

Určování poloh zájmových objektů ve scéně

Determining the positions of objects of interest in a scene

Bc. Daniel Slavík

Vedoucí práce: Ing. Tomáš Fabián, Ph.D.

Ostrava, 2025

Obsah

Seznam použitých symbolů a zkratek	3
Seznam obrázků	4
Seznam tabulek	5
1 Úvod	6
2 Přehled metod	7
2.1 YOLOv11	7
2.2 Perspektivní problém n bodů	8
3 Dataset	11
4 Návrh řešení	13
4.1 Tvorba datasetu	13
4.2 Trénink modelu	14
4.3 Odhad pózy	14
5 Výsledky	15
5.1 Výkon modelu	15
5.2 Výkon na reálných datech	16
6 Závěr	18
Literatura	19
Přílohy	19

Seznam použitých zkratk a symbolů

DNN	– Hluboká neuronová síť, Deep Neural Network
PnP	– Perspektivní problém n bodů, Perspective-n-Point
YOLO	– Podíváš se pouze jednou, You Only Look Once
RANSAC	– Náhodný výběr s konsenzem, Random Sample Consensus
VRAM	– Video paměť, Video Random Access Memory
OKS	– Podobnost klíčových bodů, Object Keypoint Similarity
mAP	– Průměrná přesnost, Mean Average Precision

Seznam obrázků

3.1	Příklady syntetických snímků z datasetu	12
3.2	Příklad snímku ze syntetického datasetu	12
4.1	Ukázka transformovaného snímku ze syntetického datasetu	14
5.1	Lokalizační metriky YOLO modelu	15
5.2	Odhad pózy modelu Hot Rod 42022	17
5.3	Nepřesný odhadu pózy modelu Hot Rod 42022	17

Seznam tabulek

5.1	Přesnost detekce a lokalizace modelu YOLOv11-s	16
5.2	Rychlost inference modelu YOLOv11-s	16

Kapitola 1

Úvod

Určování polohy objektů ve scéně patří mezi klíčové problémy současné počítačové grafiky a zpracování obrazu. Jedním z častých přístupů k jeho řešení je lokalizace klíčových bodů, tedy specifických bodů nacházejících se na objektu zájmu, které společně nesou zásadní informace o jeho tvaru, orientaci a poloze. V úspěšném případě lze odhadnout 6 stupňů volnosti (DoF) zájmového objektu.

V rámci této semestrální práce bude proveden pokus o nalezení klíčových bodů pomocí hlubokých neuronových sítí, případně detektorů, jako je YOLO – jeden z nejpoužívanějších modelů pro detekci objektů, tedy úlohu spočívající v nalezení objektů ve scéně a jejich lokalizaci pomocí tzv. ohraničujících boxů (ang. bounding boxes). Následné řešení bude evaluováno a vyobrazeno pomocí grafického rozhraní pro vizualizaci odhadu vzájemné pózy objektu ke kameře. Práce navazuje na postupy a řešení představené v mé bakalářské práci [1], přičemž jejím cílem je dále vylepšit navržený přístup, ověřit jeho funkčnost na reálných snímcích a následně provést odhad polohy objektu pomocí řešení úlohy perspektivního problému n bodů.

Kapitola 2

Přehled metod

2.1 YOLOv11

YOLOv11 (ang. You Only Look Once, verze 11) představuje jeden z posledních vývojových stupňů v řadě modelů pro úlohy počítačového vidění. Jedná se o vylepšení předcházející populární verze YOLOv8. Modely rodiny hlubokých konvolučních neuronových sítí YOLO přistupují k detekci objektů jako k regresnímu problému – tedy predikují jak třídu objektu, tak jeho pozici (bounding box) v rámci jediné dopředné propagace skrz síť. To umožňuje velmi rychlé a zároveň přesné zpracování obrazu v reálném čase.

