



Vyšší odborná škola
a Střední průmyslová škola elektrotechnická
Plzeň, Kotterovská 85

DLOUHODOBÁ MATURITNÍ PRÁCE S OBHAJBOU

Téma:

Zařízení pro rozpoznávání objektů

Autor práce: Jaroslav Šimek
Třída: 4.L
Vedoucí práce: Ing. Pavel Jedlička
Dne: 31.3.2025
Hodnocení:

width=!,height=!,pages=1

Anotace a poděkování

Téma mojí maturitní práce je rozpoznávání objektů. Cílem této práce je navrhnout a vytvořit zařízení, které bude schopno rozpoznávat objekty v reálném čase nebo ze záznamu kamery. Projekt je zaměřen na využití moderních metod strojového učení, konkrétně neuronových sítí, k rozpoznávání objektů. Součástí práce je také vytvoření grafického uživatelského rozhraní, které uživateli umožňuje snadné ovládání rozpoznávacího modelu, výběr vstupního zdroje a přehledné zobrazování detekovaných výsledků. Zařízení společně s GUI je navrženo s důrazem na jednoduché použití, přehlednost a praktickou využitelnost například v různých oblastech.

Prohlašuji, že jsem tuto práci vypracoval samostatně a použil informace z internetu, webů a AI, které vhodným způsobem cituji a uvádím v seznamu použitých zdrojů informací.

V Plzni dne 31. 3. 2025

Podpis:

Annotation and Acknowledgments

The topic of my graduation thesis is object recognition. The aim of this thesis is to design and create a device that will be able to recognize objects in real time or from camera footage. The project focuses on the use of modern machine learning methods, specifically neural networks, for object recognition. The work also includes the creation of a graphical user interface that allows the user to easily control the recognition model, select the input source and clearly display the detected results. The device together with the GUI is designed with an emphasis on ease of use, clarity and practical applicability for example in different domains.

I declare that I have prepared this thesis independently and used information from the Internet, websites and AI, which I quote appropriately and list in the list of sources of information used

In Pilsen on March 31, 2025

Signature:

Obsah

Úvod	7
1 Použitý hardware	8
1.1 Zařízení	10
2 Rozpoznávání objektů	11
3 Použité knihovny	12
3.1 OpenCV	12
3.2 Ultralytics	13
3.3 Tkinter	13
3.4 Knihovna TensorFlow	14
3.5 Knihovna Model Garden	14
4 Použitý model	15
4.1 Model YOLOv8	15
5 Uživatelské rozhraní	16
5.1 Návrh	16
5.2 Finální podoba GUI	17

5.3 Realizace	17
5.3.1 Funkcionalita a prvky GUI	18
5.3.2 Technická realizace	18
5.3.3 Funkční tlačítka	20
6 Testování	21
6.1 Testování zařízení	21
6.2 Testování GUI	21
6.3 Testování modelu	23
7 Výsledky testování	24
7.1 Úspěšné testování	24
7.2 Neúspěšné testování	27
8 Návod na použití	28
Závěr	30
Literatura	31
Seznam obrázků	32

Úvod

Moje maturitní práce je zaměřená na implementaci systému rozpoznávání objektů pomocí Raspberry Pi 4 a knihovny TensorFlow, přičemž se zaměříme na integraci web kamery. K implementaci jsme použili model YOLOv8, který bude integrován na Raspberry Pi 4. Důležitou součástí projektu je propojení zařízení s kamerou, ta slouží jako zdroj vstupů pro detekci objektů. Cílem maturitní práce je vytvořit funkční zařízení, které bude schopné rozpoznávat objekty, a uživatelské rozhraní, které bude přehledně poskytovat výstupy.

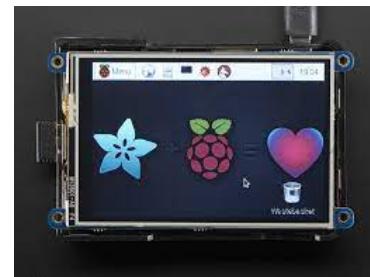
Dalším dílem práce je uživatelské rozhraní. V práci detailně popisují kompletní návrh a design GUI, důraz byl kladen především na jednoduchost a uživatelskou přívětivost. Také zde popíšu své myšlení při rozvržení celého GUI. V něm je možné spustit i zastavit detekování objektů, zvolit vložení vstupu ze zařízení(stažený obrázek, videosekvence), a možnost zobrazení výsledku přímo na obrazovce, to všechno díky funkčním tlačítkům. GUI také obsahuje možnost úpravy vloženého vstupu a ukládání výsledků detekce objektů do csv souboru.

Důležitou částí práce je také optimalizace běhu modelu, který rozpoznává objekty tak, aby byl zajištěn plynulý provoz i na výpočetně omezeném zařízení, kterým je právě námí použité Raspberry Pi 4. Systém je navržen s důrazem na jednoduché ovládání, přehlednost a praktické využití jako je reálné nasazení zařízení, sledování okolí nebo vzdělávání v oblasti strojového učení.

1 Použitý hardware



Obrázek 1.1: Raspberry Pi4, jednodeskový počítač, jsme zvolili díky jeho výkonnému hardweru a kompaktnímu provedení. Díky jeho operační paměti 8GB RAM jsme mohli provádět naročnější operace. Procesor Broadcom BCM2711 s frekvencí 1,5 GHz a čtyřmi jádry poskytuje dostatek výkonu pro různé úlohy.



Obrázek 1.2: PiTFT Plus je 3,5palcový rezistivně dotykový displej s rozlišením 480x320 pixelů, navržený speciálně pro Raspberry Pi. Pro lepší zobrazení naší aplikace by byl optimální větší displej, ale zařízení by bylo o dost nákladnější, takže jsme se rozhodli pro tuto variantu.

