

**Tedensko poročilo**

pri predmetu Umetna intelegenca

**Člani skupine:**

Nik Terglav

Luka Lamprečnik

Simona Zhirova

Marija Jovanova

Teden 1 -

1. Tabela zadolžitev

| Član | Zadolžitve |
| --- | --- |
| Simona Zhirova | Model 1  -Raziskava RNN (LSTM) arhitekture  - Priprava in organizacija podatkov (koordinate gest iz Mediapipe)  - Razdelitev v učne/testne/validacijske množice |
| Marija Jovanova | Model 1  - Implementacija osnovne strukture LSTM modela (TensorFlow/Keras)  - Dokumentacija, zakaj je LSTM ustrezen za ta tip podatkov |
| Luka Lamprečnik | Model 2  - Razistakva najbolj primerne arhitekture nevronske mreže za moj primer  - |
| Nik Terglav | Model 3  -Raziskava arhitektur nevronskih mrež in odločitev, katera je primerna za moj model  -Ustvariti učne, testne in validacijske množice  - Seznaniti se z ogrodjem/knjižnico, ki bom uporabil za moj model  - Osnovna implementacija |

1. Opis dela

V prvem sklopu smo se člani skupine seznanili z osnovnimi arhitekturami nevronskih mrež in metodami strojnega učenja, ki jih bomo uporabili pri našem projektu. Preučili smo delovanje **konvolucijskih (CNN)**, **rekurentnih (LSTM)** in **gostih (Dense)** nevronskih mrež, saj vsak od treh načrtovanih modelov zahteva nekoliko drugačen pristop. Modele smo porazdelili na naslednji način:

#### **Model 1 – Gestna kontrola (Marija Jovanova in Simona Zhirova)**

Za prepoznavo gest rok bomo uporabili **rekurentno nevronsko mrežo LSTM**, saj želimo obdelovati zaporedje gibanja (časovno odvisne podatke). LSTM je posebej primeren za sledenje zaporedju koordinat, ki jih pridobimo s pomočjo Mediapipe knjižnice. Geste bodo razvrščene najprej po ciljni napravi (npr. radio, klima), nato pa po konkretni akciji (npr. vklop, izklop).

#### **Model 2 – Detekcija drže voznika**

Za zaznavo telesne drže bomo uporabili kombinacijo **konvolucijske (CNN)** in **goste mreže (Dense)**. CNN bo iz slike zaznal ključne točke telesa, nato pa bomo s pomočjo Dense plasti razvrstili držo v različne kategorije (npr. pogled naprej, pogled v stran, sključena drža itd.).

#### **Model 3 – Razvoj modela za analizo okolje-specifičnih podatkov**

Za zaznavo okolja bomo uporabili **preprosto nevronsko mrežo z CNN arhitekturo**, ki bo analizirala povzetke iz vhodnih podatkov, kot so osvetlitev, hrup v okolici ali prisotnost dodatnih oseb.

* 1. Simona Zhirova

Podatke sem organizirala v mape glede na napravo (npr. radio, ogledala, okna) in vrsto geste (npr. vklop, izklop), kar je omogočilo boljšo preglednost in enostavnejše razdeljevanje v učne, testne in validacijske množice. Pri razdeljevanju sem pazila na enakomerno zastopanost razredov in preprečila podvajanje vzorcev v različnih množicah.

Za potrebe prvega modela, namenjenega prepoznavanju gest rok, sem raziskala rekurentne nevronske mreže, s poudarkom na arhitekturi LSTM (Long Short-Term Memory). Ta vrsta mreže je posebej primerna za obdelavo zaporednih podatkov, saj omogoča učenje dolgoročnih odvisnosti v časovni vrsti. Pri preučevanju sem se osredotočila na razumevanje, kako LSTM celice shranjujejo in posodabljajo informacije prek vhodnih, izhodnih in pozabnih vrat, ter kako to vpliva na prepoznavo gibanja rok skozi čas.

Za porazdelitev podatke v mape za učne, testne in validacijske množice najprej SEM naložila podatke iz različnih map, ki vsebujejo .npy datoteke, predstavljajoče različne geste in ukaze. Za vsak niz podatkov sem izvedla potrebno obdelavo, vključno z usklajevanjem dimenzij (npr. z zapolnjevanjem, da so vsi podatki imeli enako obliko). Nato sem iz podatkov izluščila vhodne podatke in pripadajoče oznake (X in y). Vhodni podatki (X) so predstavljali koordinate, pridobljene iz Mediapipe, medtem ko so oznake (y) vsebovale razrede, povezane z vsakim vzorcem (npr. različne geste ali ukazi). Te podatke sem shranila v datoteke X.npy in y.npy, ki sem jih nato uporabila za nadaljnje korake. Kasneje sem podatke razdelila na učne, testne in validacijske množice z uporabo funkcionalnosti train\_test\_split iz knjižnice scikit-learn, kar je omogočilo uravnoteženo zastopanost različnih razredov v vseh množicah. Na koncu sem podatke organizirala v tri mape (train, val, test), kjer sem jih pripravila za nadaljnje usposabljanje modela in oceno njegovega delovanja.