YOLOv11 však není omezeno pouze na klasickou detekci objektů. Díky své modulární architektuře podporuje také segmentaci, detekci orientovaných objektů (Oriented Bounding Boxes), klasifikaci snímků a zejména *pose estimation* – tedy odhad pozic klíčových bodů na objektu, což je přímo relevantní pro úlohu určování polohy objektu pomocí PnP algoritmu. Architektura YOLOv11, podobně jako předešlé verze, je tvořena třemi základními komponentami:

- **Páteřní síť** – slouží jako primární extraktor rysů z obrazu. Pomocí konvolučních vrstev převádí vstupní obraz na vícerozměrné mapy rysů v několika rozměrových měřítkách.
- **Krk** – představuje mezistupeň, který agreguje rysy z různých hloubek sítě (tedy různých úrovní rozlišení) pomocí operací typu upsampling a concatenace. YOLOv11 dále zavádí prostorový mechanismus pozornosti (C2PSA), který umožňuje modelu zaměřit se na klíčové části obrazu – to je výhodné například při detekci částečně překrytých nebo malých objektů.
- **Hlava** – zodpovídá za finální predikci výstupů. Na základě zpracovaných map rysů predikuje souřadnice ohraničujících boxů, skóre přítomnosti objektu a pravděpodobnosti tříd. V případě varianty YOLOv11-Pose navíc model predikuje také souřadnice klíčových bodů na objektu, které lze dále využít např. pro řešení úlohy PnP.

Mezi hlavní změny v architektuře této verze detektoru YOLO patří:

- **C3k2 bloky** – nová varianta tzv. Cross Stage Partial bloků s menší konvolučním jádrem, která zlepšuje extrakci jemných detailů při nižší výpočetní náročnosti. Nahrazuje C3k bloky z předchozích iterací modelů za tuto rychlejší verzi, kde se nahrazuje větší konvoluční kernel za 2 menší (viděny ve verzi 8), což zlepšuje výkon.
- **SPPF (Spatial Pyramid Pooling – Fast)** – rychlejší varianta klasického SPP, umožňující agregaci vícerozměrných kontextových informací.
- **C2PSA (Convolutional block with Parallel Spatial Attention)** – paralelní mechanismus prostorové pozornosti, který pomáhá lépe zachytit důležité lokální i globální rysy v obraze [2].

2.2 Perspektivní problém n bodů

Problém perspektivního problému n bodů lze definovat jako odhad polohy objektu vůči kameře. Cílem je určit 6 stupňů volnosti – tři pro rotaci a tři pro translaci – které řeší problém nalezení takové rotace a translace, která minimalizuje reprojekční chybu mezi odpovídajícími si 3D body a jejich 2D projekcemi. Pro řešení PnP problému musíme znát klíčové body známého objektu v 3D souřadném systému objektu a také kalibrační parametry kamery, zejména tedy ohniskovou vzdálenost f . Základní forma transformace ze světového souřadného systému do projekční roviny může být definována následovně:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{A}\Pi^0\mathbf{T}_w \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}, \quad (2.1)$$

resp.

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}, \quad (2.2)$$

kde A definuje matici kamery – parametry f definují ohniskovou vzdálenost a c střed snímku. Rotační 3×3 parametry r a translační vektory t matice T_w transformují souřadnice ze světového souřadného systému do souřadného systému kamery, podrobněji znázorněno následovně:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (2.3)$$

Vstupem algoritmu pro řešení jsou tedy známé 3D souřadnice bodů objektu, jejich odpovídající 2D projekce na snímku a vnitřní matice kamery, zatímco výstupem je aproximace transformační matice T_w , také často definované i v její rozložené formě jako *tvec* a *rvec* [3].

2.2.1 Metody řešení

Metod pro řešení perspektivního problému n bodů je hned několik. Jednou z nejjednodušších a nejstarších metod může být **P3P**. Metoda P3P dokazuje, že 3 přesné odhady a korespondence klíčových bodů jsou dostatečné k odhadu pózy objektu vůči kameře. Přístupů P3P existuje několik, nejstarší metoda se dá vystopovat až do roku 1841. Metody P3P jsou primárně založeny na soustavě rovnic a úhlech mezi projekčními paprsky a známými vzdálenostmi mezi body 3D objektu. Bohužel metody P3P jsou citlivé na odlehle hodnoty ve vstupních projekcích. Také produkují až 4 možná řešení, avšak čtvrtý bod může být použit k odstranění nejasnosti [4].