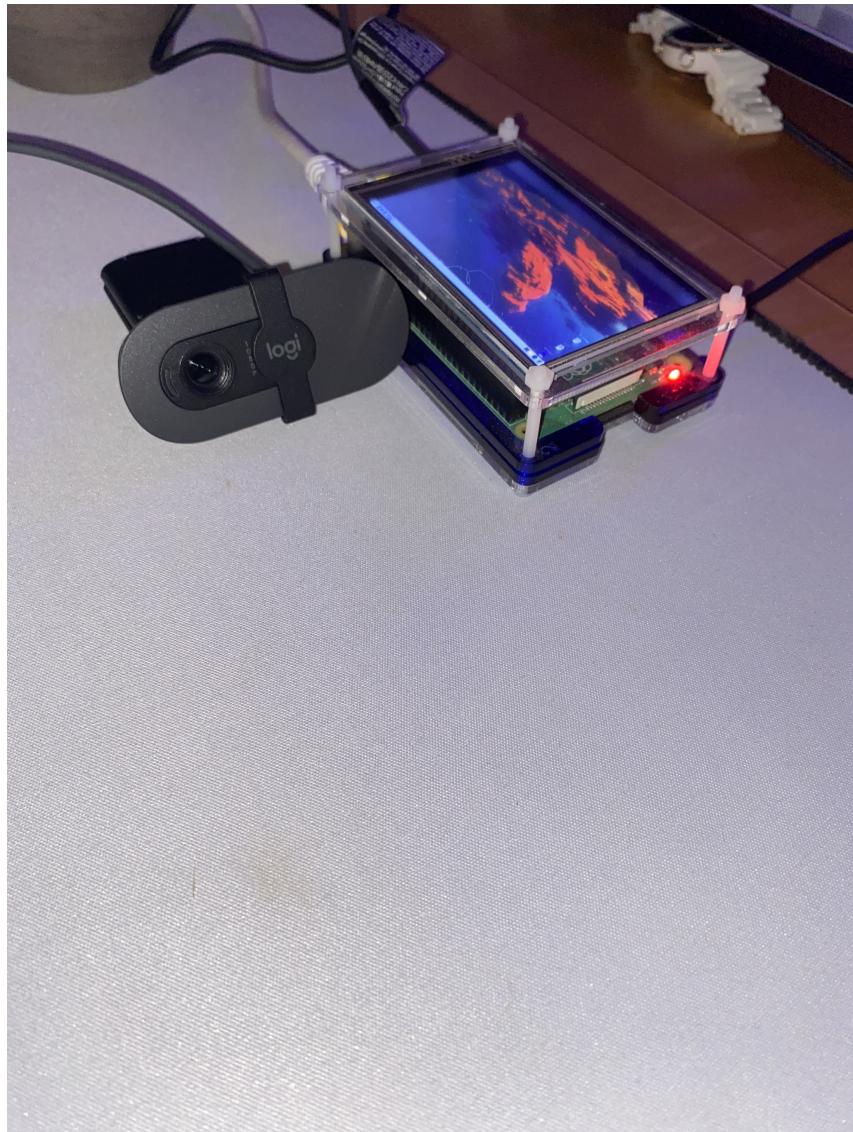


Obrázek 1.3: Kameru Logitech Brio 100 jsme zvolili pro její HD kvalitu s vysokým rozlišením 1080p. Díky této vlastnosti máme možnost pořizovat ostré snímky, které jsou následně rozpoznávány. Díky propojení s Raspberry Pi 4 pomocí USB je vhodné pro integraci s různými aplikacemi.



Obrázek 1.4: Paměťovou kartu MicSDX od Kingstona jsme zvolil, protože má dostatečný úložní prostor pro ukládání dat s kapacitou 128 GB, díky takto vysoké kapacitě můžeme dlouhodobě ukládat velké soubory a zajistit plynulý chod projektů na Raspberry Pi 4.

1.1 Zařízení



Obrázek 1.5: Na obrázku č. 5 můžete vidět finální podobu našeho zařízení pro rozpoznávání objektů. Vždy při spuštění se na ploše dotykového displeje objeví ikona, která spustí program pro rozpoznávání objektů. Výsledky detekce jsou vždy viditelné přímo na obrazovce.

2 Rozpoznávání objektů

Rozpoznávání objektů je proces, při kterém počítačový systém automaticky detekuje a identifikuje objekty v obrazech nebo videích. Nejprve jsou získána vstupní data, například z kamer, fotografií či videonahrávek. Tato data následně procházejí předzpracováním, kde dochází k úpravě kontrastu, odstranění šumu a zvýraznění klíčových prvků, aby byl výsledek analýzy co nejpřesnější.

Samotná detekce objektů probíhá pomocí algoritmů strojového učení, zejména konvolučních neuronových sítí (CNN). Tyto modely se učí rozpoznávat tvary, vzory a specifické rysy objektů na základě rozsáhlých trénovacích dat. Během učení se systém postupně zlepšuje a dosahuje stále vyšší přesnosti při klasifikaci a lokalizaci objektů.

Po úspěšném natrénování lze model aplikovat na nové obrázky či videa. V této fázi analyzuje vizuální obsah, identifikuje objekty, určuje jejich polohu a přiřazuje jim odpovídající kategorie, například osoby, vozidla nebo zvířata. Výsledky detekce lze využít v mnoha oblastech, jako je sledování pohybu, bezpečnostní dohled, analýza dopravy či autonomní řízení.

Rozpoznávání objektů je klíčovou součástí počítačového vidění a nachází široké využití v různých průmyslových odvětvích. Přesnost detekce se neustále zlepšuje díky optimalizaci modelů, rozšiřování trénovacích dat a využívání moderních výpočetních technologií, jako jsou grafické procesory (GPU) nebo akcelerátory umělé inteligence. Tyto inovace umožňují nasazení systémů rozpoznávání objektů i na zařízeních s omezeným výpočetním výkonem, čímž se otevírají nové možnosti využití této technologie v praxi.

3 Použité knihovny

3.1 OpenCV

OpenCV je rozsáhlá open-source knihovna zaměřená na zpracování obrazu a počítačové vidění. Tato knihovna, napsaná převážně v jazyce C++, nabízí více než 2500 optimalizovaných algoritmů pro analýzu obrazových dat. OpenCV poskytuje nástroje pro širokou škálu úkolů, jako je detekce objektů, rozpoznávání tvarů, analýza pohybu, zpracování videa a další. Díky své flexibilitě a rozsáhlým funkcím se stala jedním z nejpoužívanějších nástrojů v oblasti strojového vidění. V naší práci byla knihovna OpenCV klíčovým prvkem při integraci kamery s modelem pro rozpoznávání objektů. Umožnila nám efektivně získávat a zpracovávat obraz z připojené kamery, provádět předzpracování snímků a vizualizovat výsledky detekce v reálném čase.

Využívali jsme několik základních funkcí OpenCV pro zajištění optimálního výkonu systému. Pomocí ní jsme získávali živý obraz z kamery, zachytávali jednotlivé snímky pro další zpracování a prováděli různé úpravy obrazu – jako změnu velikosti, převod barevných prostorů nebo aplikaci filtrů. Po provedení detekce objektů jsme získané informace využili k vykreslení výsledků na obraz, tedy zvýraznění detekovaných objektů pomocí ohraňujících rámečků a popisků. Upravený obraz s výsledky detekce jsme následně zobrazovali na připojeném displeji, což umožnilo uživateli mít okamžitou zpětnou vazbu o rozpoznaných objektech.

