Prepoznavanje gesta dlana u svrhu simuliranog upravljanja zrakoplova

Daniel Košmerl
FER
Zagreb, Hrvatska
daniel.kosmerl@fer.hr

Jakov Jakovac
FER
Zagreb, Hrvatska
jakov.jakovac@fer.hr

Nikola Perić
FER
Zagreb, Hrvatska
nikola.peric@fer.hr

Domagoj Marić
FER
Zagreb, Hrvatska
domagoj.maric2@fer.hr

Leonarda Pribanić
FER
Zagreb, Hrvatska
leonarda.pribanic@fer.hr

Tomislav Matanović
FER
Zagreb, Hrvatska
tomislav.matanovic@fer.hr

I. Uvod

Razvoj umjetne inteligencije i specifično računalnog vida omogućuje sve šire primjene u različitim industrijama - od medicine i sigurnosti do videoigara gdje nove tehnologije otvaraju mogućnosti za inovativne i interaktivne oblike zabave u slobodno vrijeme. U ovom projektu, implementirana je simulacija zrakoplova kojeg korisnik upravlja pomoću dlana. Temelji se na analizi i obradi vizualnih podataka prikupljenih putem kamere korištenjem modela dubokog učenja za klasifikaciju i interpretaciju različitih gesti dlana. Model identificira specifične geste te šalje odgovarajuće naredbe u simulacijsko okruženje za upravljanje zrakoplovom.

II. Pregled korištenih metoda za prepoznavanje i klasifikaciju geste dlana

Glavni problem predstavlja prepoznavanje pozicije dlana sa snimke te detekcija u kojem se on položaju nalazi (zatvorena/otvorena šaka, kut koji zatvara…). U ovom poglavlju dan je pregled korištenih metoda koja služe kako bi se prepoznao dlan na slici i klasificirala klasa kojoj gesta pripada u stvarnom vremenu.

A. Konvolucijske neuronske mreže

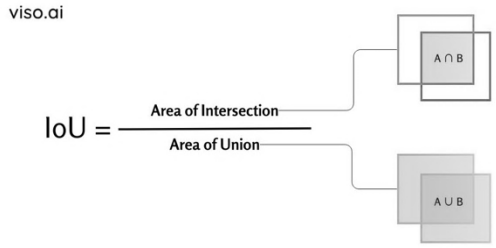
Kod običnih neuronskih mreža, svaki ulaz povezan je sa svakim neuronom u skrivenom sloju mreže. Problem nastaje kod analize slika i videozapisa, jer bi tada svaki piksel bio jedan ulaz u neuronsku mrežu. Zbog toga se koriste konvolucijske neuronske mreže (CNN) koje imaju poseban konvolucijski sloj čiji je zadatak smanjiti broj ulaza u potpuno povezani sloj neuronske mreže u kojem se odvija predikcija. CNN-ovi tipično imaju dvije faze: (1) generiranje graničnih okvira koji lociraju potencijalne objekte u slici i (2) klasifikacija pronađenih objekata. Problem dviju odvojenih faza leži u njihovoj računalnoj zahtjevnosti i vremenskoj neefikasnosti, što postaje posebno izraženo kod sustava s velikim bazama podataka ili u vremenski osjetljivim scenarijima, poput simulacije upravljanja zrakoplovom.

B. YOLO algoritam

Za razliku od običnih CNN-ova, YOLO (You Only Look Once) ima samo jednu fazu koja simultano generira granične okvire i klasificira objekte u njima. Algoritam podijeli ulaznu sliku u N x N ćelija. Svaka ćelija predviđa granične okvire objekata u njoj, a ako je središte objekta u njoj, klasificira i objekt. S obzirom na to da metoda procesira cijelu sliku u jednom prolazu, računalno je puno efikasnija od standardnih CNN-ova. No, to dolazi pod cijenu: algoritam je generalno manje precizan i postiže značajno lošije rezultate kod detekcije malih objekata. U slučaju ovog projekta, to je prihvatljivo jer će jedini objekt od interesa na slici biti dlan. S druge strane, jednostavna arhitektura YOLO algoritma, koja omogućuje obradu slika u stvarnom vremenu, vrlo je važna kako bi kontrola nad simuliranim zrakoplovom bila intuitivna i pružila korisniku osjećaj glatkog upravljanja bez kašnjenja.

