označava *ground truth* smjer pogleda, a crvena predviđanje.

Crveni okvir oko vozačevih glava označava detetekciju lica vozača pomoću modela opisanog u [8] koja je potom proslijeđena mreži radi procjenjivanja pogleda.

3.1.1. Kvaliteta rezultata

Kao sto je vidljivo iz gornjih slika rezultata, postoji relativno veliko poklapanje između *ground truth* smjera pogleda i pogleda predviđenog istreniranim modelom. To je slučaj pogotovo kada vozač gleda direktno ispred sebe (yaw ∈ [-10 °, 10°], pitch ∈ [-7°, 7°]).

S druge strane, kad se radi o slikama s pogledima izvan tog intervala, predviđanje radi značajno lošije.



Slika 3.5. Lošiji rezultati treniranja (subjekti 19 i 23)

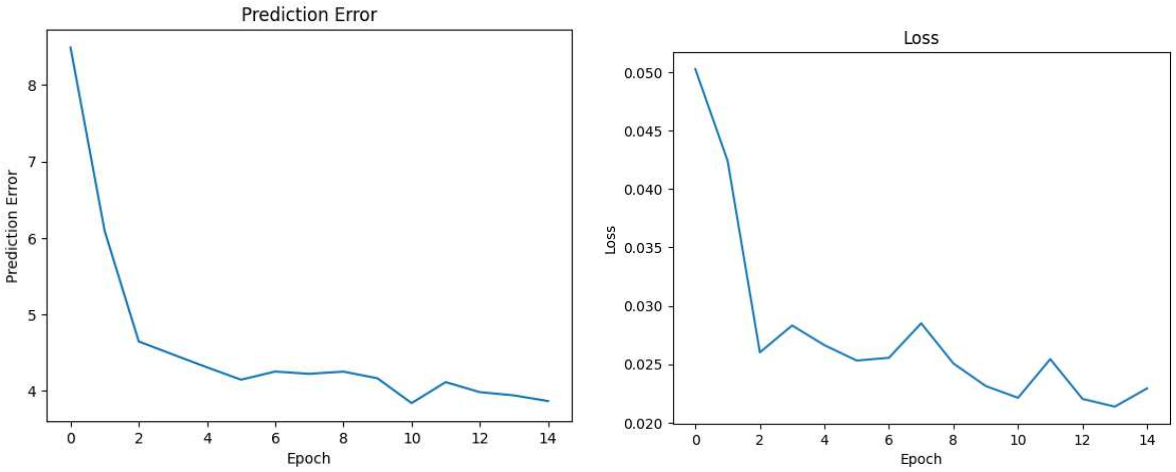
Mogući razlozi lošijih rezultata

To je moguće objasniti manjkom primjera s takvim kutevima gledanja u odnosu na kuteve iz tog intervala. Nadalje, nezanemariv broj slika s takvim kutevima ima netočne *ground truth* podatke te je to moglo dodatno pridonijeti grešci u predviđanju stvarnog kuta gledanja. Na kraju, jedna od mogućnosti za te greške jest nezadovoljavajuća arhitektura ili hiperparametri mreže - iako bi mreža trebala biti dovoljno fleksibilna da obuhvati i te rezultate kao što će biti vidljivo u sljedećem poglavlju gdje se uspoređuju rezultati najbolje mreže iz rada [1].