Velkou výhodou OpenCV je její schopnost běžet i na výkonnostně omezených zařízeních, jako je Raspberry Pi, což bylo pro náš projekt zásadní. Kromě toho nabízí širokou podporu pro různé typy obrazových dat a snadnou integraci s dalšími nástroji. Díky dostupné dokumentaci a silné komunitě vývojářů je OpenCV velmi vhodnou volbou pro rychlou a efektivní tvorbu aplikací založených na počítačovém vidění. [3]

3.2 Ultralytics

Ultralytics je knihovna pro Python, která se zaměřuje na zjednodušení práce s modely detekce objektů, konkrétně na modely YOLO, jako je například YOLOv5 a YOLOv8. Tato knihovna poskytuje uživatelsky přívětivé API, které usnadňuje implementaci a nasazení modelů pro detekci objektů v reálném čase. Ultralytics umožňuje snadné načítání předtrénovaných modelů, jejich trénování na vlastních datech, detekci objektů v obrázcích a videích a vizualizaci výsledků. Díky své jednoduchosti a vysoké výkonnosti se Ultralytics stala populární volbou pro vývoj aplikací v oblasti počítačového vidění, jako je sledování objektů, rozpoznávání scén nebo analýza obrazu. Knihovna je kompatibilní s frameworkm PyTorch, což umožňuje její integraci s dalšími nástroji pro strojové učení a hluboké učení. Ultralytics je ideální pro vývojáře, kteří chtějí využít sílu moderních detekčních modelů bez nutnosti složitého nastavování nebo hluboké znalosti strojového učení. [4]

3.3 Tkinter

Tkinter je základní knihovna pro tvorbu grafického uživatelského rozhraní (GUI) v programovacím jazyce Python. Je součástí standardní výbavy Pythonu, což znamená, že ji není nutné dodatečně instalovat. Tkinter je postaven na nástroji Tk a slouží k vytváření okenních aplikací s vizuálními prvky, jako jsou tlačítka, textová pole, štítky, přepínače, nabídky nebo kreslicí plocha. Knihovna je vhodná především pro začátečníky, jelikož je přehledná a snadno použitelná. Pracuje na principu událostmi řízeného programování – aplikace reaguje na uživatelské akce, například kliknutí myší nebo stisk klávesy, a provádí předem definované funkce. Tkinter umožňuje rychlé vytvoření jednoduchých aplikací s interaktivním ovládáním a často se používá ve školních projektech, výuce nebo při prototypování. Pokud je potřeba modernější vzhled, lze využít rozšíření knihovny, například customTkinter, které přináší stylovější a vizuálně propracovanější prvky GUI. [5]

3.4 Knihovna TensorFlow

TensorFlow je open-source knihovna pro strojové učení vyvinutá společností Google Brain Team. Poskytuje flexibilní nástroje pro vytváření a nasazení modelů strojového a hlubokého učení. S využitím TensorFlow můžete vytvářet a trénovat různé typy modelů, včetně konvolučních a rekurentních neuronových sítí. Knihovna nabízí automatickou diferenciaci pro výpočet gradientů, efektivní výpočty na různých zařízeních (CPU, GPU, TPU) a nástroje pro distribuované učení. TensorBoard umožňuje vizualizaci dat a sledování průběhu tréninku. S rozsáhlou komunitou uživatelů a vývojářů poskytuje TensorFlow širokou škálu zdrojů a podporu pro vývoj modelů strojového učení.[1]

3.5 Knihovna Model Garden

Model Garden je knihovna vyvinutá v rámci projektu OpenAI, která poskytuje širokou škálu předtrénovaných modelů a nástrojů pro strojové učení. Obsahuje implementace nejnovějších a nejvýkonnějších modelů pro klasifikaci, generativní modely, zpracování přirozeného jazyka a další oblasti. Model Garden umožňuje výzkumníkům a vývojářům rychle začít pracovat s těmito modely, provádět jemné ladění a adaptaci na své vlastní úkoly, což přispívá k rychlejšímu vývoji a vylepšení výkonnosti modelů strojového učení v různých oblastech aplikací, a tím i k demokratizaci pokročilých technik umělé inteligence.[2]

4 Použitý model

4.1 Model YOLOv8

YOLOv8 je nejnovější model pro detekci objektů od společnosti Ultralytics. Oproti předchozím verzím přináší vyšší přesnost, rychlosť a univerzálnost díky modernizované architektuře. Pracuje na principu jednorázové detekce (single-shot detection), kdy analyzuje celý snímek v jednom průchodu a vrací souřadnice objektů, jejich kategorie a pravděpodobnosti.

Model je navržen tak, aby fungoval na různých zařízeních – od výkonných GPU po minipočítače jako Raspberry Pi 4. Nabízí více variant podle výkonu: od lehké YOLOv8n (nano) po výkonnější YOLOv8x (extra large). K integraci se využívá knihovna Ultralytics YOLO založená na framework PyTorch, která nabízí jednoduché prostředí pro trénování i použití modelu. Kromě detekce objektů podporuje také segmentaci a klasifikaci.

V našem projektu jsme YOLOv8 použili k detekci objektů v reálném čase z kamery připojené k Raspberry Pi 4. Snímky jsme zpracovávali pomocí knihovny OpenCV a výsledky detekce zobrazovali přímo na obrazovce. Pro optimalizaci výkonu jsme zvolili variantu YOLOv8n, která je vhodná pro slabší hardware.

Celý systém je jednoduchý na použití a nabízí spolehlivou detekci i při omezených výpočetních možnostech. YOLOv8 se tak stal klíčovou součástí našeho řešení pro zpracování obrazu v reálném čase.[6]

