Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky

Implementácia výpočtov na hrane (MEC) s využitím robotov Rosmaster R2

Diplomová práca

2025 **Bc. Igor Gombala**

Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky

Implementácia výpočtov na hrane (MEC) s využitím robotov Rosmaster R2

Diplomová práca

Študijný program:	Počítačové siete
Študijný odbor:	Informatika
Školiace pracovisko:	Katedra elektroniky a multimediálnych telekomunikácií (KEaMT)
Školiteľ:	doc. Ing. Gabriel Bugár, PhD.

Košice 2025 **Bc. Igor Gombala**

Abstrakt v SJ

Abstrakt v slovenčine.

Kľúčové slová v SJ

LaTeX,

Abstrakt v AJ

Abstract in English.

Kľúčové slová v AJ

LaTeX,

Bibliografická citácia

GOMBALA, Igor. *Implementácia výpočtov na hrane (MEC) s využitím robotov Rosmaster R2*. Košice: Technická univerzita v Košiciach, Fakulta elektrotechniky a informatiky, 2025. 11s. Vedúci práce: doc. Ing. Gabriel Bugár, PhD.

Tu vložte zadávací list pomocou príkazu
`\thesisspec{cesta/k/suboru/so/zadavacim.listom}`
v preambule dokumentu.
Kópiu zadávacieho listu skenujte čiernobielo (v odtieňoch sivej) na 200 až 300 DPI! Nezabudnite do jednej práce vložiť originál zadávacieho listu!

Čestné vyhlásenie

Vyhlasujem, že som záverečnú prácu vypracoval(a) samostatne s použitím uvedenej odbornej literatúry.

Poďakovanie

Na tomto mieste by som rád poďakoval svojmu vedúcemu práce za jeho čas a odborné vedenie počas riešenia mojej záverečnej práce. Rovnako by som sa rád poďakoval svojim rodičom a priateľom za ich podporu a povzbudzovanie počas celého môjho štúdia.

Obsah

Zoznam obrázkov

1 Úvod

Formulácia úlohy

1. Štúdium teoretických východísk a analýza výsledkov predchádzajúcich prác. Oboznámenie sa s konceptom Multi-access Edge Computing (MEC) a decentralizovaných výpočtových sietí. Detailne preskúma výsledky predchádzajúcich diplomových prác, ktoré sa venovali optimalizácii výpočtov na hrane a bezpečnostným aspektom v 5G a 6G sieťach.

2. Návrh a realizácia MEC siete v laboratóriu počítačových sietí. Cieľom je vytvoriť funkčné laboratórne prostredie pre MEC sieť, ktorá bude umožňovať efektívne využívanie hraničných výpočtov. Študent navrhne architektúru siete, konfiguráciu zariadení a integráciu výpočtových uzlov na hrane.

3. Implementácia výpočtov na hrane MEC s využitím robotov Rosmaster R2 (vozidlá autonómnej mobility - robot so štruktúrou Ackermann podporujúci operačný systém ROS, ROS2 pre použitie AI). Návrh a zrealizovanie riešenia umožňujúce robotom Rosmaster R2 využívať výpočtový výkon MEC siete. Riešenie bude optimalizované na zníženie latencie, minimalizáciu spotreby energie a efektívne rozdelenie výpočtovej záťaže medzi hraničné a cloudové uzly.

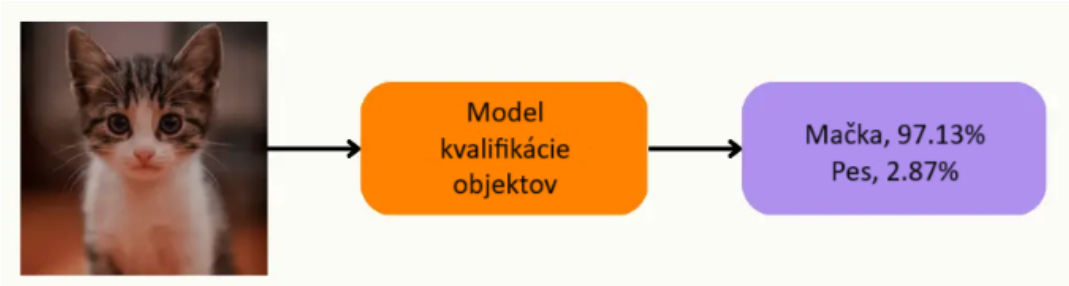
4. Testovanie a hodnotenie efektivity navrhnutého riešenia. Súčasťou budú merania výkonnosti siete a analýza parametrov, ako sú latencia, šírka pásma, výpočtový výkon a spotreba energie. Overenie stability systému a jeho schopnosť zvládnuť rôzne typy pracovných záťaží.