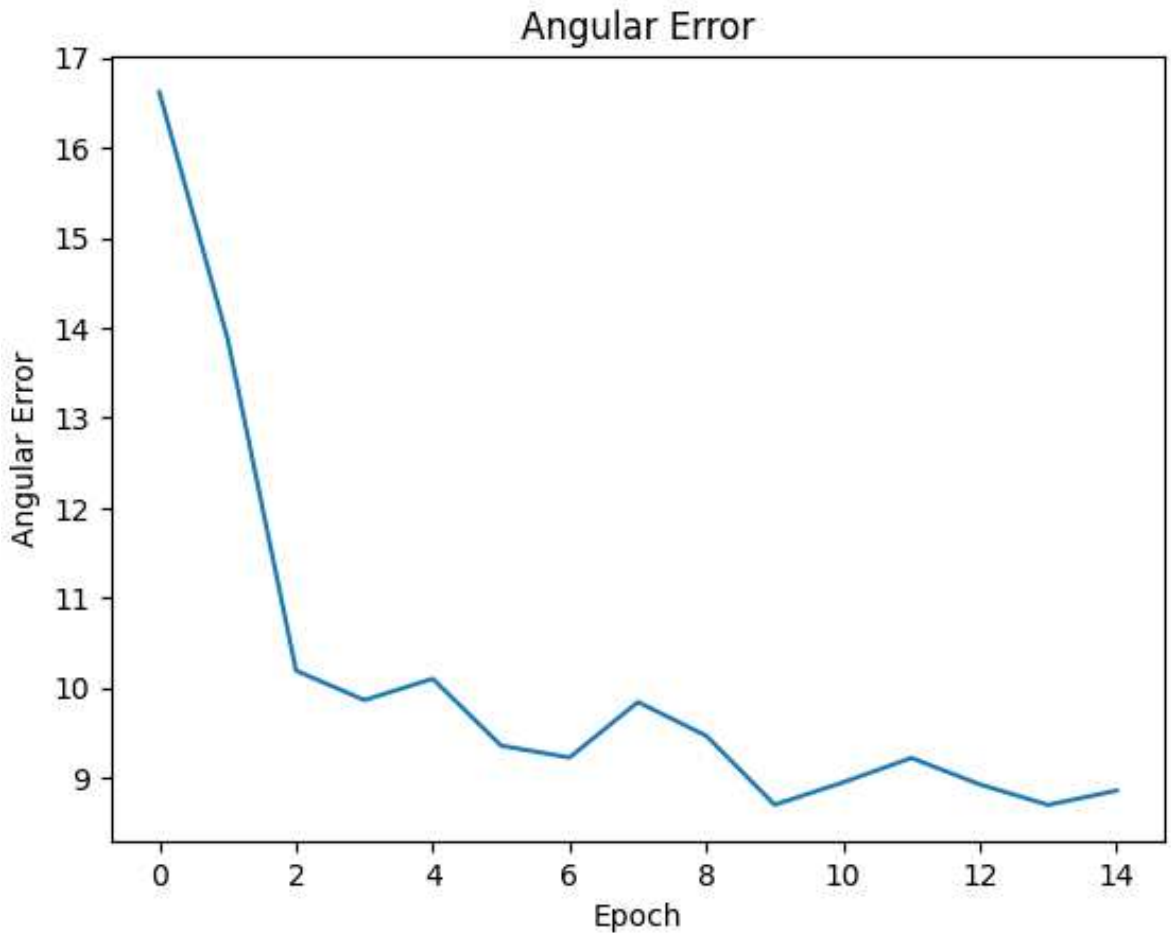
3.2. Metrike treniranja

Iz metrika treniranja vidljivo je da je treniranje u najvećoj mjeri uspješno. Funkcija gubitka *pinball loss*, kutna greška i greška predviđanja smanjuju se na karakterističan na-

čin s brojem epoha. Ukupno treniranje se nastavilo i nakon 14. epohe, no model je u toj epohi imao približno najbolju performansu gledano po veličini prosječne kutne pogreške na validacijskom skupu. Stoga su rezultati predviđanja generirani pomoću težina modela iz 14. epohe treniranja. Treniranje do te epohe trajalo je 10 sati.