5 Uživatelké rozhraní

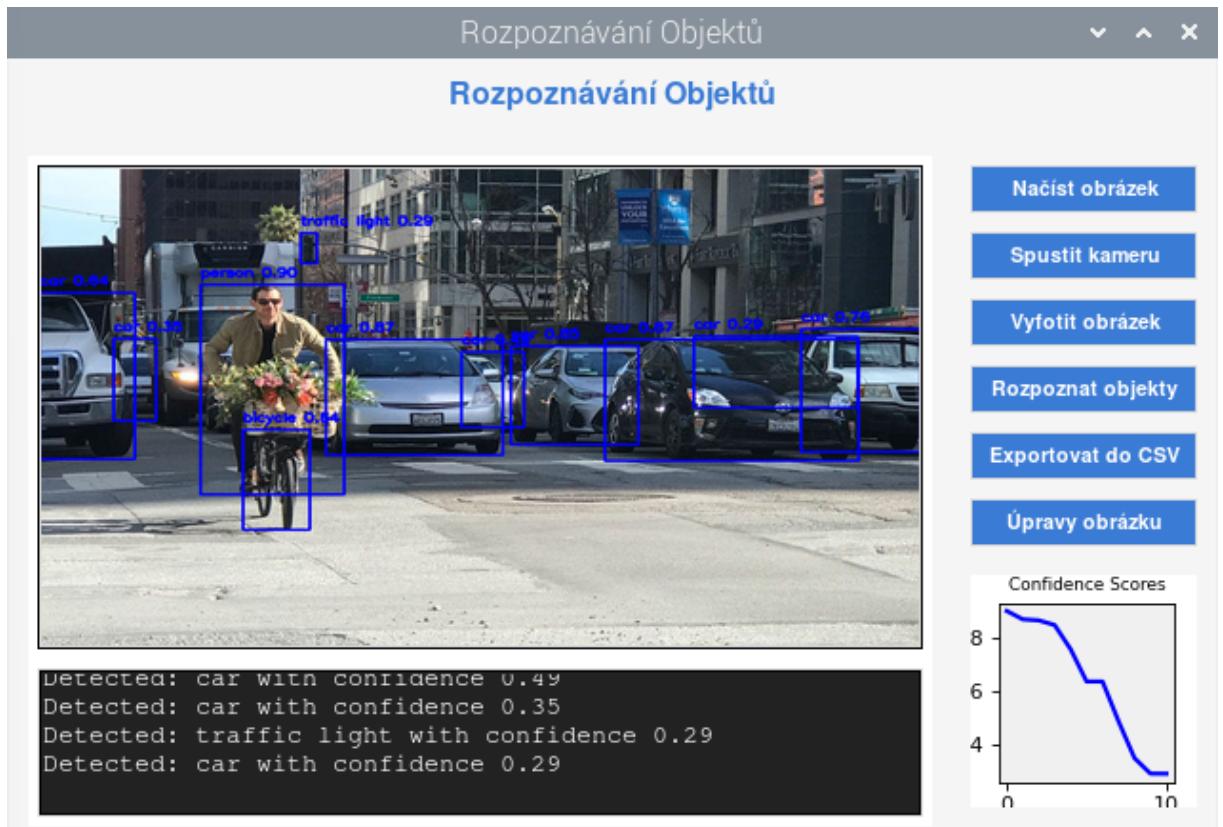
5.1 Návrh

Při návrhu grafického uživatelského rozhraní (GUI) pro systém rozpoznávání objektů bylo klíčové zajistit jednoduchost, přehlednost a intuitivní ovládání. Cílem bylo vytvořit prostředí, které umožní uživateli snadno spustit detekci objektů, zobrazit výsledky analýzy a případně upravit parametry detekce.

Aby GUI bylo uživatelsky jednoduché a přehledné, zvolil jsem variantu jednoduchých barev (bílé pozadí, modré funkční tlačítka), pokud bych zvolil příliš pestré barvy, mohlo by to být pro uživatele rušivé. Dalším důležitým aspektem v návrhu bylo, aby prostor sloužící pro záznam z kamery nebo pro nahrané obrázky, měl dostatečnou velikost. Rozhraní jsem navrhl jako jednoduchou aplikaci s hlavním oknem, které zobrazuje živý obraz z kamery, kde jsou detekované objekty označeny ohraničujícími rámečky (bounding boxes) s popisky. Vedle hlavního zobrazení jsou umístěny ovládací prvky v podobě funkčních tlačítek umožňující spuštění a zastavení detekce, načtení staženého souboru, úprava obrázku, rozpoznání objektů nebo export výsledků do csv souboru.

Celkově návrh GUI klade důraz na jednoduchost použití, rychlou odezvu a přehledné zobrazení výstupů, čímž umožňuje efektivní využití systému v reálném nasazení.

5.2 Finální podoba GUI



Obrázek 5.1: Zde je vidět finální podoba námi vytvořeného GUI. Je patrný velký prostor pro obrázek, či videozáznam přímo z kamery. Pod ním je umístěn termínál, kde se vypisují výsledky detekce a pravděpodobnost v cconfidence score. Pravá část aplikace patří funkčním talcítkům, které jsou v další části práce detailněji popsány, a grafu, který znázorňuje míru důvěryhodnosti detekce objektů. Tento graf zobrazuje vývoj hodnot pravděpodobnosti pro jednotlivé rozpoznávané objekty.

5.3 Realizace

V rámci maturitní práce jsem vyvinul grafické uživatelské rozhraní pro rozpoznávání objektů s využitím knihovny Ultralytics a modelu YOLOv8. Cílem bylo vytvořit uživatelsky přívětivou aplikaci, která umožní snadné načítání obrázků, detekci objektů a analýzu výsledků.

5.3.1 Funkcionalita a prvky GUI

Aplikace obsahuje několik klíčových prvků:

- Zobrazení obrázku s detekovanými objekty – Po načtení nebo pořízením snímku kamera je prováděna detekce objektů. Výsledky jsou vizualizovány přímo v obrázku, kde jsou kolem objektů vykresleny ohraničující boxy s popiskem a hodnotou pravděpodobnosti detekce.
- Tlačítka pro ovládání aplikace – Uživateli je umožněno načíst obrázek z disku, spustit kameru pro živou detekci, rozpoznat objekty v aktuálním snímku a exportovat výsledky do CSV souboru. Další funkce umožňuje provádět úpravy obrázku pro lepší vizualizaci a analýzu.
- Textová konzole – Slouží k zobrazení podrobného seznamu detekovaných objektů spolu s jejich pravděpodobnostmi. Tento prvek poskytuje přehledný výstup, který může být užitečný při vyhodnocování detekčního modelu.
- Graf znázorňující confidence scores – Grafické znázornění hodnot pravděpodobností detekovaných objektů umožňuje vizuální analýzu přesnosti modelu.

