

### **3. Výstup Maturitní práce**

---

## **Zařízení na snímání gest**

---

*autor:*

Štěpán Bílek

*konzultant:*

Jaroslav Kořínek

Školní rok 2025/2026

# 1 Úvod

Cílem 3. výstupu obecně bylo dodělat náležitosti HW, rozšiřovat ovládací program, vytvořit soubory tréninkových dat, implementovat model AI. Hardware a tréninková data měl na starosti spolupracovník Lukáš Karásek, mým hlavním úkolem byla implementace neuronové sítě. Kvůli okolnostem jsem se ale musel zprvu soustředit na rozšiřování ovládacího programu a na neuronovou síť až později, aby úkoly byly zvládnutelné do daného terminu.

## 2 Ovládací program a uživatelské rozhraní

Původně jsem o ovládacím programu(dále OP) přemýšlel pouze jako o ovladači jakékoli jiné periferie např. myši, tedy jako o procesu o kterém uživatel neví a běží na pozadí. Při otázce sběru dat a trénování neuronové sítě mě napadly dva odlišné architektonické způsoby, jak se k OP stavět. Prvním byl zachovat jednoduchost OP a dva výše zmíněné úkoly udělat pomocí vedlejších scriptů/nástrojů. Druhým bylo spojit všechny potřeby dohromady a udělat jeden komplexní celek, který to zvládne. Rozhodl jsem se pro přístup druhý.

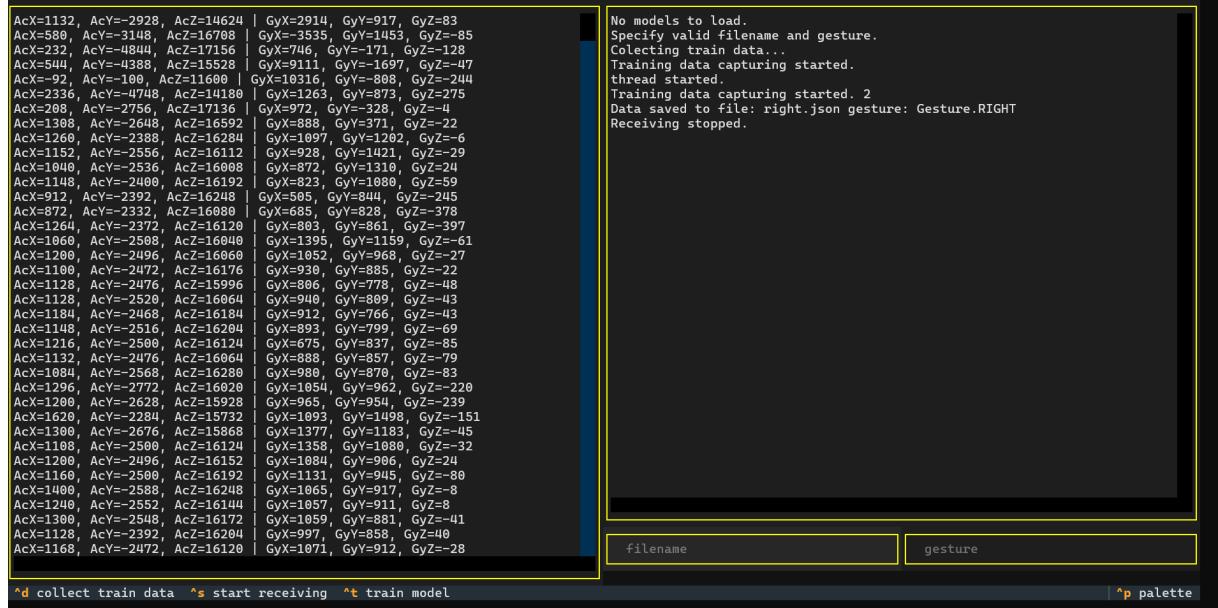
Jelikož už OP není standarním ovladačem a zastává několik úkolů (ovladač, sběrač trénovacích dat, manažer modelů AI) a bude vidět, potřebuje rozumné uživatelské rozhraní. Uživatelské rozhraní bude pro jednoduchost ponecháno v terminálu, avšak musí přehleně vypisovat veškeré důležité záznamy a přívětivě registrovat vstupy uživatele.