Dalším významným přístupem k řešení PnP problému, na který se dále v této práci zaměříme, jsou **iterativní metody**, jako je ta implementovaná v knihovně OpenCV pod příznakem `cv.SOLVEPNP_ITERATIVE`. Tento přístup funguje jako zpřesnění pózy pomocí nelineární minimalizace nejmenších čtverců reprojekční chyby, konkrétně s využitím Levenberg-Marquardtova algoritmu. Tento algoritmus je iterativní a kombinuje dva přístupy:

1. **Gradientní sestup** – Pokud je aktuální odhad parametrů pózy (rotace a translace) daleko od optimálního řešení (což se projeví vysokou reprojekční chybou), Levenberg-Marquardtův algoritmus se chová podobně jako metoda největšího spádu. Toho dosahuje zvýšením tlumicího faktoru λ , což vede k menším, opatrnějším krokům ve směru nejstrmějšího poklesu reprojekční chyby.
2. **Gauss-Newtonův algoritmus** – Pokud je aktuální odhad parametrů pózy blízko optimálního řešení, Levenberg-Marquardtův algoritmus se chová podobně jako Gaussova-Newtonova metoda. Snížením tlumicího faktoru λ algoritmus využívá aproximaci problému k výpočtu větších a rychlejších kroků směřujících k minimu reprojekční chyby, což umožňuje rychlou konvergenci.

Levenberg-Marquardtův algoritmus mezi těmito dvěma chováními dynamicky přechází úpravou tlumicího faktoru λ na základě úspěšnosti předchozí iterace, počet iterací je dán jako hyperparametr funkce [5].

2.2.2 RANSAC

RANSAC (ang. **R**ANdom **S**Ample **C**onsensus, česky "náhodný výběr s konsenzem") – je iterativní algoritmus určený k aproximaci parametrů matematického modelu z datové sady, která obsahuje potenciálně nezanedbatelný podíl odlehklých hodnot. Jeho využití spočívá v iterativním nalezení nej-

lepší aproximace modelu pomocí vybraných minimálních podmnožin dat a následném vyhodnocení kvality každého takto vzniklého modelu na celé datové sadě.

Prvním krokem iterace je náhodný výběr podmnožiny bodů z našich dat. V kontextu metody RANSAC je to právě ta nejmenší velikost podmnožiny k úspěšnému řešení modelu, jelikož s klesajícím počtem datových bodů bude intuitivně růst pravděpodobnost, že náhodně vybraná podmnožina bude obsahovat výhradně vnitřní pozorování. Pokud se tak stane, model odvozený z této „čisté“ podmnožiny bude dobrým kandidátem na skutečný model. RANSAC tento proces provede iterativně:

1. Náhodně vybere minimální počet bodů potřebných k odhadu modelu.
2. Z této sady odhadne parametry modelu.
3. Ověří, kolik všech bodů v původní datové sadě tomuto modelu odpovídá s určitou tolerancí.

Po mnoha takových iteracích je za nejlepší odhad modelu prohlášen ten model, který byl podpořen největším počtem vnitřních pozorování. Odlehlé hodnoty, které žádnému z úspěšných modelů neodpovídají, jsou tímto způsobem efektivně ignorovány. V kontextu PnP problému se RANSAC používá k robustnímu nalezení transformační matice T_w i v přítomnosti chybných korespondencí klíčových bodů [5].

Kapitola 3

Dataset

Dataset pro tuto práci je vygenerován synteticky, pozadí jsou focena kamerou StereoLabs ZED 2 a popředí synteticky vykresleným modelem auta LEGO Hot Rod 42022. Pro tvorbu syntetického datasetu, augmentaci dat a transformaci augmentovaných pozic klíčových bodů/ohraničujících boxů byla využita knihovna `albumenations` v rámci jazyka Python 3.

