MATURITNÍ PRÁCE

Aplikace neuronových sítí (AI)

André Schuhmacher

|  |  |
| --- | --- |
| Vedoucí práce: | Tomáš Havrda |
| Studijní program: | IT Technik pro výrobní systémy |
| Studijní obor: | 26-41-M/01 |

2024

Odevzdáním této maturitní práce na téma Aplikace neuronových sítí (AI) potvrzuji, že jsem ji vypracoval pod vedením vedoucího práce samostatně za použití v práci uvedených pramenů a literatury. Dále potvrzuji, že tato práce nebyla využita k získání jiného nebo stejného titulu.

Místo a datum odevzdání práce

Volitelné poděkování vedoucímu práce, případně dalším osobám, které chcete jmenovat.

**ABSTRAKT**Abstrakt v rozsahu 200 slov je stručný výtah práce. Slouží především jako pomoc čtenáři rychle se zorientovat v dané práci, obsahuje stručnou charakteristiku práce, její cíle a použité metody. Abstrakt obsahuje také shrnutí výsledků práce.

**KLÍČOVÁ SLOVA**Neuronová síť, GUI, automatizace, šipky

Obsah

[Úvod 5](#_Toc161052206)

[2 Sběr dat a trénink neuronové sítě 6](#_Toc161052207)

[2.1 Snímání dopadové plochy 6](#_Toc161052208)

[2.2 Anotace a příprava tréninkových dat 7](#_Toc161052209)

[2.3 Naučení neuronové sítě 8](#_Toc161052210)

# 

# Úvod

Tato maturitní práce zkoumá možnosti využití neuronových sítí (AI) pro zefektivnění a zjednodušení procesu počítání bodů ve hře šipky. Cílem práce je navrhnout a implementovat systém s umělou inteligencí, který by dokázal automaticky sledovat skóre a usnadnit tak hru pro všechny účastníky.

V šipkách se snaží jeden nebo více hráčů dosáhnout nulového skóre odečítáním bodů od 501. Toho dosahují házením šipek na terč, rozdělený do segmentů s různou hodnotou. Hra končí v okamžiku, kdy jeden z hráčů dosáhne nulového skóre prostřednictvím hození šipky do vnějšího segmentu s dvojitou hodnotou. Například, pokud zbývá 20 bodů, hráč musí dosáhnout dvojité 10 pro ukončení hry.

Vyhodnocování bodů v této maturitní práci se realizuje prostřednictvím kamery, která je zaměřena na terč. Tato kamera je připojena k počítači Intel NUC, na kterým běží software, který aktivně sleduje a zaznamenává dopad šipek. Tento proces probíhá pomocí sledování změn na povrchu terče a monitorování zvukových úrovní. Jakmile šipka dopadne na terč, začne systém provádět vyhodnocování a přenáší získané hodnoty do databáze. Tyto údaje jsou následně získávány webovým serverem, který je zodpovědný za zobrazení dat na uživatelském rozhraní.

# Sběr dat a trénink neuronové sítě

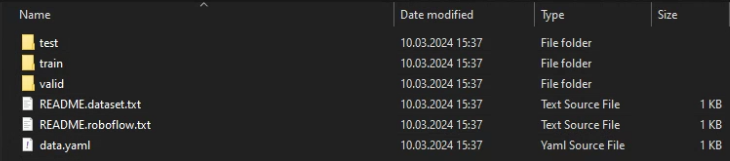
## Snímání dopadové plochy

Prvním krokem je důkladně zvážit možné metody detekce šipky, která dopadla na kamerovou plochu. Kontrola, zda se obraz mění každých x snímků, a ověření, zda nový objekt na novém snímku je šipka (pomocí neuronové sítě), je až moc náročná práce. Proto jsem zvolil systém využívající knihovnu PortAudio k zachycení zvukových úrovní z mikrofonu kamery. Pokud dojde k náhlému nárůstu zvukové úrovně, uloží se hodnota do proměnné (true/false). Tato zvuková data jsou kombinována s kamerou, která snímá dopadovou plochu a každých 0,3 sekundy kontroluje změnu obrazu. V případě pozitivní změny je pomocí knihovny OpenCV vypočítána změna v obraze a porovnána s parametry odpovídajícími šipce, jako je třeba srovnání nových pixelů s mými specifikovanými parametry, například velikostí - pomocí obdélníku kolem největší pixelové změny.

