

NAO – Knihovník

Zpráva k semestrálnímu projektu

Daniil Myronov

Martin Škopek

Abstrakt

Cílem semestrálního projektu bylo vyvinout klient – server aplikaci pro humanoidního robota NAO, který bude sloužit jako asistent při vyhledávání správné krabice (poličky) pro danou knihu z dynamického katalogu knih.

Obsah

Řešení.....	3
Klient server architektura.....	4
Klient.....	5
Server.....	6

Řešení

Původní myšlenka těsně po výběru tématu semestrálního projektu byla autonomní robot procházející se prostorem, ke kterému by bylo možné přijít, předat mu knihu a robot by ji založil do správné krabice. To se vzhledem k nosnosti NAO robota ukázalo jako neproveditelné, spokojili jsme se tedy pouze s klasifikací knihy a lokací správné krabice.

Prvotní návrh klasifikátoru počítal se statickým indexem známých knih a jejich klasifikací pouze na základě fotografie, bez zpracování textu. Podobně jako např.
https://cs229.stanford.edu/proj2015/127_report.pdf.

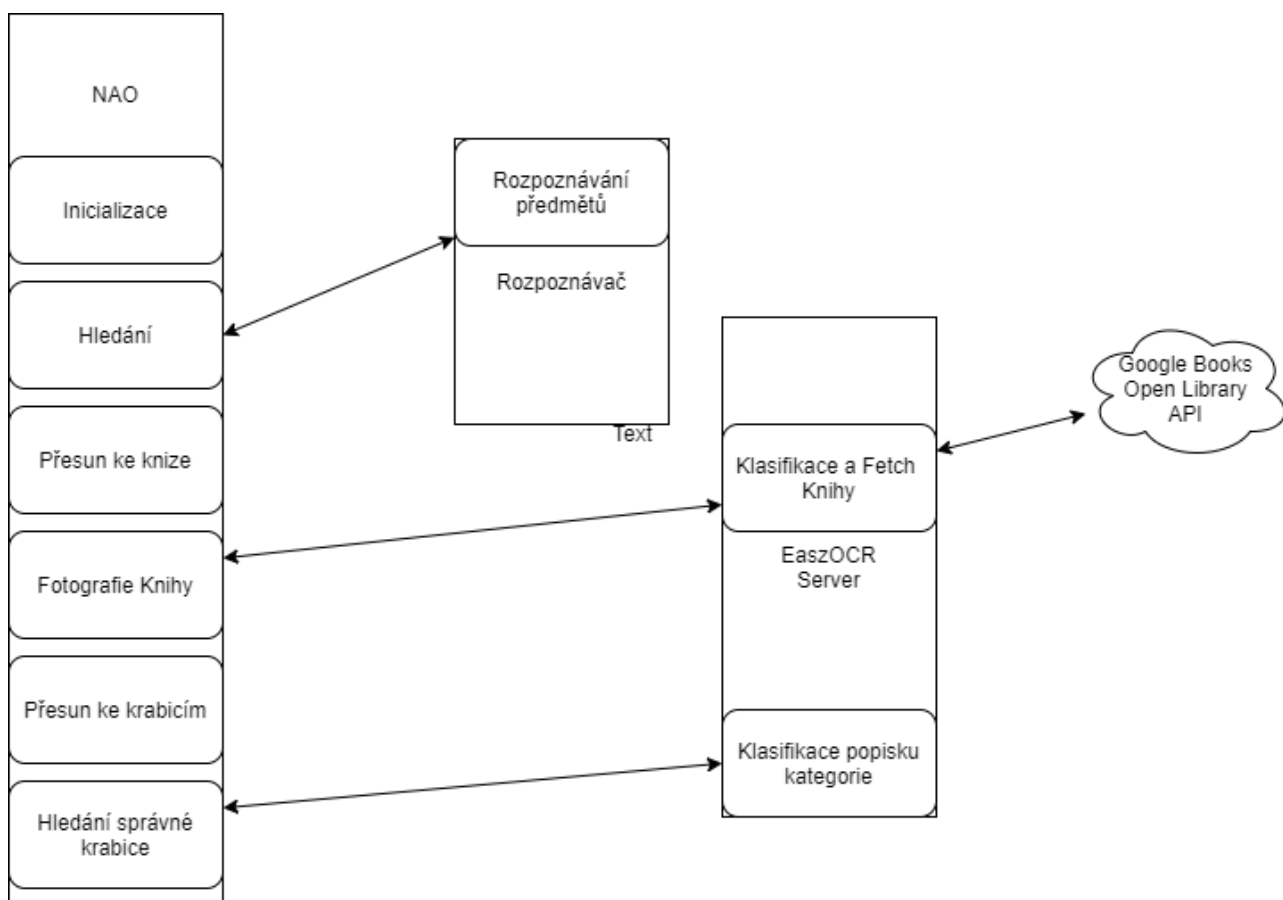
Tato varianta se ukázala jako slepá ulička, zejména kvůli potenciálním nárokům na výkon a kolísavou kvalitou kamer NAO robota.