V rámci několika světelných nastavení bylo nafoceno v auto-kalibračním režimu kamery pozadí pro generování dat ve formátu PNG s následujícími parametry: `1920×1080`, `8-bit/color RGB`, `non-interlaced`. Nutno podotknout, že při reálném použití by v daný moment platné parametry kamery sloužily ke generování dvou vstupů do funkce `solvePnP`, a to matice kamery a parametru `distCoeffs` do funkce:

- Ohnisková vzdálenost – f_x, f_y
- Hlavní body – c_x, c_y
- Zkreslení objektivu – k_1, k_2, k_3, p_1, p_2

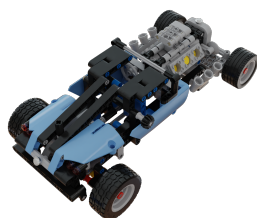
Objektem zájmu v této práci je 3D model LEGO automobilu, který je identický vůči tomu použitému v předchozí bakalářské práci [1]. Zde je dostupná sada syntetických renderovaných snímků celého modelu v různých natočeních kamery soustředěné na střed objektu. Sada obsahuje identické snímky v 5 různých virtuálních nastaveních, každé s odlišnými světelnými podmínkami. Součástí modelu jsou také pozice klíčových bodů v lokálním souřadném systému společně s jejich normálami. Každý snímek vykresleného modelu má taky v páru se sebou uložen `NumPy/pickle` slovník s následujícími metadaty:

- MV matice
- MVP matice
- Pozice kamery v kartézském souřadném systému

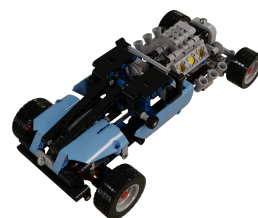
Normály společně s těmito metadaty nám v tomto kontextu mohou posloužit jako praktický indikátor viditelnosti.



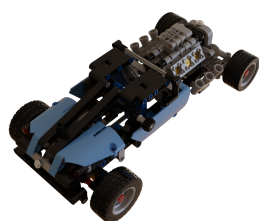
(a) EA408



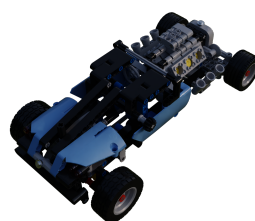
(b) EA408 světlé



(c) Scéna typu `empty_warehouse`

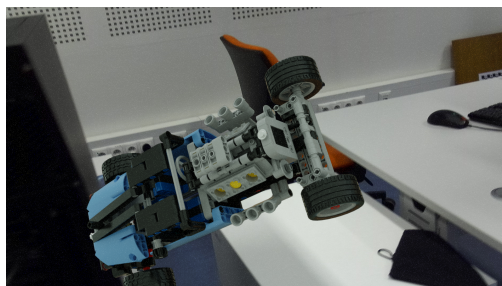


(d) Scéna typu `lebonbo`



(e) Scéna typu `lilienstein`

Obrázek 3.1: Příklady syntetických snímků z datasetu



Obrázek 3.2: Příklad snímku ze syntetického datasetu

Kapitola 4

Návrh řešení

Pro řešení lokalizace klíčových bodů a jejich korespondenci s 3D klíčovými body jsem se rozhodl využít model YOLOv11. Důvodem byla jeho přímá návaznost na populární verzi YOLOv8 a fakt, že YOLOv11 dle oficiální dokumentace nabízí rychlejší a přesnější inferenci, zejména v režimu odhadu pózy na COCO datasetu, což je pro naši úlohu relevantní [6]. Klíčové body datasetu jsou definovány bodovými souřadnicemi v kartézském souřadném systému. Ohraničující boxy lze k našim datům snadno vygenerovat, jelikož pozadí je na snímcích vykresleného modelu LEGO auta definováno nulovým alfa kanálem.