5. Vypracovanie dokumentácie a odporúčaní pre ďalší vývoj. Na záver bude zdokumentovaný celý proces návrhu, implementácie a testovania. Študent vypracuje odporúčania pre ďalší vývoj a optimalizáciu MEC sietí v spojení s autonómnymi robotmi.

2 Analytická časť

2.1 Klasifikácia objektov

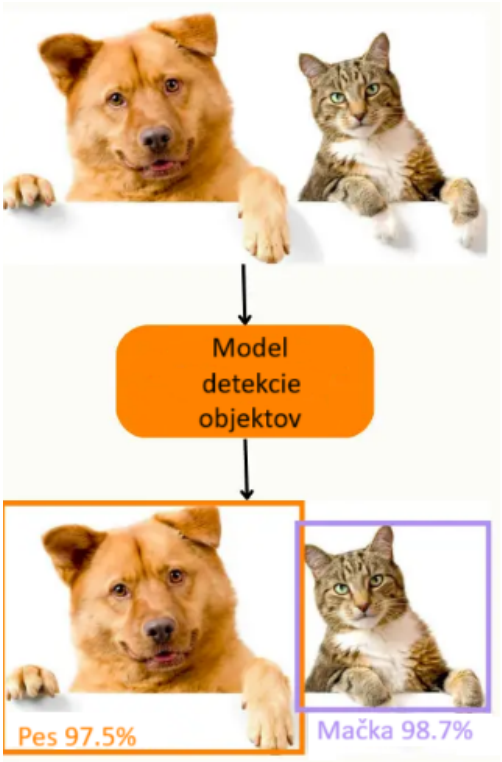
Pod názvom klasifikácia objektov môžeme chápať úlohu, ktorá identifikuje a kategorizuje objekty v obrázku s vopred definovanými triedami. Klasifikácia objektov je typicky vykonávaná technológiami strojového učenia, kde model je trenovaný na vybranej dátovej sade obrázkov a ich priradenie do označených tried. Trenovací model môže byť použitý na klasifikovanie nových obrázkov priradením označenia triedy na základe ich naučených vlasností. Použitie klasifikácie objektov je zahrnuté napr. pri rozpoznávaní dopravných značiek alebo identifikáciu rastliny na obrázku. [1]



Obrázok 2.1: Zreťazenie klasifikácie objektov

2.2 Detekcia objektov

Detekcia objektov je z pohľadu počítača úloha, ktorá identifikuje a lokalizuje objekty na základe preddefinovaných triedach vo vstupných obrázkov. S vývojom neurónových sietí, detekcia objektov dosiahla veľmi sľubné výsledky. Existuje pár modelov a algoritmov ako sú Region-Based Convolutional Neural Network (R-CCN), rýchlejší R-CCN, You Only Look Once (YOLO) a Single-Shot Detector (SSD), ktoré boli vyvinuté na detekciu objektov. Tieto algoritmy a modely sa používajp v rôznych aplikáciách, ako je samojazdiace autá, sledovacie systémy a sledovanie objektov. [1]