5.3.2 Technická realizace

Aplikace byla implementována v jazyce Python s využitím následujících knihoven:

- Tkinter - knihovna, která v GUI umožňuje snadno vytvářet a ovládat vizuální prvky aplikace, jako jsou tlačítka, textová pole, štítky, nabídky či grafické plochy. Tkinter zajistila propojení mezi Pythonem a GUI komponentami, čímž umožnila zobrazovat data, reagovat na uživatelské akce (např. kliknutí nebo zadání textu) a celkově ovládat chování aplikace pomocí událostmi řízeného programování. Posloužila jako nástroj k návrhu vzhledu i funkčnosti uživatelského rozhraní, včetně rozvržení prvků na obrazovce, jejich interakce a zpracování vstupů, což vývojářům umožňuje vytvářet přehledné, interaktivní a uživatelsky přívětivé aplikace. [5]

- OpenCV – rozsáhlá knihovna pro počítačové vidění a zpracování obrazu, která umožňula snadno pracovat s obrazovými daty v reálném čase. V kontextu GUI a detekčních aplikací slouží OpenCV k načítání, úpravě, analýze a vizualizaci obrázků a videí – například pro zachytávání obrazu z kamery, detekci hran, objektů nebo pohybu, či vykreslování výsledků. OpenCV se jevilo jako ideální volba, jelikož se často používá v kombinaci s dalšími knihovnami jako je Tkinter nebo Ultralytics pro tvorbu interaktivních aplikací s vizuálním výstupem v reálném čase. [3]
- Ultralytics – knihovna zaměřená na jednoduché použití modelu YOLO pro detekci objektů. V rámci našeho GUI Ultralytics zajistila komfortní rozhraní pro načítání trénovaných modelů, zpracování vstupních obrázků nebo videí a výstup predikcí (např. rozpoznaných objektů, jejich polohy a tříd). Funguje jako inteligentní nástroj, který kombinuje výkonnost detekčních modelů s jednoduchým API, čímž výrazně usnadňuje vývoj pokročilých aplikací v oblasti počítačového vidění. [4]

5.3.3 Funkční tlačítka

- Načíst obrázek – Toto tlačítko umožňuje uživateli vybrat a nahrát obrázek z počítače pro další zpracování. Po nahrání lze s obrázkem provádět analýzu a detekci objektů.
- Spustit Kameru – Aktivuje webovou kameru a umožňuje zachytit živý obraz v reálném čase. To je užitečné například pro analýzu scén přímo z kamery bez nutnosti nahrávat externí obrázky.
- Vyfotit obrázek – Po aktivaci kamery umožňuje uživateli pořídit snímek přímo z živého přenosu. Tento snímek je poté použit pro detekci objektů nebo další úpravy.
- Rozpozнат objekty – Spustí algoritmus detekce objektů na aktuálně zvoleném obrázku. Systém identifikuje různé objekty v obraze, přiřazuje jim kategorie a zobrazí jejich přesnost rozpoznání.
- Exportovat do CSV – Umožňuje uložit výsledky detekce objektů do souboru ve formátu CSV. Tento soubor může být později využit pro další analýzu nebo zpracování dat.
- Úpravy obrázku – Nabízí různé nástroje pro základní editaci nahraného nebo zachyceného snímku. Může zahrnovat změny jasu, kontrastu, oříznutí nebo jiné úpravy pro lepší rozpoznání objektů.

6 Testování

6.1 Testování zařízení

Testování zařízení bylo zaměřeno na ověření funkčnosti, přesnosti a výkonnosti objektové detekce v reálném čase pomocí modelu YOLOv8n nasazeného na zařízení Raspberry Pi 4 s připojenou webkamerou Logitech Brio 100. Cílem bylo zjistit, zda je možné provozovat spolehlivý detekční systém i na hardwarově omezené platformě.

Testování zařízení obsahovalo tyto oblasti:

- Přesnost detekce: Model byl testován na různých objektech a za různých světelných podmínek. Hodnotila se schopnost správné identifikace objektů, odolnost vůči falešným poplachům a stabilita při pohybu objektů. Výsledky potvrdily vysokou spolehlivost i v náročných scénách.
- Rychlosť a odezva systému: Sledovala se plynulost zpracování obrazu, rychlosť detekce a odezva GUI při přímém přenosu videa. I při omezeném výpočetním výkonu bylo dosaženo minimálního zpoždění a dostatečně plynulého chodu.
- Celkové hodnocení: Testy potvrdily, že systém splňuje požadavky na funkčnost, stabilitu a uživatelskou přívětivost. YOLOv8n se ukázal jako vhodné řešení pro detekci objektů na nízkonákladových zařízeních, jako je Raspberry Pi 4.

6.2 Testování GUI

V rámci vývoje systému bylo testováno také grafické uživatelské rozhraní, jehož cílem bylo umožnit snadnou a přehlednou obsluhu aplikace i méně zkušeným uživatelům. Testování GUI bylo zaměřeno především na funkčnost jednotlivých ovládacích prvků, správnou odezvu systému na uživatelské akce a celkovou stabilitu rozhraní.

Bylo ověřeno, že všechny základní prvky rozhraní – jako jsou tlačítka pro načtení obrázku, spuštění kamery, zachycení snímku, spuštění detekce, export výsledků či úpravu obrázku – reagují správně a provádějí požadované akce, které jsou zadány uživatelem. Dále byla kontrolována správná aktualizace obrazu, zobrazení detekčních výsledků a funkčnost grafické vizualizace výsledků.

Sekce	Spokojenost (%)	Počet uživatelů
Přehlednost	92%	10
Funkčnost	94%	10
Spuštění/Zastavení	96%	10
Zobrazení výsledků	91%	10
Vizualizace objektů	93%	10

Tabulka 6.1: Hodnocení uživatelského rozhraní (GUI) 10 uživateli

Testování uživatelského rozhraní (GUI) proběhlo za účasti deseti uživatelů, kteří hodnotili různé aspekty aplikace pomocí procentuální stupnice – vyšší procento vyjadřovalo vyšší míru spokojenosti. Pro každou hodnocenou sekci byl následně vypočten průměr spokojenosti.

Nejlépe hodnocenou částí se stala sekce pro spuštění a zastavení detekce s průměrnou spokojeností 96%, což odráží vysoké ocenění jednoduchého a intuitivního ovládání. Funkčnost aplikace obdržela 94 %, což svědčí o bezproblémovém chodu a stabilitě systému. Přehlednost uživatelského rozhraní získala 92 %, což potvrzuje, že rozmištění a organizace prvků jsou pro uživatele snadno pochopitelné. Vizualizace objektů dosáhla 93%, čímž uživatelé ocenili jasnost a srozumitelnost grafického znázornění. Zobrazení výsledků bylo ohodnoceno 91 %, což sice ukazuje na velmi dobrou úroveň, ale zároveň naznačuje možnost drobných zlepšení.

Celkové výsledky hodnocení potvrzují, že uživatelské rozhraní je kvalitně navržené, přehledné a snadno ovladatelné. Získaná zpětná vazba poukazuje na jeho uživatelskou přívětivost a intuitivnost.