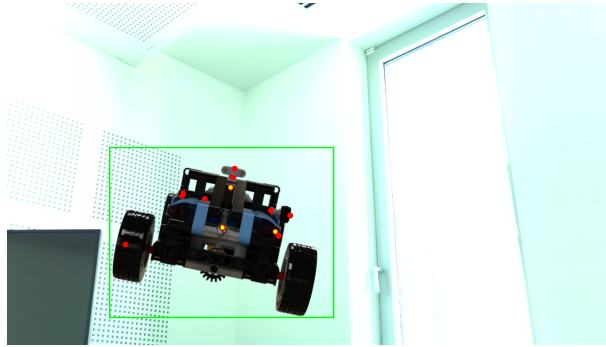
4.1 Tvorba datasetu

Dataset byl rozdělen v poměru 70:15:15. Každý snímek ze syntetického datasetu vygenerovaný pro danou sadu byl tvořen snímkem pozadí i zájmového objektu dostupného pouze pro danou sadu. Po vygenerování datasetu byl počet finálních snímků 9009:1929:1939, kde každé vykreslení auta v jednom z 5 možných scén bylo použito právě jedenkrát pro každý výsledný snímek ze syntetického datasetu. K augmentaci dat byl navržen dvouprůchodový proces, kde první stupeň provedl afinní transformace modelu auta a druhý stupeň vizuální augmentaci snímku ze syntetického datasetu. Součástí tohoto procesu byla také transformace pozic klíčových bodů a ohraničujících boxů.

V datasetu jsou zahrnuty následující augmentace:

- **Afinní transformace modelu** – otáčení, posun, změna velikosti
- Náhodná změna jasu, kontrastu, sytosti a tónu
- Náhodná úprava gamy (světlosti/tmavosti)
- Vyrovnání histogramu (kontrast)
- Rozmazání a zaostření

- Gaussovský šum, ISO šum, JPEG/WebP kompresní chyby
- Atmosférické efekty – Mlha, sníh, déšť
- Optické zkreslení kamery



Obrázek 4.1: Ukázka transformovaného snímku ze syntetického datasetu, červené body jsou klasifikovány jako neviditelné, zatímco žluté jsou v dostatečně nízkém úhlu pohledu vůči kameře ($<75^\circ$).

4.2 Trénink modelu

Pro trénink modelu byly snímky ze syntetického datasetu zmenšeny na velikost 640×384 px, a to kvůli výpočetní náročnosti při tuningu modelu na kartě GTX 1070 s 8 GB VRAM. Tato volba je také podpořena dokumentací a je to častou volbou velikosti vstupů modelů YOLO a také jako kompromis kvality a výpočetní náročnosti [7]. Velikost dávky (ang. batch size) bude nastavena na přirozený úsudek vůči dostupné grafické paměti na 16. Velikost modelu pro první experiment bude použit model verze *small* (2 nejmenší z 5 velikostí modelu).