Obr. PortAudio Callback

## Anotace a příprava tréninkových dat

Anotace představuje proces přípravy obrázků pro trénink neuronové sítě. Pro anotaci jsem se rozhodl využít platformu Roboflow, která nabízí jednoduchý přístup k této přípravě. Nahraji obrázky a provedu anotaci kde pomocí obdélníků oznařím plochu objektů, které má model detekovat, v mém případě jsou to hroty šipek. Například vytvořím 300 obrázků pro trénink, kde každý obsahuje kombinaci tří šipek vržených různým způsobem do terče. Na každém obrázku musím označit všechny hroty všech šipek. Poté mi Roboflow sestaví tréninkový balíček, který je složen z 70% obrázků pro trénink, 20% pro testování a 10% pro predikci.



Obr. Ukázka trénovacího balíčku

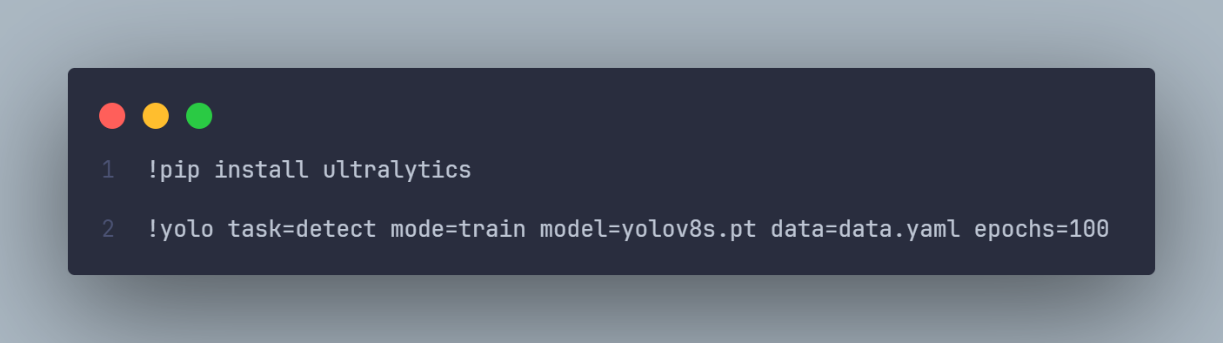


Obr. Ukázka anotace

## Naučení neuronové sítě

Učení neuronové sítě je sice relativně jednoduché, avšak náročné zejména na čas a hardware. Proto jsem objevil platformu, která mi umožňuje využít výkonný hardware, který je obvykle drahý a pro většinu lidí, včetně studentů jako jsem já, nedostupný. Tato platforma se nazývá Google Colab, je zdarma dostupná a poskytuje bezplatný přístup např. k grafické kartě NVIDIA Tesla T4 která je určená pro trénování neuronových sítí, nebo k jednotce zpracování tensorů (TPU), která může být až 2/3x rychlejší než GPU. (Trénování s CPU je v dnešní době zastaralé, jelikož je nesmírně pomalé a neefektivní.)

Jako neuronovou síť a model jsem vybral YOLOv8 díky jeho dostupnosti, jednoduchosti a obsáhlé dokumentaci.

Pro trénování neuronové sítě potřebujeme několik věcí: stažený model YoloV8, (v mém případě) knihovny CUDA NN a CUDA Toolkit, můj tréninkový balíček vytvořený v předešlé kapitole a určený počet epoch. Termín "epocha" odkazuje na jeden kompletní průchod přes tréninkový balíček. Ideálně bych chtěl trénovat přibližně 100 epoch dle potřeby. 

Obr. Příkazy na trénování modelu YoloV8