Slika 3.6. Rezultati treniranja: metrike greške predviđanja i funkcije gubitka (*Pinball loss*)

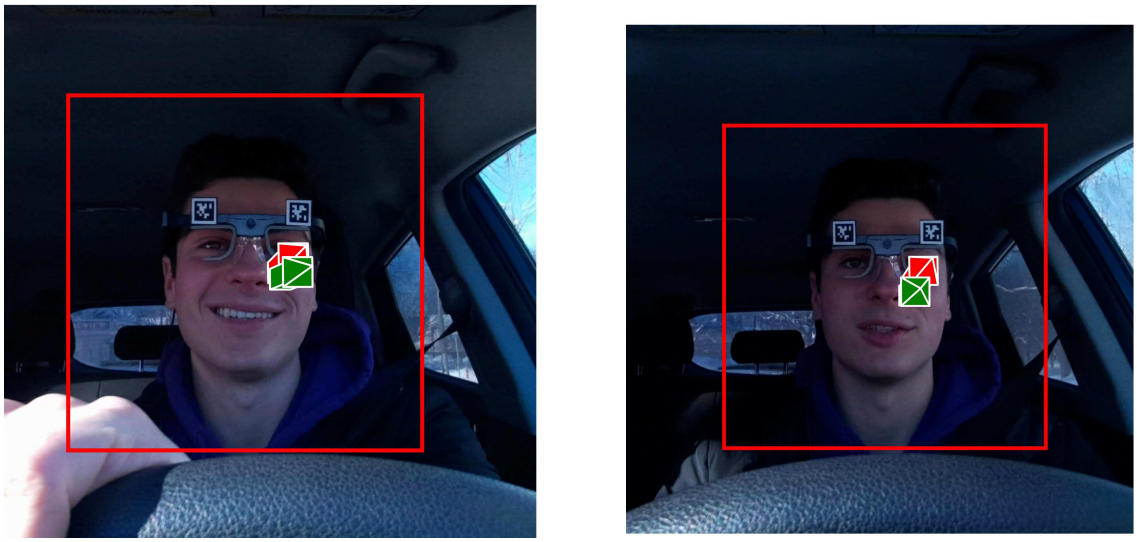


Slika 3.7. Rezultati treniranja: metrika kutne greške

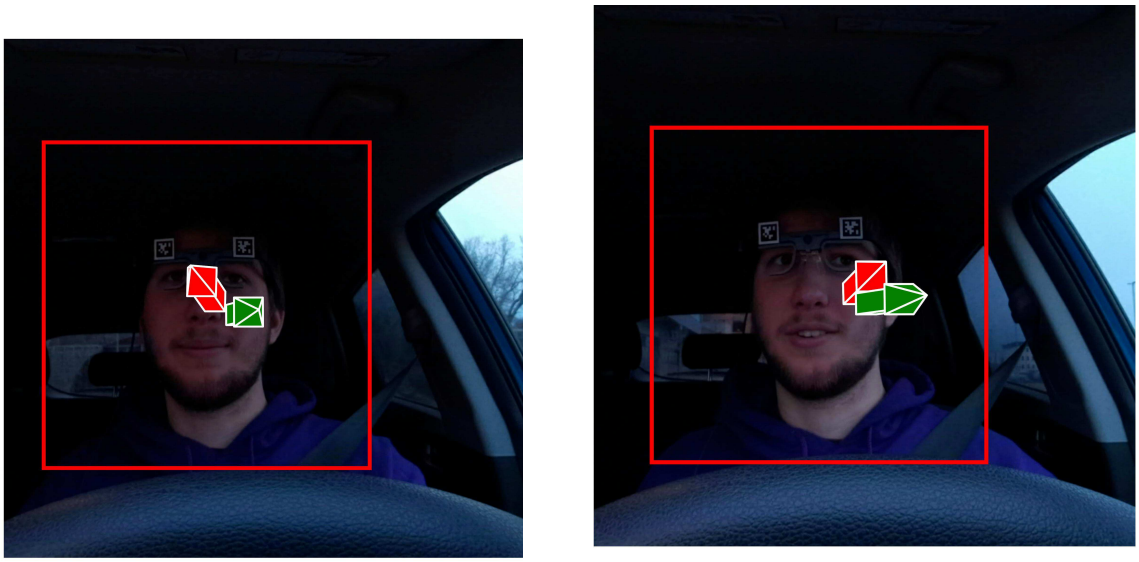
4. Diskusija

4.1. Usporedba s predviđanjem pogleda iz [1]

U nastavku ćemo prikazati i predviđanje smjera pogleda pomoću modela treniranog u *MultiZoomGaze* radu [1], te ćemo vizualno usporediti rezultate s predviđanjima modela iz ovoga rada.