4.3 Odhad pózy

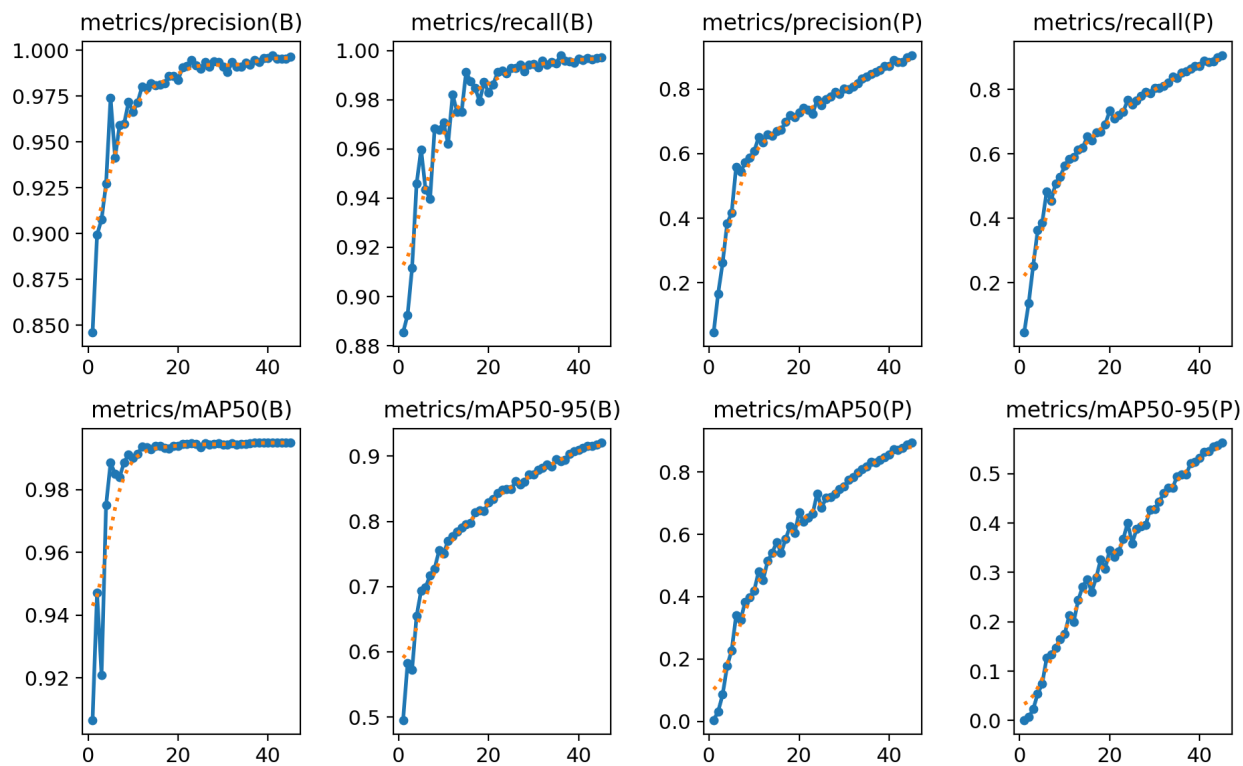
Pro následný odhad pózy bude využita metoda PnP společně s metodou RANSAC v rámci knihovny OpenCV. Metoda RANSAC je zde patřičně užitečná díky představení odolnosti proti potenciálně nepřesným odhadům klíčových bodů. Jako první nastavení volby algoritmu pro řešení problému n bodů byl zvolen iterativní Levenberg-Marquardtův algoritmus. Hyperparametry jako počet iterací, maximální reprojekční chyba nebo minimální spolehlivost výsledku mohou být doladěny dle potřeby. Distorzní koeficienty by v ideálním případě měly být přizpůsobeny kameře. V případě využití na proudu dat či na inferenci v reálném čase je zde pro Levenberg-Marquardtův algoritmus dostupná volba `useExtrinsicGuess` pro nastavení prvotního odhadu matice `tvec` a `rvec` [8].

Kapitola 5

Výsledky

5.1 Výkon modelu

Trénink modelu YOLO byl na základě prvotních experimentů nastaven na 45 epoch s následujícími výsledky viděnými na obrázku 5.1.



Obrázek 5.1: Lokalizační metriky YOLO modelu

Z obrázku 5.1 je patrné, že metriky pro lokalizaci klíčových bodů se stále inkrementálně zlepšovaly. Bohužel, v rámci této práce se po dosažení přednastavené hodnoty trénovaných epoch nepovedlo navrátit stav optimalizátoru a plánovače rychlosti učení do stavu, kdy by mohl trénink efektivně a spolehlivě pokračovat.

V rámci evaluace lokalizace byl parametr σ pro metriku OKS nastaven na $\sigma = 0,05$. Volba hodnoty je v rozsahu hodnot σ z dat pro odhad lidské pózy COCO [9]. Výsledky na základě testovací množiny jsou následující:

Detekce (Box)		Odhad pózy (Pose)	
Metrika	Hodnota	Metrika	Hodnota
Přesnost (P)	0,995	Přesnost (P)	0,757
Úplnost (R)	0,991	Úplnost (R)	0,732
mAP@0,5	0,995	mAP@0,5 OKS	0,730
mAP@0,5:0,95	0,909	mAP@0,5:0,95 OKS	0,435

