



Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko

Končno poročilo za projekt “Upravljanje z gestami”

pri predmetu Signale in slike

Simona Zhirova

1. Uvod

V okviru projekta pri predmetu Signali in slike sem se posvetila razvoju sistema za upravljanje avtomobilskega radia z gestami. Moj namen je bil zasnovati rešitev, ki bi voznikom omogočila varno in intuitivno upravljanje radijskih funkcij, kot so preklapljanje radijskih postaj, vklop ali izklop radia ter prilagajanje glasnosti, brez odvracanja pozornosti od vožnje. Sistem temelji na razpoznavanju gest, zajetih s spletno kamero, kjer sem uporabila knjižnico MediaPipe za zaznavanje položajev točk na roki, in nevronske mreže za klasifikacijo gest.

Cilj moji del projekta je bil razviti robusten model, ki prepozna šest specifičnih gest: `next_station` (preklop na naslednjo postajo), `previous_station` (preklop na prejšnjo postajo), `turn_off_radio` (izklop radia), `turn_on_radio` (vklop radia), `volume_down` (zmanjšanje glasnosti) in `volume_up` (povečanje glasnosti).

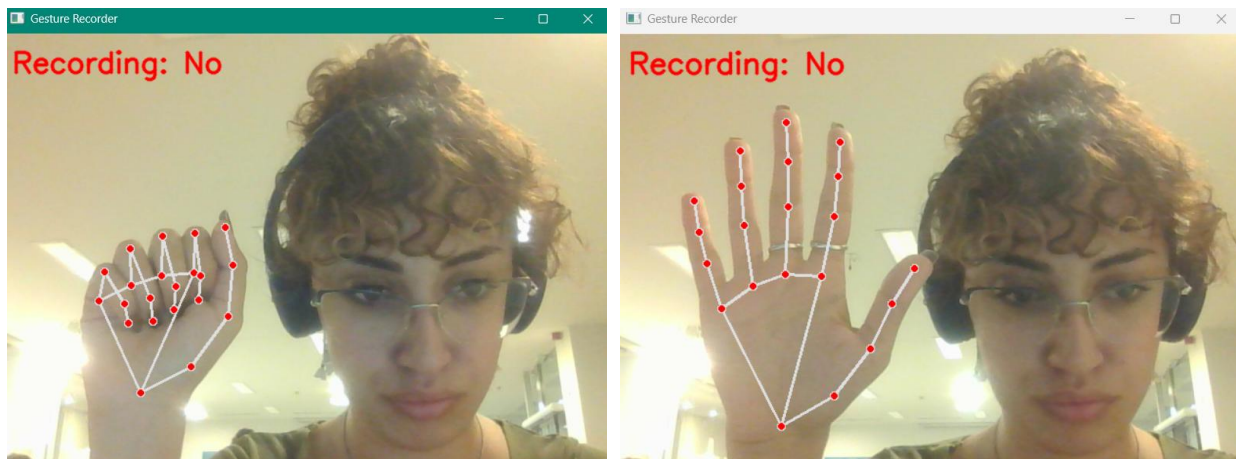
2.1. Zajem podatkov

Za zajemanje gest sem uporabila **spletno kamero** in knjižnico **MediaPipe Hands**, ki omogoča zaznavo **21 točk na roki** (skelet roke).

Postopek:

- Zagnala sem Python skripto, ki omogoča ročno upravljanje snemanja (s tipko r).
- Ko sem pritisnila r, se je začelo snemanje koordinat točk na roki.
- Ko sem ponovno pritisnila r, se je snemanje ustavilo.
- Po končanem snemanju sem vnesla oznako za posneto gesto.

Vsak zajem je bil shranjen v **.npy datoteko**, ki vsebuje zaporedje koordinat (x, y, z) za vsako točko na roki v vsakem posnetem okvirju.



2.2. Označevanje podatkov

Za svojo nalogo sem zajela skupaj 180 signalov iz 6 različnih gest. Vse gestikulacije, ki sem jih snemala, so vključevale geste za upravljanje z radio. Vsaka gesta je bila zajeta s pomočjo Web kamere in knjižnice MediaPipe, ki omogoča natančno sledenje in beleženje gibov rok skozi več zaporednih posnetkov (okvirjev). Za vsako gesto sem zajela določeno število primerov, pri čemer sem poskrbela, da bi vsak primer vseboval najmanj 30 zaporednih okvirjev, ki predstavljajo gibanje roke med izvedbo geste.

Za vsako izmed teh gest sem ustvarila ustrezne .npy datoteke, ki vsebujejo shranjena zaporedja koordinat točk na roki. Na ta način imam za vsako gesto zbrane po 3 primerov s pravilno označenimi koordinatami, ki bodo uporabljeni pri kasnejšem modeliranju.