C. Presjek kroz Unija (IoU)

Kako bi sustav klasifikacije gesti dlana bio učinkovit i efikasan u stvarnom vremenu, potrebno je precizno ocijeniti koliko dobro predviđeni granični okvir odgovara stvarnoj poziciji dlana. Presjek kroz Unija (engl. Intersection over Union) ili IoU ključna je metrika za evaluaciju performansi modela detekcije objekata. Kao što je vidljivo na slici 1, ona predstavlja omjer između preklapanja predviđenog i stvarnog graničnog okvira objekta.



Slika 1

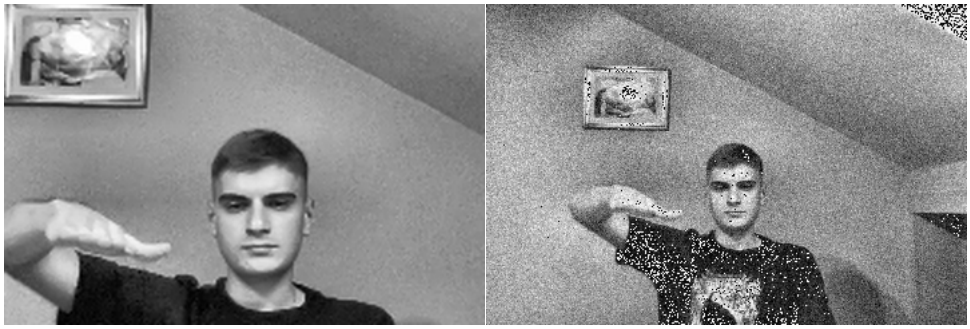
Matematički, to je omjer površine presjeka i unije predviđenog i stvarnog okvira. Prihvatljive vrijednosti su tipično iznad 0.5 tj 50%, dok su jako dobre vrijednosti iznad 70%. IoU osigurava da je detekcija precizna i pouzdana, što je ključno za aplikacije u stvarnom vremenu kao što je simulirano upravljanje zrakoplovom.

III. Implementacija programskog rješenja

Prije same implementacije potrebno je stvoriti bazu podataka nad kojim će se mreža učiti. S obzirom na

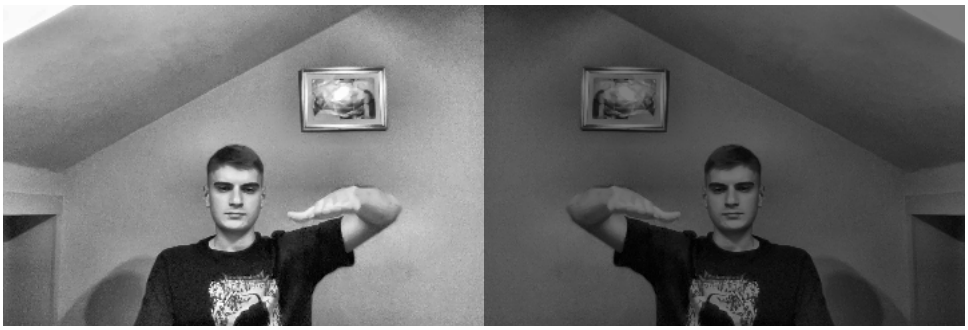
specifičnost zadatka, nema postojećih javno dostupnih baza podataka zbog čega je stvorena nova. Svaki član grupe snimio je minutu videosnimke koja ja programski isječena na 100 slika. Zatim je manualno označen pravi granični okvir dlana i klasa kojoj gesta pripada. Definirane su četiri klase: (1) PAUSE – otvoreni dlan; služi za pauziranje igre, (2) NOPALM – nema dlana na slici, (3) FLY – dlan prema dolje sa sklopljenim prstima i (4) UNDEFINED – neželjene geste koje nemaju odgovarajuće naredbe u simulaciji. Za upravljanje zrakoplovom najvažnija je klasa FLY koja položajem tj rotacijom dlana određuje smjer kretanja zrakoplova u simulaciji. Kut dlana također utječe na kontrolu zrakoplova – veći kut znači da će se zrakoplov brže kretati u tom smjeru. Kako bi se dodatno povećao skup podataka za učenje mreže, primijenjene su razne augmentacije nad postojećim skupom. Podacima je dodan Gaussov šum, izmijenjena svjetlina slika, zumiranje slika, te su transformirane da budu zrcalne slike originalnih. Na slici 2 prikazan je originalni primjer iz baze podataka i četiri primjera stvorena augmentiranjem originalnog primjera.