Tabulka 5.1: Přesnost detekce a lokalizace modelu YOLOv11-s

Z tabulky 5.1 je patrné, že detekce a odhad ohraničujících boxů dosahuje vysokých hodnot všech měřených metrik přesnosti, úplnosti a mAP. Avšak výsledky jsou nejzajímavější pro odhad pózy (lokalizaci), kde průměrná přesnost nad prahem 0,5 dosahuje zhruba 0,730. Přísnější metrika mAP@0,5 však dosahuje zhruba přesnosti 0,435. Po vizuální inspekci inferenčních výsledků s nízkou hodnotou mAP se dá usoudit, že na vině jsou primárně snímky s neúplným modelem v záběru. Na lokálním systému jsme dosáhli následujících průměrných inferenčních rychlostí:

Předzpracování	Inference	Post-zpracování	Velikost vstupu
1,0 ms	11,8 ms	1,8 ms	$640 \times 384 \times 3$

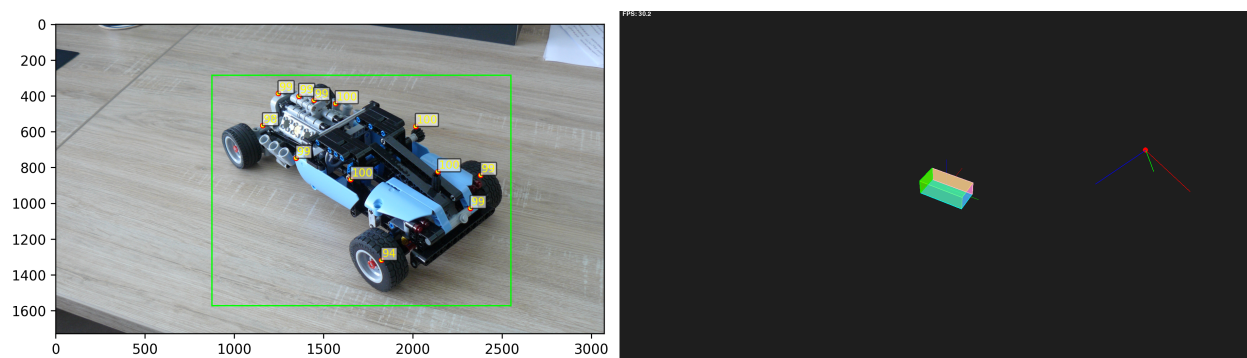
Tabulka 5.2: Rychlost inference modelu YOLOv11-s

5.2 Výkon na reálných datech

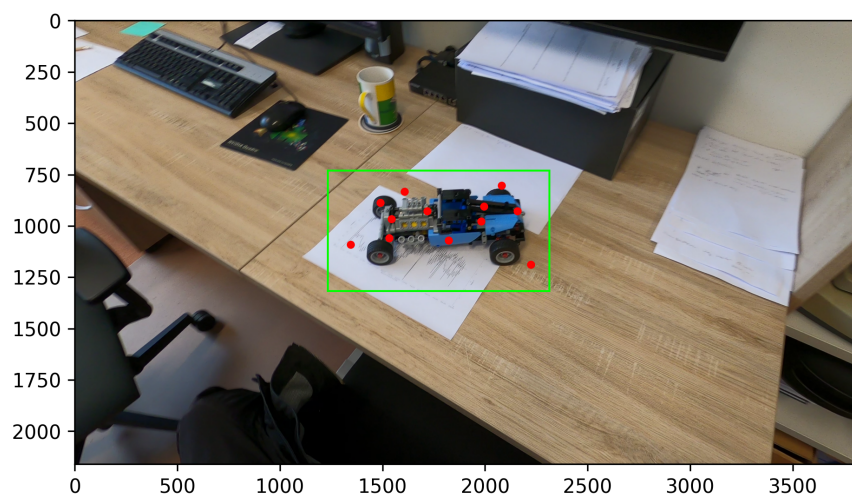
Model YOLOv11-s použitý pro lokalizaci objektu Hot Rod 42022 v kombinaci s metodou řešení perspektivního problému n bodů `cv2.SOLVEPNP_ITERATIVE` a robustní metodou RANSAC dosáhl uspokojivých výsledků. Při testování na videosekvencích se osvědčil jak samotný detekční model, tak i následný odhad pózy. Za předpokladu správné kalibrace kamery a odpovídajícího nastavení kamerové matice přístup poskytuje přesné a stabilní výsledky. Při testování volby `useExtrinsicGuess` ve spojení s předchozími hodnotami vektorů `tvec` a `rvec` během zpracování videosekvence nebyl pozorován žádný rozdíl ve výsledcích.