6.3 Testování modelu

Pro testování modelu YOLOv8 byl použit dataset COCO (Common Objects in Context), který patří mezi nejrozsáhlejší a nejčastěji využívané datasety pro úlohy počítačového vidění. Obsahuje více než 200 000 obrázků s anotacemi pro detekci objektů, segmentaci i popisování scén, a představuje tak ideální základ pro trénování i hodnocení výkonnosti moderních detekčních modelů.

Dataset byl získán z oficiálního zdroje Ultralytics a následně upraven pro použití s modelem YOLOv8. Při přípravě byly využity nástroje dostupné v knihovně Ultralytics, které umožňují převod dat do formátu kompatibilního s tímto modelem.[7]

V rámci testování modelu YOLOv8 jsme se zaměřili na několik klíčových oblastí hodnocení:

- Přesnost detekce (mAP) – hodnotili jsme střední průměrnou přesnost detekovaných objektů na základě jednotlivých kategorií v datasetu COCO. Výsledky ukázaly, že model vykazuje vysokou míru přesnosti u většiny objektů, s mírně horšími výsledky u kategorií s větší rozmanitostí tvarů či velikostí.
- Rychlosť zpracování (FPS) – testovali jsme rychlosť predikce při různých vstupních rozlišeních. Výsledky potvrdily, že vyšší rozlišení sice přináší více detailů, ale zároveň zpomaluje zpracování a zvyšuje nároky na výpočetní výkon. Pro aplikace v reálném čase je proto vhodnější použití nižších rozlišení.
- Robustnost v různých podmínkách – testovali jsme také odolnost modelu vůči změnám scénických parametrů, jako jsou různé úhly pohledu, změny osvětlení nebo částečné překrytí objektů. Model prokázal schopnost adaptovat se i na méně ideální podmínky, i když s mírně nižší přesností oproti standardním scénám.

7 Výsledky testování

7.1 Úspěšné testování

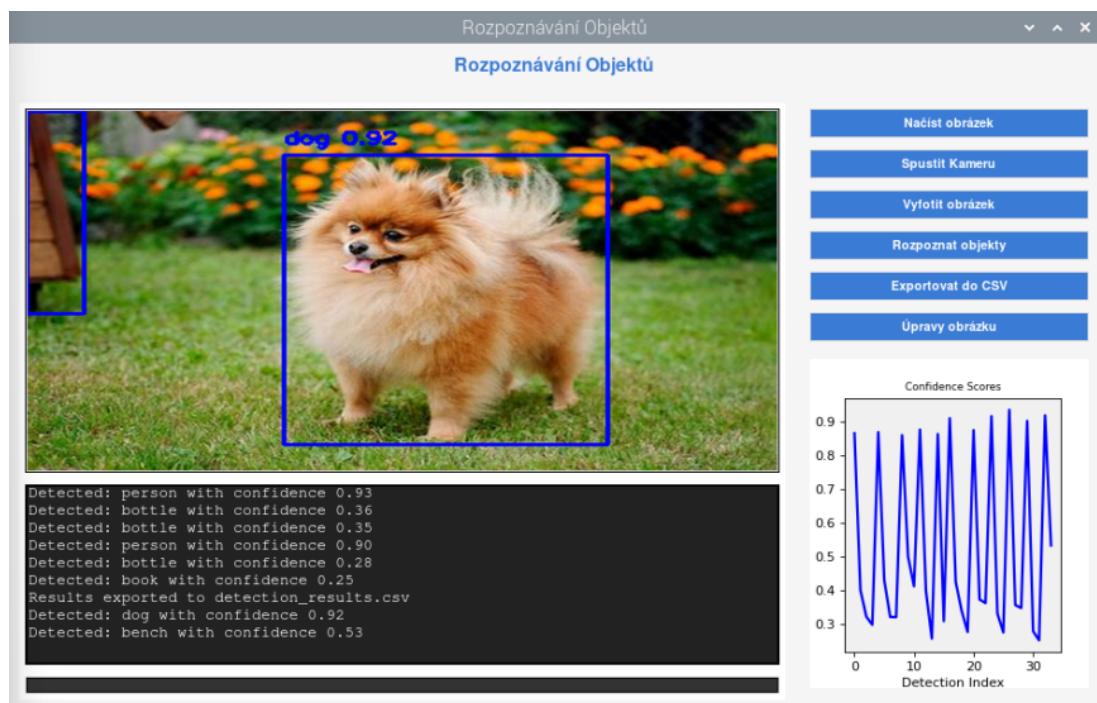
Během úspěšného testování detekce objektů pomocí modelu YOLOv8n na zařízení Raspberry Pi 4 bylo ověřeno, že tento systém je plně funkční a použitelný i pro náročnější úlohy strojového vidění. Detekce probíhala plynule v reálném čase a model si spolehlivě poradil s rozpoznáváním různých typů objektů v různých prostředích, za různých světelných podmínek i pod různými úhly pohledu. Výsledky ukázaly vysokou přesnost detekce a stabilitu systému. Klíčovým ukazatelem byla hodnota score, která vyjadřuje míru jistoty modelu při rozpoznání objektu. Celkově lze říci, že testování dopadlo velmi dobře a potvrzuje vhodnost tohoto řešení pro praktické využití.

Grafické uživatelské rozhraní bylo úspěšně otestováno po stránce funkčnosti i stability. Všechna tlačítka a ovládací prvky reagovala správně na uživatelské vstupy a prováděla požadované akce. Detekované objekty se správně zobrazovaly jak ve formě označených oblastí v obraze, tak ve výpisu výsledků. Součástí rozhraní byla i grafická vizualizace přesnosti, která fungovala bez problémů. Během úspěšného testování se GUI chovalo stabilně a přehledně, bez výskytu chyb nebo pádů. Bylo potvrzeno, že rozhraní je vhodné i pro běžné uživatele a umožňuje snadnou a intuitivní obsluhu celého systému.

Tabulka 7.1 zachycuje výsledky detekce objektů modelem YOLOv8 na jednom snímku. Každý řádek představuje jeden rozpoznaný objekt a uvádí jeho název (label), souřadnice ohraňujícího rámečku (levý horní a pravý dolní roh – Left, Top, Right, Bottom) a míru jistoty modelu (Score), s jakou byl objekt identifikován. Tyto údaje umožňují určit, co bylo v obraze detekováno, kde se jednotlivé objekty nacházejí a s jakou přesností byly rozpoznány.