Originalna slika



Zumirana slika

Gassov šum



Zrcalna slika

Smanjenje svjetline

Slika 2

Nakon stvaranja baze podataka, trenira se neuronska mreža temeljena na YOLOv1 algoritmu. Mreža pomoću konvolucijskih slojeva izdvaja važne značajke iz slika nakon čega predviđa vjerojatnosti klasa gesti i koordinate graničnih okvira tj. poziciju dlana na slici. Izlaz mreže je lista od osam elemenata. Prva četiri elementa predstavljaju vjerojatnost da slika pripada svakoj od četiri klase, a sljedeća četiri određuju poziciju graničnog okvira: x i y

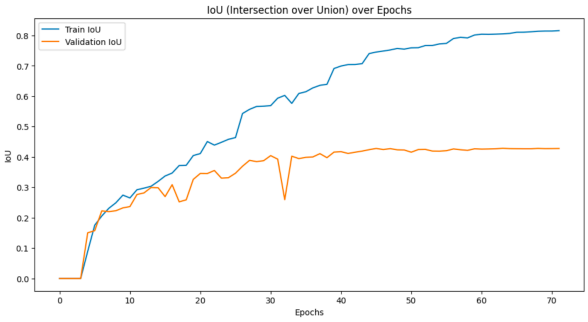
koordinate te širinu i visinu okvira. Na slici 3 dan je primjer jednog izlaza mreže zajedno sa predviđenim i stvarnim graničnim okvirom na slici.

Slika 3

Ako slika pripada klasi FLY, potrebno je dodatno odrediti kut dlana kako bi bilo moguće odrediti smjer i brzinu kretanja zrakoplova. Za detekciju nagiba dlana potrebne su koordinate palca i malog prsta. Da bi se to dobilo, koristi se već trenirani Googleov MediaPipe okvir koji iz slike us stvarnom vremenu može detektirati dijelove šake. Pomoću koordinata palca i malog prsta, dobiva se novi vektor čiji je nagib približno jednak nagibu dlana.

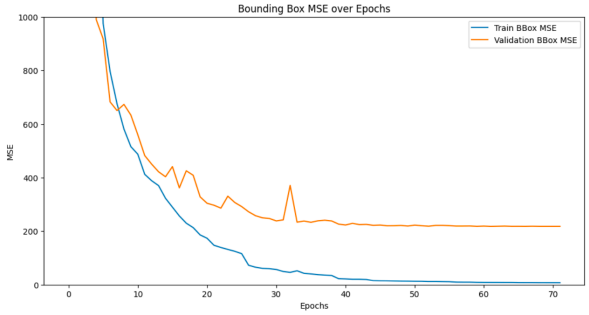
IV. Rezultati

Kako bi se analizirale performanse modela, korištene su različite mjere kao što su IoU, srednja kvadratna pogreška, preciznost, funkcija gubitka, izvješće o klasifikaciji i matrica zabune nad skupom za učenje i validaciju. Na slici 4 prikazana je vrijednost IoU kroz epohe algoritma na skupu za treniranje i testiranje.



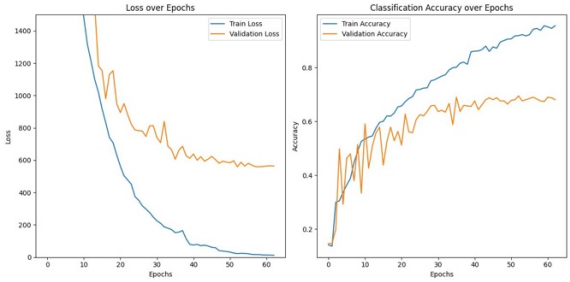
Slika 4

Iz grafa je vidljivo da IoU kroz epohe kontinuirano raste na skupu za treniranje dostižući vrijednost približno 0.8. No, na skupu za validaciju IoU stagnira na vrijednosti oko 0.44. Ta značajna razlika između skupa za učenje i skupa za validaciju ukazuje na prenaučenost algoritma koja je posljedica relativno male baze podataka. Osim metrike IoU, na slici 5 prikazan je graf vrijednosti srednje kvadratne pogreške (MSE) graničnih okvira kroz epohe algoritma.