Slika 4.1. Predviđanja pomoću [1] modela (subjekt 19)



Slika 4.2. Predviđanja pomoću [1] modela (subjekt 22)

Na slikama je jasno vidljivo da je model iz [1] rada postigao dobre rezultate u usporedbi s *ground truth*-om. Čak i kod ekstremnijih kuteva predviđanje je dovoljno vjerno da se ne radi o očitom promašaju. Slično kao i kod modela u ovome radu, predviđanje ima slabiju performansu kod ekstremnijih kuteva nego kod onih bliže ravnom pogledu.

Iako ovaj model ne generalizira izvrsno, ipak performira dovoljno dobro s obzirom da nije bio treniran na ovim slikama, nego na drugačijem skupu podataka.

4.2. Nedostaci provedene metode

Budući da su rezultati provedene metode i najboljeg modela iz rada [1] slične kvalitete, razlozi neidealne performanse obaju modela mogu se svesti na nekoliko mogućih kategorija.

Ekspresivnost modela

Teoretski je moguće da korišteni model nije bio dovoljno sofisticiran da obuhvati sve značajke na slikama vozača. Ipak, budući da se radi o modelu koji u sebi sadrži *Resnet 18* mrežu koja ima 11.5 milijuna parametara i može s visokom točnošću provoditi klasifikaciju 1000 različitih kategorija slika, ovo nije pretjerano realna mogućnost.

Proces treniranja

Da je treniranje provedeno na još više različitih varijacija u vrijednostima hiperparametara i da su skupovi za treniranje, testiranje i validaciju bili drugačije raspodijeljeni, postoji mogućnost da bi rezultati bili bolji. No s obzirom na arhitekturu i dostupan skup podataka, većina mogućnosti su u ovom smislu bile iscrpljene.

Primjerice, različita stopa za učenje te različiti *batch size*-evi nisu značajno utjecali na rezultate. S druge strane, skupovi za treniranje testiranje i validaciju su bili raspodijeljeni u pravilnom omjeru te su sadržavali dovoljno velik broj slika.

Podaci za treniranje

Na kraju, preostala mogućnost jest kvaliteta podataka za treniranje. Iz prošlih poglavlja je vidljivo da *ground truth* u nezanemarivom broju slučajeva nije bio zadovoljavajuć te je moguće da su te razlike dodatno unijele nesigurnost u predviđanju. To je zbog toga što ne postoji velika pravilnost u tim greškama koju bi model posljedično mogao "naučiti".

Da su podaci imali manje očitih pogrešaka (pogotovo u ekstremnijim kutevima), vjerojatno bi rezultati treniranja bili značajno kvalitetniji.

5. Budući rad

Budući da je dostupni skup podataka imao značajnu količinu netočnih smjerova pogleda u sebi, vjerojatno bi imalo smisla koristiti drugačiji skup podataka ili provesti sofisticiranu augumentaciju podataka tako da se utjecaj grešaka umanji.

Isto tako, u modelu [1] nalazi se i razni drugi modeli koji kombiniraju različite *backbone*-ove i mogu se iskoristiti da bi se dalje istražio potencijal procjene pogleda u kontekstu slika vozača.

Neki od boljih prijedloga za sljedeće radove uključuju modele iz radova [7] te [19]. Isto tako mogli bi se i kombinirati model iz [1] (kao što je to bio slučaj u ovome radu) samo s drugim skupom podataka tako da se vidi do koje mjere je skup podataka [2] kvalitetan u odnosu na ostale.