Label	Left	Top	Right	Bottom	Score
Sklenička	63	199	205	376	0,68
Stůl	-5	127	622	479	0,56
Nůž	341	250	592	419	0,46
Kniha	94	127	402	219	0,43
Lednice	259	0	486	123	0,28

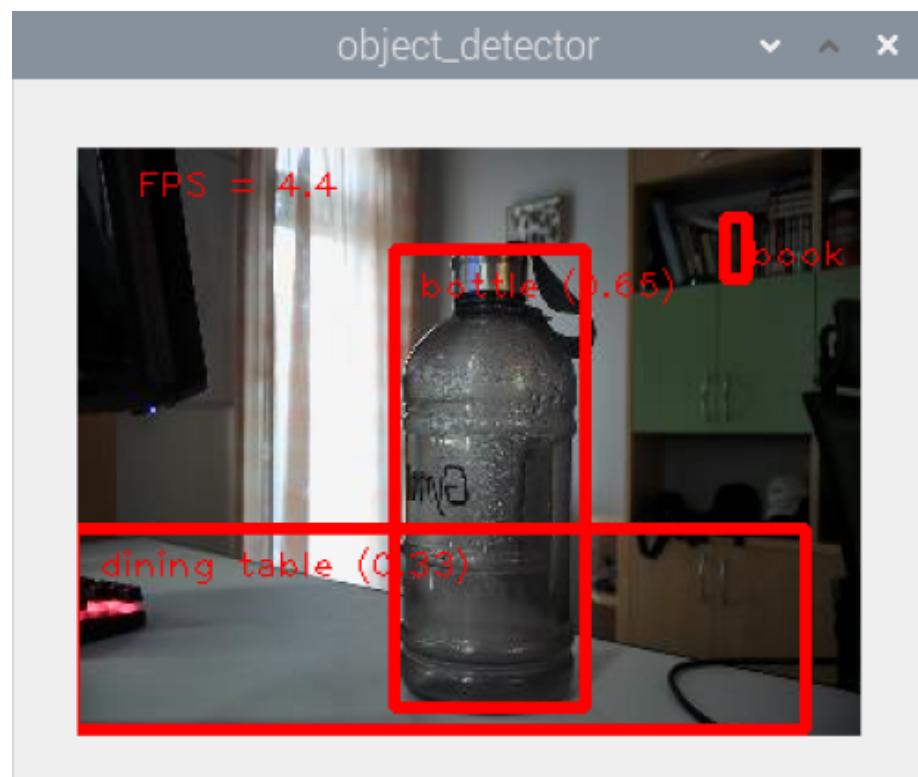
Tabulka 7.1: Tabulka výsledků testu rozpoznávání objektů



Obrázek 7.1: testovaný obrázek č. 1



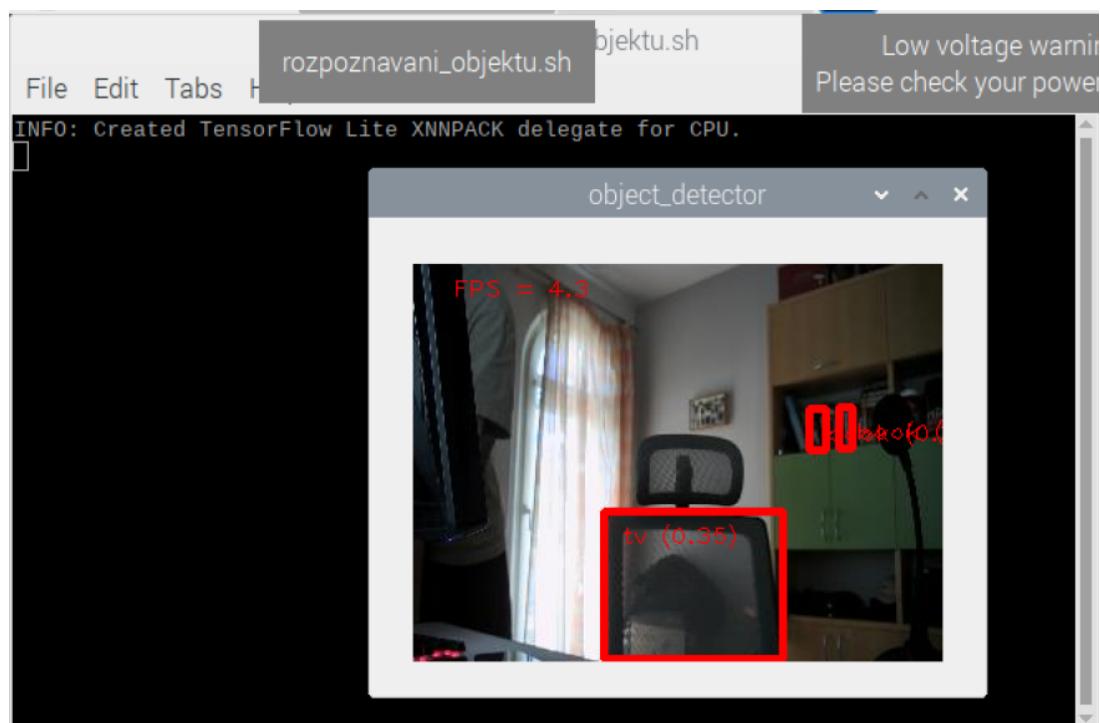
Obrázek 7.2: testovaný obrázek č. 2



Obrázek 7.3: testovaný obrázek č. 3

7.2 Neúspěšné testování

Při testování modelu YOLOv8 na Raspberry Pi 4 jsme narazili na značné problémy s rozpoznáváním objektů. I když jsme postupovali podle doporučených pokynů pro instalaci a použili nejnovější verzi softwaru, algoritmus pro detekci objektů někdy fungoval nepravidelně. Během testování naší aplikace pro rozpoznávání objektů jsme zjistili, že Raspberry Pi 4 s TensorFlow nedokázal spolehlivě identifikovat objekty při zpracování videa. Například při pokusu o rozpoznání židle na záznamu z kamery byl výsledek často nepřesný nebo zcela chybný (viz obrázek 10). I při použití širokého spektra testovacích scénářů se nám nepodařilo dosáhnout úplné spolehlivosti při rozpoznávání objektů. Problém byl způsoben kombinací hardwarových omezení Raspberry Pi 4 a možnými nedostatky v implementaci TensorFlow pro tuto platformu. Navzdory několika pokusům o optimalizaci a ladění parametrů TensorFlow jsme nedosáhli uspokojivých výsledků pro reálné použití v praxi.



Obrázek 7.4: testovaný obrázek č. 4

8 Návod na použití

Toto zařízení slouží k automatickému rozpoznávání objektů v reálném čase pomocí modelu YOLOv8. Zařízení je postaveno na platformě Raspberry Pi 4 a je vybaveno kamerou a dotykovým displejem. Ovládání probíhá přes jednoduché a intuitivní grafické uživatelské rozhraní.