Obrázok 2.2: Zreťazenie detekcie objektov

2.3 You only look once (YOLO)

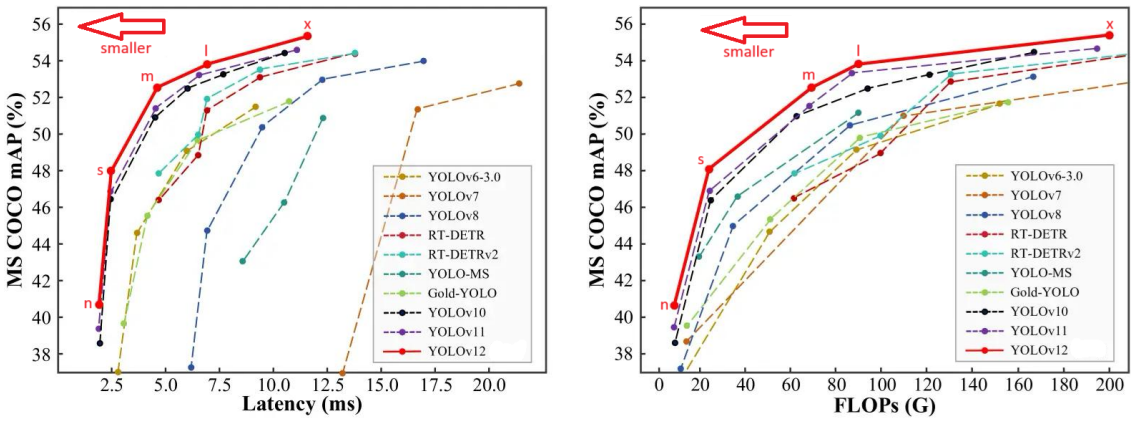
YOLO je algoritmus detekcie objektov v reálnom čase, vyvinutý v roku 2015 dvojicou autorov Joseph Redmon a Ali Farhadi. Ide o jednostupňový detektor objektov, ktorý využíva konvolučnú neurónovú sieť (CCN) na prepovedanie ohraničujúcich políčok a prevdepodobností tried objektov vo vstupných obrázkoch.

Algoritmus YOLO rozdeľuje vstupný obrázok na mriežku buniek a pre každú bunku predpovedá pravdepodobnosť prítomnosti objektu a súradnice ohraničujúceho rámčeka objektu a prepovedá tiež triedu objektu. Na rozdiel od dvojstupňovových detektorov objektov, ako je R-CCN a jeho variantov, YOLO spracováva celý obraz v jednom priechode, vďaka čomu je rýchlejší a efektívnejší.

YOLO bolo vyvinuté v niekoľkých verziách od YOLOv1 až po najnovšie YOLOv12. Každá verzia bola postavená na predchádzajúcej verzii s vylepšenými funkciami, ako je vylepšená presnosť, rýchlejšie spracovanie a lepšia manipulácia s malými predmetmi.

YOLO je široko používaný v rôznych aplikáciach, ako sú samoriadiace autá a sledovacie systémy. Je tiež široko používaný na úlohy detekcie objektov v reálnom čase, ako je analýza videa v reálnom čase a video dohľad v reálnom čase. [1]

Ľavá časť obrázku vykresľuje pomer strednej priemernej presnosti ku odozve, ktoré sú zoradené podľa veľkosti od nano (n) až po x (xlarge), kde druhá časť obrázku znázorňuje pomer strednej priemernej presnosti ku Floating point operations per second (FLOPs) a taktiež zoradené podľa veľkosti od nano (n) až po x (xlarge).



Obrázok 2.3: Porovnanie s ďalšími známymi metódami pokiaľ ide o kompromisy latencie-presnosť (vľavo) a FLOP-presnosť (vpravo)