* 1. Marija Jovanova

Pri zasnovi modela za prepoznavo gest rok smo se odločili za uporabo rekurentne nevronske mreže, natančneje arhitekture LSTM (Long Short-Term Memory). Geste rok so zaporedni pojav – sestavljene so iz niza premikov, ki se dogajajo skozi čas. Vsaka gesta predstavlja zaporedje koordinat, ki jih pridobimo s pomočjo knjižnice Mediapipe, in te koordinate moramo obdelati tako, da ohranim njihovo časovno povezanost. Izbrali smo LSTM, saj omogoča modelu, da "si zapomni" pomembne informacije iz prejšnjih časovnih korakov in jih uporabi pri interpretaciji trenutnega stanja. Za razliko od navadnih nevronskih mrež, ki vsako sliko ali podatkovno točko obravnavajo ločeno, LSTM upošteva zaporedje in tako bolje razume kontekst gesta. Ta sposobnost zaznavanja dolgoročnih odvisnosti je ključna za prepoznavo zapletenejših gest, kjer pomen izhaja iz celotnega gibanja, ne le posamezne pozicije. S to arhitekturo tako dosegamo boljše razumevanje dinamičnih vzorcev v podatkih in s tem tudi natančnejšo klasifikacijo uporabniških gest.

* 1. Nik Terglav

Za moj model prepoznave okolja sem se odločil za uporabo arhitekture konvolucijske nevronske mreže CNN za prepoznavanje okoljskih podatkov. Ker so posnetki zaporedje slik, mi omogoča CNN ekstrakcijo prostorskih značilnosti iz posameznih okvirjev, to mi omogoča prepoznavo določenih vzorcev. Začetni model bo zgrajen na tej arhitekturi, namreč želim prepoznati ali je na posnetku dan ali noč, CNN mi omogoča obravnavo svetlobnih pogojev, znaćnilnosti slike, kontrast… Ker želim omogočiti mojemu modelu možnost nadaljnjega razvoja sem prav tako izbral arhitekturo, ki jo je možno nadgraditi v hibridni model, npr. Dodatek LSTM ali GRU, kar omogoča obdelavo časovne dimenzije in povezanosti med okvirji posnetkov. Tako bom lahko implementiral tudi kompleksnejše funkcionalnosti. Torej trenutno sem za model zasnoval CNN arhitekturo s pomočjo katere bom rešil osnovne funkcionalnosti s sprotnim učenjem pa bom model tudi nadgradil do končnih zahtev. Model bom učil na podlagi podatkov, ki sem jih razdelil s pomočjo skripte glede na delitev: trening(70% posnetkov), validacija(15%), testiranje(15%).

* 1. Luka Lamprečnik

Cilj tega modela (detekcija drže voznika) je razvrstiti telesno držo voznika med vožnjo in tako zagotoviti, da sistem za upravljanje gest deluje le, kadar je voznik v ustreznem položaju za varno uporabo. Z modelom želimo zaznati, ali ima voznik roke na volanu, ali gleda naprej ali vstran, in ali je njegova drža vzravnana ali sključena.

Podatke za učenje modela smo zajeli z lastnimi video posnetki, kjer vsak član ekipe izvaja različne geste pred kamero. Ti posnetki ne prikazujejo resničnega okolja v avtomobilu, temveč simulirano okolje doma. S pomočjo knjižnice MediaPipe smo iz videoposnetkov pridobili koordinate ključnih točk telesa (posebno rok, ramen in glave). Te točke smo shranili v .npy datoteke in jih uporabili kot vhodne podatke za učenje.

Za razpoznavo drže uporabljamo kombinacijo konvolucijske nevronske mreže (CNN), ki prepozna vzorce v položaju telesa, in goste plasti (Dense), ki iz teh značilnic izpelje končno klasifikacijo. Model predvidi, v kateri kategoriji se nahaja drža: npr. “roki na volanu in pogled naprej” ali “eno roko spodaj in pogled v stran”. Ti razredi služijo kot dodatni filter za sprejemanje ali zavrnitev geste.

Model bo treniran s pomočjo knjižnic TensorFlow/Keras, podatki pa bodo razdeljeni na učne, validacijske in testne množice, da preverimo njegovo zanesljivost.

1. Težave, rešitve in ideje za nadaljne delo

**Model 1** - Ena izmed glavnih težav je bila, da so nekatere kategorije v podatkovnem naboru vsebovale premalo primerov (manj kot dva). Zaradi tega je prišlo do napake pri razdeljevanju podatkov na učni, validacijski in testni del z uporabo metode train\_test\_split in argumenta stratify. Funkcija zahteva, da ima vsaka razred najmanj dva primera, kar pri redkejših razredih ni bilo zagotovljeno. **Rešitev** je bila odstranitev vseh primerov, ki so pripadali razredom s premalo podatki, preden sva izvedla delitev. Tako sva zagotovila, da ima vsaka kategorija dovolj primerov za pravilno deljenje in treniranje modela.

**Ideje za nadaljnje delo:** Razširitev podatkovnega nabora, da bi vsi razredi imeli uravnoteženo število primerov, lahko uporabljamo naprednejših arhitektur, kot so **Bidirectional LSTM** ali **CNN-LSTM** za boljše razpoznavanje, dodajanje augmentacije podatkov (npr. hrup, razteg) za izboljšanje generalizacije modela in na koncu izdelava uporabniškega vmesnika za prepoznavanje gest v realnem času.

**Model 2** - Največja težava je, da posnetki niso narejeni v realnem avtomobilu, temveč doma, pogosto brez celotnega telesa v kadru. Zaradi tega bo model težko generaliziral na prave razmere. Rešitev je zajem novih posnetkov, kjer bo telo voznika bolje vidno in bo kamera postavljena podobno kot v avtomobilu. V prihodnje razmišljamo tudi o razširitvi modela z LSTM ali 3D CNN arhitekturo, ki bi zaznala spremembe drže skozi čas, ne le v eni sličici.