Před začátem práce na výstupu 3 uměl ovládací program přijímat a zpracovávat data ze senzorického zařízení a nepraktickým způsobem sbírat a ukládat data pro vytrénování neuronové sítě. Program nebyl interaktivní - vše se muselo měnit v kódu - a jeho výstup byl poněkud nepřehledný. Pro efektivní práci jsem se rozhodl používat na TUI knihovnu textualize [1], která po vizuální stránce maximalizuje potenciál terminálu.

Okno terminálu jsem rozdělil na 2 velká a 2 malá pole. Levé pole slouží k zobrazování živých dat ze senzorického zařízení, pravé hlavní pole slouží k zobrazování informačních hlášek. Dvě malá pole slouží k zadání jména souboru a aktuálně nahrávaného gesta při sběru trénovacích dat. Kvůli tomu, jak knihovna funguje jsem musel kód poněkud upravit, aby architektonicky dával smysl a vše běželo plynule. Zároveň byly části programu, jež to vyžadovaly např. příjem dat při normálním provozu, upraveny aby běžely na vlastních vedlejších vláknech. Díky tomu program dokáže zároveň přijímat data, zobrazovat je a vyhodnocovat.

Aktuálně program po spuštění umí zachytávat trénovavací data, vytrénovat neuronovou síť a začít normální provoz. Mezi těmito funkcemi si díky UI uživatel interaktivně volí sám. Výjimky jsou ošetřeny a popřípadě zobrazeny, ovládací program tedy nemůže znenadání skončit. Ačkoli program jěště není za běžného provozu plně otestován, byla

snaha, aby působil robustně.



Obrázek 1: uživatelské rozhraní

### 3 Implementace neuronové sítě

Neuronová síť byla realizována dle návrhu z mého předchozího výstupu pomocí knihovny tensorflow [2]. Pro trénování jsou brána dříve zachycená serializovaná data. Jak jsem uváděl v předchozím výstupu, výsledkem neuronové sítě je pravděpodobnost s jakou dokázala určit správnou třídu;gesto. To známená, že přijme-li na vstupu data se senzoru o pohybu doleva, výstupem bude např. 0.91, 0.09, pravděpodobnost gesta doleva následovná pravděpodobností gesta doprava.

Při trénování zná síť vstupní data i očekávaný výsledek. Vstupní data musí být nejdříve normalizována do intervalu  $< -1; 1 >$  pro případ, že by např. jedna z hodnot byla v tisících a druhá v desítkách, tehdy by větší čísla neuronovou síť ovlivňovala více. O výstup v pravděpodobnostních třídách se stará funkce softmax, která při tréniku vyžaduje tzv. one-hot encoding. To pro naše dvě gesta vypadá následovně:

$$doleva = [1; 0] \quad (1)$$

$$doprava = [0; 1] \quad (2)$$

Výsledkem trénování je model, který se dá uložit jako soubor a ovládací program je schopen ho načíst.

## 4 Závěr

Model se zdánlivě povedlo vytrénovat úspěšně, avšak je potřeba ho otestovat, což bude z části náplní 4. výstupu maturitní práce. Jak je psáno v zadání a analýze maturitní práce, účelem je efektní přepínání snímků prezentace pomocí gest prezentujícího. Zmíněno ale nebylo, že se spolupracovníkem LK hodláme na projektu pokračovat mimoškolně a maturitní práce má sloužit jako ověření konceptu a prototyp. Zařízení má být v budoucnu schopné rozpoznat složitá gesta jako např. kruh. Právě pro to je neuronová síť i ovládací program takto komplexní, z důvodu rozšířitelnosti. Díky této zatím nevyužité síle a složitosti model při tréninku dosáhl přesnosti 100%, což by se mylně mohlo jevit jako chyba.

**Veškeré soubory k projektu jsou veřejně k dispozici na platformě github [3].**

## Odkazy

- [1] URL: <https://textual.textualize.io/>.
- [2] URL: <https://www.tensorflow.org/>.
- [3] URL: <https://github.com/C4LL1ST0/MAP-Zarizeni-pro-snimani-gest>.

## Obsah

1 Úvod	II
2 Ovládací program a uživatelské rozhraní	II
3 Implementace neuronové sítě	III
4 Závěr	IV