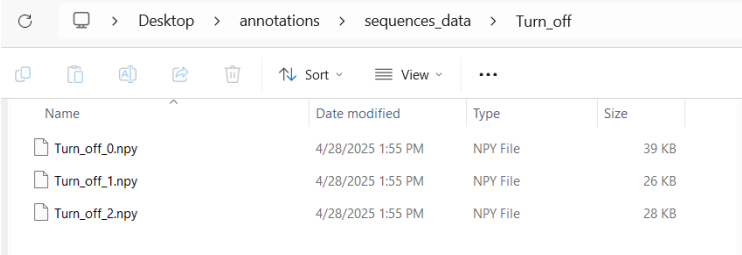
(30 primerov * 6 gestov = 180 primerov)

```
[INFO] Started recording...
[INFO] Recording stopped. Please enter label:
>> Volume_down
[INFO] Saved sequence: sequences_data\Volume_down\Volume_down_0.npy
[INFO] Started recording...
[INFO] Recording stopped. Please enter label:
>> Volume_down
[INFO] Saved sequence: sequences_data\Volume_down\Volume_down_1.npy
```

Po vsakem snemanju sem ročno vnesla oznako gibe (npr. turn_radio_off).

Struktura mape z oznakami:

```
sequences_data/
├── turn_off/
│   ├── turn_off_0.npy
│   ├── turn_off_1.npy
│   └── ...
├── volume_up/
│   ├── volume_up_0.npy
│   └── ...
└── ...
```

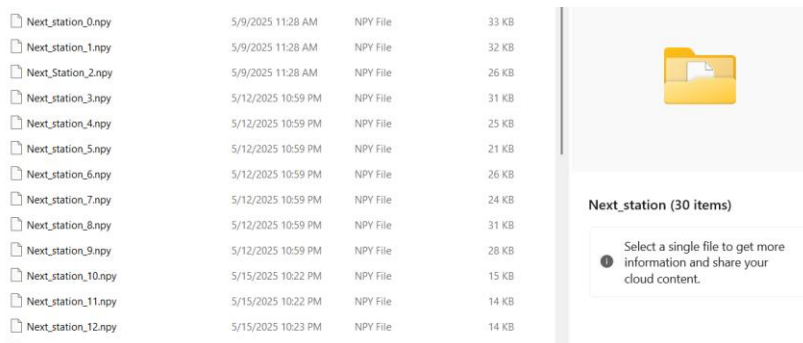
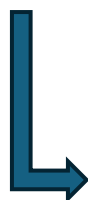
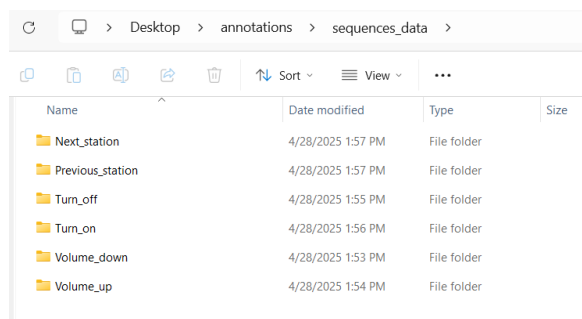


Name	Date modified	Type	Size
Turn_off_0.npy	4/28/2025 1:55 PM	NPY File	39 KB
Turn_off_1.npy	4/28/2025 1:55 PM	NPY File	26 KB
Turn_off_2.npy	4/28/2025 1:55 PM	NPY File	28 KB

Vsaka gesta ima svojo podmapo.

Imena datotek so v formatu: oznaka_številka.npy.

Tako so signali pravilno organizirani za nadaljnje učenje modela.



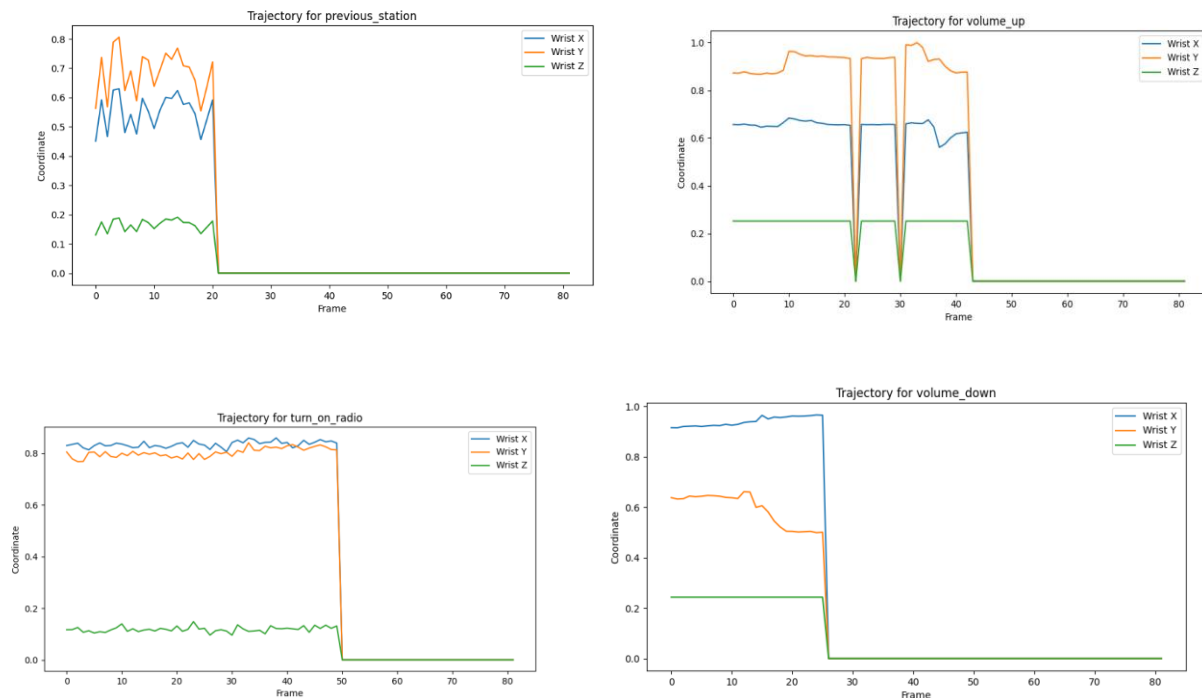
Za obdelavo zajetih podatkov sem uporabila Python skripto `data_radio.py`, ki je posnetke razporedila v datoteki `X_radio.npy` (značilke: 80 sličic z 63 koordinatami 21 točk na roki) in `y_radio.npy` (oznake gest), kar je zasedlo približno 80 MB prostora na disku.