Nakonec jsme se rozhodli využít textu na obálce a pomocí předtrénovaného klasifikátoru easyOCR (<https://github.com/JaidedAI/EasyOCR>) se pokusit získat název knihy. V této fázi jsme se rozhodli namísto statického indexu využít služeb veřejně dostupných API Google Books (books.google.com) to se osvědčilo jako dobrý zdroj ISBN, na druhou stranu má problém s poskytováním relevantních kategorií, a proto jsme se rozhodli získané ISBN využít k dotazu vůči API Open Library (<https://openlibrary.org/>) a tato data doplnit a poté už jen najít správnou krabici.

Klient server architektura

„Klient-server je síťová architektura, která odděluje klienta (často aplikaci s grafickým uživatelským rozhraním) a server, kteří spolu komunikují přes počítačovou síť. Klient–server aplikace obsahují jak klienta, tak i server. Alternativou architektury klient–server je peer-to-peer, kde spolu však komunikují počítače mezi sebou bez serveru. „
(Klient–server, <https://cs.wikipedia.org/w/index.php?title=Klient%E2%80%93server> (naposledy navštíveno 23. 01. 2023).)

Vzhledem k výpočetní náročnosti rozpoznávání textu jsme se rozhodli využít NAO robota pouze jako agenta prohledávajícího prostředí a samotné rozpoznávání knih a popisků kategorií na krabicích je delegováno na server, se kterým robot komunikuje pomocí REST API



Obrázek 1: Schema projektu (via app,diagrams.net)

Klient

Script spouštěný na klientovi má za cíl ho nechat pohybovat prostorem a v případě, že narazí na knihu, tak jí nasměrovat ke správné krabici to se dá rozdělit na několik fází:

1. Inicializace

Nejprve se inicializuje samotná třída robota, zapíše se IP serverů a připravíme si objektové abstrakce API robota

2. Čekáme na dotek hlavy

3. Hledání

Robot se postupně podívá před sebe 30, resp 60 stupňů doleva a pak doprava pokud knihu nenajde, posune se dopředu

Robot prochází prostorem s metodou look_for_book, průběžně fotí spodní kamerou a za pomoci metody find_book se průběžně dotazuje rozpoznávacího serveru, jestli našel knihu.

4. Kniha nalezena

Pomocí metody move_with_stops se iterativně přiblížíme ke knize, abychom měli lepší záběr kamery.

5. Vylepšíme úhel kamery

pomocí change_posture

6. Vyfotíme obálku

a provedeme transformaci

6. Obálku si necháme vyhodnotit serverem

7. Z odpovědi sestavíme info o knize

8. Zamíříme ke krabicím

Na základě historie pozic

9. Hledáme správnou krabici

Na základě popisku krabice vyhodnoceného serverem a posunem o konstantní počet kroků

10. Krabice nalezena

Nebo jsme alespoň prověřili všechny možnosti a žádná nevyhovovala

11. Zeptáme se, jestli pokračovat

Server

Server byl naprogramován v jazyce Python a pro obsluhu komunikace byl zvolen vestavěný `http.server`. Server má na starosti dva úkoly:

1. Rozpoznávání textu na krabicích

Při požadavku z url „category“ se obrázek pořízený robotem nahraje na sever, kde je následně předzpracován. Výběr metod pro předzpracování probíhal na základě nejlepšího zpracování testovacích dat bez fotografií z reálného prostředí (k nalezení v adresáři serveru) a následně korigován podle výsledků při práci s robotem v laboratoři. Při výběru metod zpracování mi pomohly články ^{1 2} a ³.

2. Rozpoznávání textu na obálce knihy

Při požadavku z url „cover“ se obrázek pořízený robotem nahraje na server a snažíme se zjistit jméno knihy. V tomto kroku se předzpracování obrázku ukázalo spíše kontraproduktivní a tak jsme se rozhodli pouze o otáčení obrázku a využití balíčku `nonsense` (<https://github.com/casics/nostril>) který jisté chyby odhalí, dále se omezujeme na alespoň pětiznaková slova a nejpravděpodobnější dvojici poté posíláme jako dotaz na Google Books API . V případě úspěchu se knihu podle ISBN pokusíme dovyhledat v Open Library a získané informace v JSON formátu vrátit ke zpracování klientovi.

1<https://towardsdatascience.com/pre-processing-in-ocr-fc231c6035a7>

2<https://nextgeninvent.com/blogs/7-steps-of-image-pre-processing-to-improve-ocr-using-python-2/>

3<https://pyimagesearch.com/2021/11/22/improving-ocr-results-with-basic-image-processing/>