Slika 5

Na podacima za treniranje, MSE naglo i značajno opada i približava se 0, dok na podacima za validaciju ostaje značajno viši, uz izražene oscilacije kroz epohe. To može biti posljedica prekomjerne složenosti modela ili, u ovom slučaju vjerojatnije, nedovoljno velike baze podataka zbog čega algoritam ne generalizira dovoljno dobro. Na slici 6, prikazani su grafovi funkcije gubitka i točnosti klasifikacije na skupu za treniranje i validaciju kroz epohe.



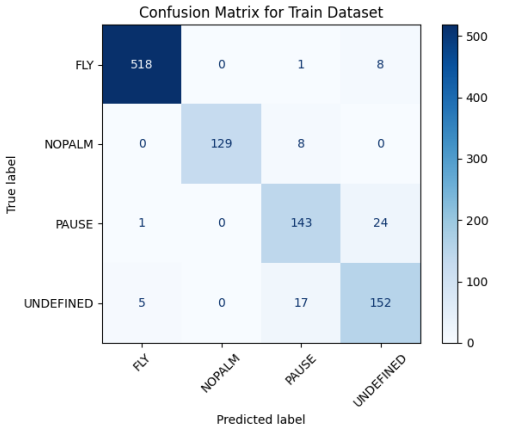
Slika 6

Na grafu funkcije gubitka vidljiv je veliki disparitet između performansi na skupu za učenje i validaciju. Dok na skupu za učenje gubitak brzo opada tijekom prvih epoha nakon čega se stabilizira na niskim vrijednostima, na validacijskom skupu on značajno oscilira i stagnira na značajno višoj razini. Sukladno tome algoritam na skupu za treniranje postiže točnost klasifikacije od skoro 100 %, dok na skupu za validaciju u prosjeku postiže točnost malo manju od 70 %. Na slici 7 prikazano je izvješće o klasifikaciji na skupu za treniranje i validaciju.

```
----- Metrics for Train Dataset -----
Accuracy: 0.9930

Classification Report:
              precision    recall  f1-score   support

         FLY       0.99      1.00      1.00       527
      NOPALM       1.00      1.00      1.00       137
       PAUSE       0.99      0.99      0.99       168
   UNDEFINED       0.99      0.97      0.98       174

    accuracy                           0.99      1006
   macro avg       0.99      0.99      0.99      1006
weighted avg       0.99      0.99      0.99      1006

----- Metrics for Test Dataset -----
Accuracy: 0.6944

Classification Report:
              precision    recall  f1-score   support

         FLY       0.80      0.83      0.82       215
      NOPALM       0.82      0.78      0.80        63
       PAUSE       0.56      0.62      0.59        74
   UNDEFINED       0.39      0.33      0.36        80

    accuracy                           0.69       432
   macro avg       0.64      0.64      0.64       432
weighted avg       0.69      0.69      0.69       432
```