2.3. Augmentacija podatkov

Vsak vzorec, zajet v obliki 80 sličic z 63 značilkami (3D koordinate 21 točk na roki), sem najprej normalizirala na interval $[0, 1]$, nato pa naključno uporabila do tri od naslednjih augmentacij z verjetnostjo 0.5: dodajanje Gaussovega šuma (nivo 0.01) za simulacijo tresenja rok, skaliranje amplitude (razpon 0.8–1.2) za variacije v velikosti giba, premik sličic (do 5 sličic) za različne začetke gest, zameglitev gibanja z drsečim povprečjem (jedro velikosti 3), naključno izpuščanje 5 % sličic za simulacijo manjkajočih podatkov in rotacija točk okoli zapestja (do $\pm 10^\circ$). Po vsaki augmentaciji sem podatke ponovno normalizirala, da sem ohranila konsistentnost. Augmentirani podatki so bili shranjeni kot `demo_X_augmented.npy` in `demo_y_augmented.npy`, kar je povečalo raznolikost učne množice in izboljšalo prilagodljivost modela na realne pogoje.

2.4. Statistika

V okviru projekta sem izvedla tri snemalne seje, v katerih sem zajela skupno 180 primerkov podatkov, enakomerno razporejenih po 30 primerkov za vsako od šestih radijskih gest (next_station, previous_station, turn_off_radio, turn_on_radio, volume_down, volume_up). Vsak primer je vseboval 80 sličic z 63 značilkami (3D koordinate 21 točk na roki), shranjenih v formatu NumPy (X_radio.npy, y_radio.npy), kar je zasedlo približno 80 MB prostora na disku. Snemanje in obdelava sta zahtevala več časa, saj sem nekatere posnetke ponovno zajela zaradi neustrezne kakovosti, kot so slaba osvetlitev ali netočni gibi rok, skupno približno 8 ur za snemanje in 4 ure za anotacijo ter preverjanje podatkov. Za učno množico sem uporabila 80 % podatkov, kar pomeni približno 144 primerkov, preostalih 36 primerkov pa sem rezervirala za testiranje. Po augmentaciji, ki je vključevala tehnike, kot so dodajanje šuma in rotacija točk, ocenjujem, da je učna množica narasla na približno 400–500 primerkov, kar je povečalo robustnost nevronske mreže.



3. Zaključek

Moj del projekta za brezkontaktno upravljanje avtomobilskega radia sem uspešno izvedla, saj sem zajela, označila in augmentirala podatke za šest gest (`next_station`, `previous_station`, `turn_off_radio`, `turn_on_radio`, `volume_down`, `volume_up`), usposobila nevronske mreže ter razvila grafični uporabniški vmesnik (GUI) za testiranje v simuliranem avtomobilskem okolju. Zadovoljna sem s kakovostjo zajetih podatkov, ki so raznoliki in vključujejo različne svetlobne pogoje ter variacije gest, ter z augmentacijo (dodajanje šuma, skaliranje amplitude, rotacija točk), ki je izboljšala robustnost modela, prav tako pa je GUI omogočil intuitivno testiranje. Kljub temu sem opazila, da sta gesti `next_station` in `previous_station` včasih napačno prepoznani, verjetno zaradi podobnosti gibov ali prekomernega izpuščanja sličic. V prihodnje bi zmanjšala izpuščanje sličic, zajela dodatne posnetke za ločitev gest in izboljšala postavitev kamere. Ideje za izboljšave vključujejo realnočasovno prepoznavanje z drsečim oknom, razširitev na druge funkcije (npr. navigacija) in testiranje v pravem vozilu. Skupno sem zadovoljna z izkušnjami na področju računalniškega vida in razvoja vmesnikov ter z potencialom sistema za nadaljnji razvoj.