Također, uz istu kombinaciju modela i skupa podataka kao i u ovome radu, mogla bi se istražiti mogućnost sekvencijalne obrade slika, a ne samo statična obrada.

6. Zaključak

6.1. Sumacija metode

U okviru predviđanja smjera pogleda, korišteni model za predviđanje isti je kao i u radu [1] te je dovoljno ekspresivan da obuhvati kompleksnosti skupa treniranja dobivenog uz rad [2]. Uz minimalne modifikacije rada modela uspjela se uspostaviti mogućnost treniranja tog modela na podacima koji su imali drugačiju rezoluciju u odnosu na podatke na kojima je model originalno treniran.

Od mnoštva informacija što su bile dostupne u sklopu skupa podataka [2], korištene su isključivo informacije o trodimenzionalnom vektoru smjera pogleda, specifično desnog oka vozača radi jednostavnosti izračuna.

Kao najbolji model uzeta je instanca treniranja iz 14. epohe koja je imala dovoljno malenu grešku i dovoljno dobru performansu za usporedbu s drugim modelima sličnih kapaciteta.

6.2. Rezultati

Rezultati predviđanja smjera pogleda pomoću spomenutog modela zadovoljavajući su s obzirom da većina predviđenih pogleda nema veću devijaciju od *ground truth*-a veću od 20°, ako se radi o kutovima iz intervala yaw $\in$ [-10 °, 10°], pitch $\in$ [-7°, 7°]. S druge strane, kod kuteva izvan tog intervala nisu dale dobre rezultate i imale su velike devijacije u odnosu na *ground truth*.

Jedna od bitnijih činjenica jest to što skup podataka ima nezanemariv broj netočnih *ground truth* podataka za ekstremne kuteve gledanja pa je to u nekoj mjeri sigurno utjecalo na rezultate predviđanja.

Sažetak

Procjena smjera pogleda iz slika lica korištenjem dubokog učenja

Jakov Gracin

Procjena smjera pogleda na temelju lica osobe važna je primjena računalnog vida u posljednja dva desetljeća u autoindustriji i šire, prvenstveno zbog porasta količine slika i videa na koja se svakodnevno generira pomoću kamera visoke rezolucije. U okviru ovoga rada koristi se duboka neuronska mreža temeljena na mreži Resnet18 koja analizira slike tijekom vožnje automobila i na temelju njih procjenjuje trodimenzionalni vektor smjera pogleda vozača. Istražuje se točnost predviđanja smjera pogleda u odnosu na neuronsku mreže iz ostalih radova te se prilagođavaju informacije o pogledu u skupu dostupnih slika da bi predviđanje bilo dovoljno precizno za daljnje primjene u polju računalnog vida.

Ključne riječi: Računalni vid, procjena smjera pogleda, duboke neuronske mreže, ResNet18, analiza slike, automobilska industrija