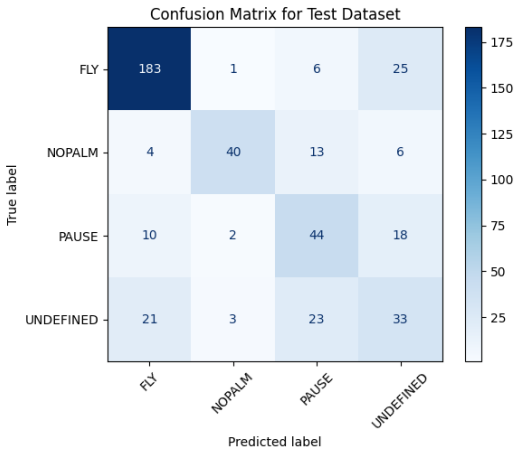
Slika 7

Vidljivo je da na skupu za treniranje algoritam postiže odlične rezultate s vrijednostima preciznosti i odaziva 0.99 za svaku klasu. S druge strane, algoritam postiže preciznost 0.69 i odaziv 0.64 na skupu za testiranje što ukazuje na probleme generalizacije modela. Vidljivo je kako algoritam postiže solidne vrijednosti preciznosti i odaziva za klasu FLY, koja ima najviše primjera u skupu. Model također ima dobre performanse za klasu NOPALM tj. ,vrlo dobro prepoznaje da na slici nije prisutan dlan osobe. Značajno lošije rezultate postiže kod klasa PAUSE i UNDEFINED koje nije tako lako prepoznati kao NOPALM, a nemaju veliki udio primjera kao što ima klasa FLY. Na slici 8 prikazana je matrica zabune skupa za učenje koja ima pretežito dijagonalan oblik što označava dobre performanse algoritma. Model jako dobro razlikuje klase, osim manjih poteškoća u razlikovanju klase PAUSE i UNDEFINED



Slika 8

S druge strane, matrica zabune skupa za validaciju, prikazana na slici 9, ima značajno manje dijagonalan oblik što ukazuje na probleme u razlikovanju klasa. Osim toga što značajno slabije razlikuje klase UNDEFINED i PAUSE, model ima problema u razlikovanju između klase FLY i UNDEFINED.



Slika 9

V. Diskusija i usporedba rezultata

Model pokazuje solidne performanse na skupu za treniranje, što je vidljivo iz jako dobrih IoU metrika, visoke preciznosti, odaziva i F1-mjere algoritma za sve klase, te vrlo niske vrijednosti funkcije gubitka. Glavni problem implementiranog modela je prenaučenost prouzročena neadekvatnom veličinom skupa podataka za treniranje. Algoritam je temeljen na YOLO v1 algoritmu koji koristi napredne tehnike dubokog učenja, što omogućuje veliku fleksibilnost i prilagodljivost algoritma. No, složeni modeli uvelike ovise o kvaliteti i veličini skupa za treniranje. Neadekvatna veličina baze podataka uzrokuje prenaučenost algoritma kao što je vidljivo iz danih rezultata. Drugi modeli, poput Google-ovog Hand Landmarker, osim što imaju složene arhitekture, trenirani su na puno većoj bazi podataka i dobro generaliziraju.

VI. Zaključak

U ovom radu razvijen je sustav za prepoznavanje gesti dlana koji omogućuje upravljanje simuliranim zrakoplovom. Dan je pregled i opis korištenih metoda za prepoznavanje objekata sa slike i metrika za evaluaciju tih metoda. Istaknute su prednosti i nedostaci konvolucijskih neuronskih mreža i YOLO algoritma koji je izabran zbog ravnoteže između preciznosti i brzine koja omogućuje analizu slika u stvarnom vremenu. Implementacija programskog rješenja uključivala je stvaranje vlastite baze podataka i korištenje raznih tehnika augmentacije kako bi se proširila baza i time poboljšala učinkovitost modela. Rezultati su pokazali jako dobre performanse na skupu za treniranje, ali značajno lošije performanse na skupu za validaciju što ukazuje na prenaučenost modela zbog relativno male baze podataka.

VII. Literatura

[1] N. Buhl, „YOLO Object Detection Explained: Evolution, Algorithm, and Applications, “ Encord, April 2024 [Online]. Dostupno: https://encord.com/blog/yolo-object-detection-guide/

[2] Sachinsoni, „Concept of YOLOv1:The Evolution of Real-Time Object Detection, “ Medium, Oct. 2023 [Online]. Dostupno: https://medium.com/@sachinsoni600517/concept-of-yolov1-the-evolution-of-real-time-object-detection-d773770ef773

[3] R. Kundu, „YOLO: Algorithm for Object Detection Explained [+Examples], “ V7labs, Jan. 2023 [Online]. Dostupno: https://www.v7labs.com/blog/yolo-object-detection

[4] G. Boesch, „What is Intersection over Union (IoU)?, “ Viso.ai, Jan. 2024 [Online]. Dostupno: https://viso.ai/computer-vision/intersection-over-union-iou/

[5] V. S. Subramanyam „IOU (Intersection over Union), “ Medium, Jan. 2021 [Online]. Dostupno: https://medium.com/analytics-vidhya/iou-intersection-over-union-705a39e7acef

[6] „Introduction to object detection with deep learning, “ Superannotate, Sept. 2023 [Online]. Dostupno: https://www.superannotate.com/blog/object-detection-with-deep-learning