Za pozoruhodné omezení lze považovat sníženou přesnost lokalizace klíčových bodů v případech, kdy objekt není plně zachycen v záběru – a to i přesto, že podobné situace jsou v trénovacím datasetu

poměrně hojně zastoupeny. Naopak odhad ohraničujícího boxu je ve většině případů spolehlivý. U vzdálenějších záběrů modelu auta byl v krajních případech zaznamenán nepřesný odhad pózy, což lze pravděpodobně přičíst nedostatečnému zastoupení takových snímků v trénovací množině.



Obrázek 5.2: Odhad pózy modelu Hot Rod 42022



Obrázek 5.3: Nepřesný odhadu pózy modelu Hot Rod 42022

Kapitola 6

Závěr

V rámci této práce byl prozkoumán postup detekce modelu Hot Rod 42022, následná lokalizace jeho klíčových bodů pomocí sítě YOLOv11-*pose* a odhad pózy prostřednictvím algoritmů PnP za využití metody RANSAC. Výsledkem je implementace funkčního přístupu pro odhad pózy, fungujícího jak na jednotlivých snímcích, tak i ve videosekvencích, a to s úspěšnými výsledky ve většině případů. Funkčnost řešení byla ověřena v grafickém rozhraní, které slouží k vizualizaci relativní pozice kamery vůči objektu zájmu.

Použitý model YOLOv11-s dosáhl metriky mAP@0,5 OKS ve výši 0,73 a mAP@0,5:0,95 ve výši 0,435, přičemž hodnoty metrik naznačují fakt, že delším tréninkem lze tohoto výsledku dále zlepšit. V rámci navazující práce lze navrhnout rozšíření rozsahu škálování zájmového modelu při generování syntetického datasetu, čímž by mohlo dojít ke zvýšení škálové invariance.

Literatura

1. SLAVÍK, Daniel. *Lokalizace klíčových bodů pomocí neuronových sítí*. 2024. Dostupné také z: <http://hdl.handle.net/10084/153644>. Online. [cit. 2025-05-19].
2. KHANAM, Rahima; HUSSAIN, Muhammad. *YOLOv11: An Overview of the Key Architectural Enhancements*. 2024. Dostupné z arXiv: 2410.17725 [cs.CV].
3. *Perspective-n-Point (PnP) pose computation* [https://docs.opencv.org/4.x/d5/d1f/calib3d_solvePnP.html]. 2025. Online. [cit. 2025-05-19].
4. SHRESTHA, Rashik. *Perspective-n-Point (PnP)*. 2023-09. Dostupné také z: <https://medium.com/@rashik.shrestha/perspective-n-point-pnp-f2c7dd4ef1ed>. Online. [cit. 2025-05-19].
5. HAQ, Abdul. *SolvePnP PRANSAC and OPTIMIZATION*. 2024-09. Dostupné také z: <https://medium.com/@abdulhaq.ah/solvepnp-pransac-and-optimization-47a0683227b1>. Online. [cit. 2025-05-19].
6. ULTRALYTICS. *YOLOv8 vs YOLO11: A Technical Comparison*. 2024. Dostupné také z: <https://docs.ultralytics.com/compare/yolov8-vs-yolo11/>. Online. [cit. 2025-05-19].
7. ULTRALYTICS. *Configuration – Ultralytics YOLO Docs*. [B.r.]. Dostupné také z: <https://docs.ultralytics.com/usage/cfg/>. Online. [cit. 2025-05-19].
8. *Camera Calibration and 3D Reconstruction* [https://docs.opencv.org/4.x/d9/d0c/group_calib3d.html]. 2025. Online. [cit. 2025-05-19].
9. DUTTA, Tapas; DAWN, Kunal. *Object Keypoint Similarity in Keypoint Detection*. 2023-08. Dostupné také z: <https://learnopencv.com/object-keypoint-similarity/>. Online. [cit. 2025-05-19].