Abstract

Gaze estimation from face images using deep learning

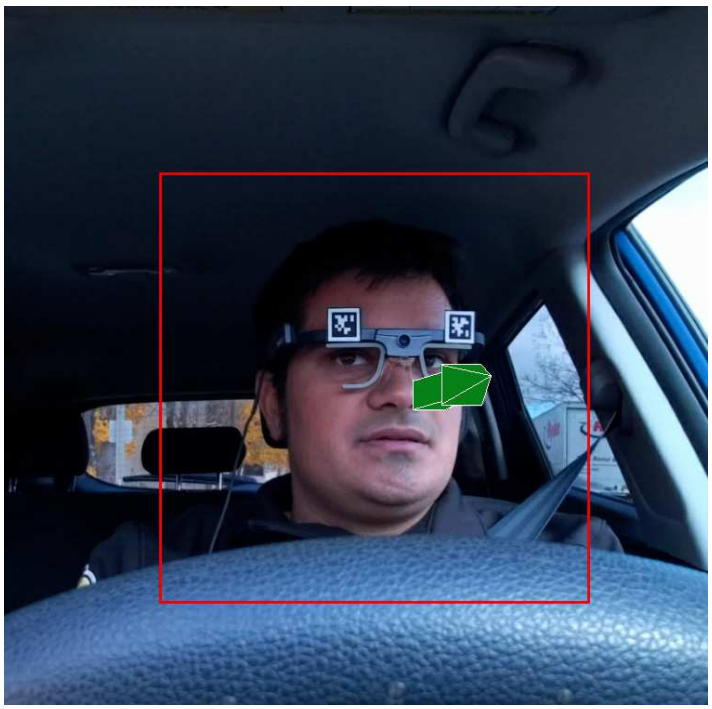
Jakov Gracin

Estimating the direction of gaze based on a person's face is an important application of computer vision in the last two decades in the automobile industry, primarily due to the increase in the amount of images and videos generated daily using high-resolution cameras. This paper utilizes a deep neural network based on the ResNet18 architecture, which analyzes images captured while driving a car and estimates the three-dimensional vector of the driver's gaze direction based on these images. The accuracy of gaze direction predictions is compared to neural networks from other studies, and the gaze information is adapted within the available image dataset to ensure that the predictions are sufficiently accurate for further applications in the field of computer vision.

Keywords: Computer vision, gaze estimation, deep neural network, ResNet18, image analysis, automobile industry