- Spuštění zařízení

Připojte napájení do Raspberry Pi a vyčkejte, dokud se systém zcela nespustí. Po spuštění operačního systému se na displeji objeví ikona s GUI.

Ujistěte se, že je kamera správně připojena a funkční.

- Ovládací prvky GUI

Po spuštění aplikace se zobrazí hlavní obrazovka s několika tlačítky.

V tuto chvíli je zařízení připraveno reagovat podle potřeby uživatele.

- Práce se zařízením

Po spuštění detekce se na obrazovce začne zobrazovat živý přenos z kamery, ve kterém budou automaticky označeny rozpoznané objekty rámečky spolu s popisky a pravděpodobností detekce.

Pravá část aplikace dává uživateli více možností, jakým způsobem chce provádět detekci objektů nebo jaký typ souboru chce rozpoznávat.

- Doporučení pro použití

Pro co nejpřesnější detekci zajistěte dobré světelné podmínky.

Ujistěte se, že je kamera namířena na oblast s objekty, které mají být detekovány.

Nedotýkejte se kamery během detekce – pohyb může ovlivnit stabilitu rozpoznávání.

Pokud zařízení reaguje pomalu, snižte rozlišení obrazu.

- Řešení problémů

Kamera nezobrazuje obraz – zkontrolujte připojení kamery a znovu spusťte aplikaci.

Nízký výkon nebo zpoždění detekce – snižte rozlišení vstupu nebo zjednodušte scénu před kamerou.

Nereagující dotyková obrazovka – zkuste k zařízení připojit počítačovou myš, vzhledem k nedokonalé kalibraci displeje může docházet ke špatnému vnímání dotyku.

Závěr

Cílem této práce bylo vytvořit funkční systém pro rozpoznávání objektů v reálném čase s využitím modelu YOLOv8 na platformě Raspberry Pi 4 a přehledného grafického uživatelského rozhraní. Systém jsme úspěšně integrovali tak, aby umožňoval detekci objektů přímo ze živého video přenosu. Díky přehlednému a intuitivnímu GUI je uživatelské rozhraní snadno ovladatelné a poskytuje zpětnou vazbu v reálném čase. Tento systém efektivně kombinuje strojové učení, počítacové vidění a přívětivé ovládání, což ho činí vhodným pro praktické použití v různých aplikacích, kde je třeba detektovat objekty v reálném čase.

Během vývoje jsme narazili na několik omezení. Jedním z hlavních problémů bylo ovládání dotykového displeje, který není dokonale kalibrován, což vedlo k nižší přesnosti při práci s menšími prvky uživatelského rozhraní. Tento problém ovlivnil uživatelský komfort, zejména při interakci s jemnějšími ovládacími prvky. Ačkoliv dotykové ovládání stále umožňovalo používání systému, pro profesionální využití by bylo nutné zajistit lepší kalibraci displeje. Dalším omezením byl výkon Raspberry Pi 4, který je dostatečný pro základní detekci, ale při použití vyšších rozlišení obrazu nebo složitějších scénách došlo k poklesu výkonu a zpomalení zpracování.

Přesnost detekce byla ovlivněna faktory, jako jsou světelné podmínky, kvalita vstupního obrazu a podobnost objektů mezi jednotlivými kategoriemi. Tyto faktory mohou způsobit variabilitu v přesnosti detekce, zejména v reálných podmínkách, kde se podmínky často mění. Například špatné osvětlení nebo objekty, které jsou podobné jiným třídám, mohou snížit úspěšnost detekce. Přesto, i přes tato omezení, považujeme celý projekt za úspěšný. Podarilo se nám vytvořit funkční a efektivní řešení, které ukazuje možnosti využití neuronových sítí a strojového vidění i na omezeném hardwaru, jako je Raspberry Pi 4. Projekt demonstруje, že s správně optimalizovaným systémem lze dosáhnout výborných výsledků i při omezených výpočetních schopnostech zařízení.

Literatura

- [1] GOOGLE. *TensorFlow: An end-to-end open source machine learning platform.* [online]. [cit. 2025-29-03]. Dostupné z: <https://www.tensorflow.org>.
- [2] GOOGLE. *Model Garden: A collection of pre-trained models for TensorFlow.* [online]. [cit. 2025-29-03]. Dostupné z: <https://github.com/tensorflow/models>.
- [3] OPENCV. *OpenCV: Open Source Computer Vision Library.* [online]. [cit. 2025-29-03]. Dostupné z: <https://opencv.org>.
- [4] ULTRALYTICS. *Ultralytics : State-of-the-art object detection and segmentation.* [online]. [cit. 2025-30-03]. Dostupné z: <https://github.com/ultralytics/ultralytics>.
- [5] PYTHON SOFTWARE FOUNDATION. *tkinter—Python interface to Tcl/Tk.* [online]. [cit. 2025-29-03]. Dostupné z: <https://docs.python.org/3/library/tkinter.html>.
- [6] YOLOv8. *YOLOv8: State-of-the-art object detection and segmentation.* [online]. [cit. 2025-29-03]. Dostupné z: <https://github.com/ultralytics/yolov8>.
- [7] LIN, Tsung-Yi, Michael MAIRE, Serge BELONGIE, Lubomir BOURDEV, Ross GIRSHICK, James HAYS, Pietro PERONA, Deva RAMANAN, C. Lawrence ZITNICK a Piotr DOLLÁR. *Microsoft COCO: Common Objects in Context.* [online]. [cit. 2025-03-29]. Dostupné z: <https://docs.ultralytics.com/datasets/detect/coco/>.

Seznam obrázků

Obrázek 1.1: Raspberry Pi 4 (zdroj: <https://bit.ly/4bj53n5>)

Obrázek 1.2: PiTFT Plus 3,5 palce (zdroj: <https://bit.ly/3QqHUap>)

Obrázek 1.3: Logitech Brio 100 (zdroj: <http://bit.ly/4iVWCCq>)

Obrázek 1.4: MicSDXC Kingston (zdroj <https://bit.ly/3WlfHFR>)

Obrázek 1.6: Finální zrealizované zařízení (zdroj: vlastní)

Obrázek 5.1: Finální podoba GUI(zdroj: vlastní)

Obrázek 7.1: Testovaný obrázek č.1(zdroj: vlastní)

Obrázek 7.2: Testovaný obrázek č.2 (zdroj: vlastní)

Obrázek 7.3: Neúspěšně Testovaný obrázek č.3 (zdroj: vlastní)

Obrázek 7.4: Neúspěšně Testovaný obrázek č.4 (zdroj: vlastní)