Privitak A: Programski kod

A1. Generiranje slike strelice pomoću *pyplot* paketa



Slika A1. Primjer vizauliziranog smjera pogleda pomoću strelice

```
from mpl_toolkits.mplot3d.art3d import Poly3DCollection
import matplotlib.pyplot as plt
import numpy as np
import math

import matplotlib.transforms as mtransforms

def rad_to_deg(alpha_rad):
  return math.floor(alpha_rad / math.pi / 2 * 360)
```

```
def deg_to_rad(alpha_deg):
  return alpha_deg / 360 * 2 * math.pi

'''
Rotates a 3D point around the center_of_rotation (also in 3D)
by the angle pitch and yaw
'''
def rotate(point, center_of_rotation, pitch=0, yaw=0):
  # pitch and yaw are in radians
  x1,y1,z1 = np.array(point) - np.array(center_of_rotation)
  a = math.sqrt(y1*y1 + z1*z1)

  #rotate for pitch
  alpha0 = 0
  if(z1 >= 0):
    alpha0 = math.acos(y1 / a)
  else:
    alpha0 = 2 * math.pi - math.acos(y1 / a)

  x2 = x1
  y2 = math.cos(alpha0 - pitch) * a
  z2 = math.sin(alpha0 - pitch) * a

  #rotate for yaw
  a = math.sqrt(x2*x2 + z2*z2)
  beta0 = 0
  if(z2 >= 0):
    beta0 = math.acos(x2 / a)
  else:
    beta0 = 2 * math.pi - math.acos(x2 / a)
```

```
    x3 = math.cos(beta0 + yaw) * a
    y3 = y2
    z3 = math.sin(beta0 + yaw) * a

    cx, cy, cz = center_of_rotation
    return [x3 + cx,y3 + cy,z3 + cz]

'''
Second-level utility function for offsetting the Axis object
See 'offset_ax' below
'''
def offset_ax_by_percentage(x_offset_percent, y_offset_percent,
    ax1, ax2):
    #ax1
    ax1_pos_bb = ax1.get_position()
    ax1_x0 = ax1_pos_bb.x0
    ax1_x1 = ax1_pos_bb.x1
    ax1_y0 = ax1_pos_bb.y0
    ax1_y1 = ax1_pos_bb.y1

    ax1_w = ax1_x1 - ax1_x0
    ax1_h = ax1_y1 - ax1_y0

    #ax2
    ax2_pos_bb = ax2.get_position()
    ax2_x0 = ax2_pos_bb.x0
    ax2_x1 = ax2_pos_bb.x1
    ax2_y0 = ax2_pos_bb.y0
    ax2_y1 = ax2_pos_bb.y1

    ax2_w = ax2_x1 - ax2_x0
    ax2_h = ax2_y1 - ax2_y0
```

```
    ax2.set_position(pos=[ax2_x0 + x_offset_percent * ax1_w,
      ax2_y0 + y_offset_percent * ax1_h, ax2_w, ax2_h])



'''
Utility function for offsetting the Axis object

Offset gaze arrow defined by axis 'ax' by centering it
around the bbox in the image.
The arrow is further shifted in the direction that it points to,
according to the parameters yaw and pitch
'''
def offset_ax(img, bbox, yaw_rad, pitch_rad, ax1, ax2): ##
    #unpacking img and bbox dimensions and coordinates
    img_y, img_x, _ = img.shape

    bbox_y0 = bbox[0]
    bbox_x0 = bbox[1]
    bbox_h = bbox[2]
    bbox_w = bbox[3]

    yaw_offset = 0.2 * bbox_w * math.sin(yaw_rad)
    pitch_offset = 0.2 * bbox_h * math.sin(pitch_rad)

    arrow_offset = 50
    x_offset_percent = -(img_x/2 - (bbox_x0 + bbox_w * 0.5) \
      - arrow_offset + yaw_offset) / img_x # offset in direction right
    y_offset_percent = (img_y/2 - (bbox_y0 + bbox_h * 0.5) \
      + pitch_offset) / img_y  # offset in direction up
```

```
    offset_ax_by_percentage(x_offset_percent, y_offset_percent, ax1, ax2)


'''
Displays the arrow in the gaze direction, defined by the
angles yaw and pitch (in radians)
Axis object is optional
'''
def display_arrow(yaw_rad, pitch_rad, ax=False):
    if(not ax):
        fig = plt.figure()
        ax = fig.add_subplot(111, projection='3d')

    h1 = 1
    h2 = 6
    h3 = 10
    v0 = [ [0.5,-0.5,h1],[0.5,0.5,h1],[-0.5,-0.5,h1],[-0.5,0.5,h1], \
            [0.5,-0.5,h2],[0.5,0.5,h2],[-0.5,-0.5,h2],
            [-0.5,0.5,h2],  [0.7,-0.7,h2],[0.7,0.7,h2],
            [-0.7,-0.7,h2],[-0.7,0.7,h2],    [0,0,h3]    ]
    v1 = []
    for point in v0:
        v1.append(rotate(point, [0,0,0], pitch_rad, yaw_rad))

    f = [ [0,1,3,2],  [0,1,5,4], [1,3,7,5], [2,3,7,6],
    [0,2,6,4],  [4,5,9,8]   ,[5,7,11,9], [6,7,11,10],
    [4,6,10,8],    [8,9,12], [9,11,12], [11,10,12], [8,10,12]]

    verts1 =  [ [ v1[i] for i in p] for p in f]

    ax.add_collection3d(Poly3DCollection(verts1,color='green',
    ls='-', edgecolor='white'))
```

```
ax.set_xlabel('X')
ax.set_xlim3d(-7, 9)
ax.set_ylabel('Y')
ax.set_ylim3d(-7, 9)
ax.set_zlabel('Z')
ax.set_zlim3d(0, 14)


ax.set_axis_off()
ax.view_init(90,-90,0)
```

Literatura

[1] Ashesh, Chen, Lin, "360-Degree Gaze Estimation in the Wild Using Multiple Zoom Scales", https://arxiv.org/abs/2009.06924

[2] Kasahara, Isaac and Stent, Simon and Park, Hyun Soo, "Look Both Ways: Self-Supervising Driver Gaze Estimation and Road Scene Saliency", In Proceedings of the European Conference on Computer Vision (ECCV), 2022.

[3] yihua c, baoyiwei, lufeng, "Appearance-based Gaze Estimation With Deep Learning: A Review and Benchmark", https://arxiv.org/abs/2104.12668

[4] Shahabeddin Khalighy, Graham Green, Christoph Scheepers, Craig Whittet, "Quantifying the qualities of aesthetics in product design using eye-tracking technology", International Journal of Industrial Ergonomics, 2015. https://doi.org/10.1016/j.ergon.2015.05.011

[5] Laurence R. Young, David Sheena, "Survey of eye movement recording methods", 1974. https://doi.org/10.3758/BF03201553

[6] Pavan Kumar Sharma, Pranamesh Chakraborty, "A Review of Driver Gaze Estimation and Application in Gaze Behavior Understanding", 2023. https://doi.org/10.48550/arXiv.2307.01470

[7] Petr Kellnhofer, Adria Recasens, Simon Stent, Wojciech Matusik, Antonio Torralba, "Gaze360: Physically Unconstrained Gaze Estimation in the Wild", 2019. https://arxiv.org/abs/1910.10088

[8] Adrnabdhar, "YOLOv8-Face-Detection", https://huggingface.co/arnabdhar/YOLOv8-Face-Detection

[9] Pandey, Gaurav and Srivastava, Sumit, “ResNet-18 comparative analysis of various activation functions for image classification”, In Proceedings of the 2023 International Conference on Information Communication Technology, 2023. 10.1109/ICICT57646.2023.10134464

[10] Robinson, David A., “A Method of Measuring Eye Movement Using a Scleral Search Coil in a Magnetic Field”, IEEE Transactions on Bio-medical Electronics, 1963. 10.1109/TBMEL.1963.4322822

[11] Pai, Yun Suen and Dingler, Tilman and Kunze, Kai, “Assessing hands-free interactions for VR using eye gaze and electromyography”, Virtual Reality, 2019. 10.1007/s10055-018-0371-2

[12] Gheorghe, Consuela-Madalina et al, “Using eye-tracking technology in Neuromarketing”, Romanian Journal of Ophthalmology, 2023. 10.22336/rjo.2023.2

[13] Alberto Gimeno, “Eye tracking heatmap in UX”, https://medium.com/@gimenete/built-in-eye-tracking-the-next-dimension-in-ux-b275ad3f3b0b

[14] Multi Theft Auto Wiki, “Rotation axes of the head”, https://wiki.multitheftauto.com/images/6/6b/The-yaw-pitch-and-roll-angles-in-the-human-head-motion-11.png

[15] Papers with Code, “Max pooling”, https://production-media.paperswithcode.com/methods/MaxpoolSample2.png

[16] Ashesh, Chen, Lin, “Supplemental: 360-Degree Gaze Estimation in the Wild Using Multiple Zoom Scales”, https://arxiv.org/abs/2009.06924

[17] Pai, Yun Suen and Dingler, Tilman and Kunze, Kai, “Assessing hands-free interactions for VR using eye gaze and electromyography - image”, Virtual Reality, 2019. 10.1007/s10055-018-0371-2

[18] Domański, Marek and Dziembowski, Adrian and Grajek, Tomasz and Grzelka, Adam and Kurc, Maciej and Łuczak, Adam and Mieloch, Dawid and Siast, Jakub and Stankiewicz, Olgierd and Wegner, Krzysztof, “[FTV AHG] Video and

depth multiview test sequences acquired with circular camera arrangement – “Poznan Service” and “Poznan People””, ISO/IEC JTC1/SC29/WG11, MPEG2015, Doc. M36569, 2015.

[19] Fischer, Tobias and Chang, Hyung Jin and Demiris, Yiannis, “RT-GENE: Real-Time Eye Gaze Estimation in Natural Environments”, In Proceedings of the European Conference on Computer Vision (ECCV), September 2018.