Princíp YOLO algoritmu

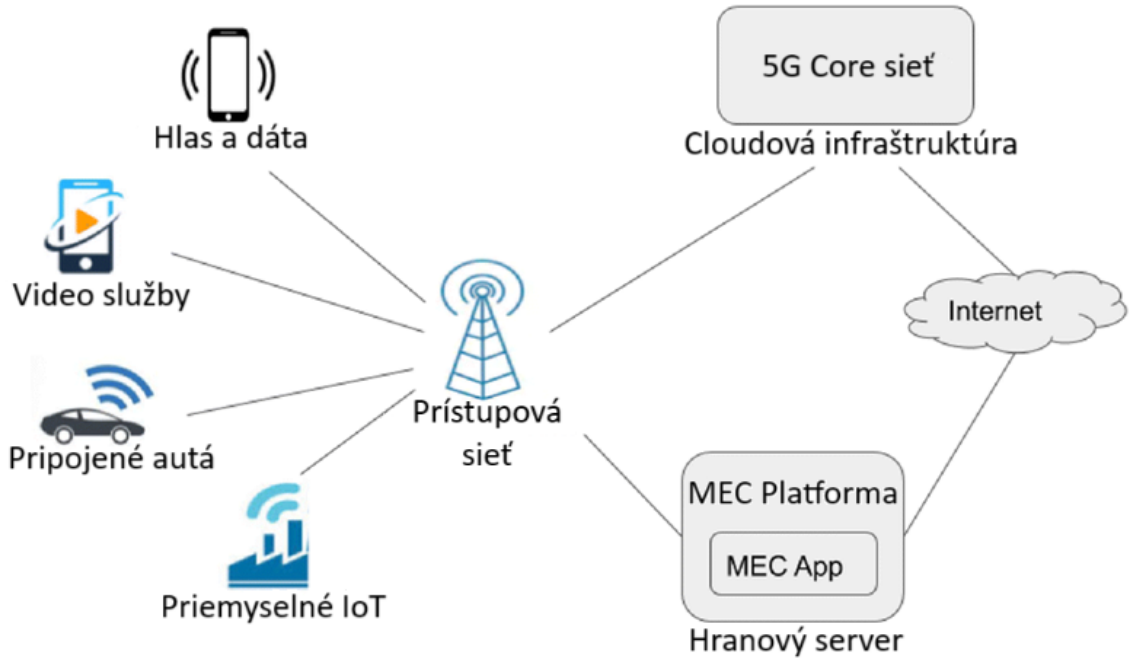
Základný princíp YOLO spočína v rozdelení vstupného obrázka na mriežku buniek a pre každú bunku prepovedá pravdepodobnosť prítomnosti objektu a súradnice ohraničujúceho rámčeka objektu. Proces YOLO sa môže rozdeliť do niekoľkých krokov:

1. Vstupný obrázok sa odovzdáva cez CCN, aby sa z obrázka extrahovali prvky.
2. Prvky sa potom prechádzajú sériou plne prepojených vrstiev, ktoré predpovedajú pravdepodobnosť tried a súradnice ohraničujúceho rámčeka.
3. Obrázok je rozdelený na mriežku buniek a každá bunka je zodpovedná za predpovedanie množiny ohraničujúcich políčok a pravdepodobností tried.
4. Výstupom siete je súbor ohraničujúcich rámčekov a pravdepodobností tried pre každú bunku.
5. Ohraničujúce rámčeky sa potom filtrujú pomocou algoritmu následného spracovania nazývaného non-max suppression, aby sa odstránili prekrývajúce sa rámčeky a aby sa vybral rámček s najvyššou prevdepodobnosťou.

6. Konečným výstupom je sada predpokladaných ohraničujúcich rámčekov a označaní tried pre každý objekt na obrázku. [1]

2.4 Multi-access edge computing (MEC)

Multi-access edge computing alebo MEC architektúra, ktorá umožňuje poskytovateľom výpočtových, sieťových a mobilných služieb presunúť niektoré výpočtové a cloudové procesy na okraj sieťového prostredia, čím sa dosiahne lepší výkon, latencia a bezpečnosť. Multi-access edge computing (MEC) je myšlienka alebo koncept vyvinutý Európskym inštitútom pre telekomunikačné normy (ETSI). Na rozdiel od dátového centra poskytuje táto sieťová architektúra služby požadované používateľmi a operáciae cloud computingu na okraji siete, ktorý je bližšie ku koncovým používateľom. [2]



Obrázok 2.4: Multi-access edge computing (MEC) architektúra

Význam MEC možno rozdeliť takto:

- Viacnásobný prístup (Multi-Access) - Viacnásobný prístup umožňuje systému MEC poskytovať bezproblémový zážitok zákazníkom pristupujúcim k sieti prostredníctvom technológie podľa vlastného výberu.
- Hrana (EDGE) - Premiestnenie sieťových operácií a aplikácií na okraj siete umožňuje mimoriadne nízku latenciu.

- Výpočtová technika (Computing) - Výpočtové možnosti siete sú rozložené smerom k hrane siete.

Medzi kľúčové vlastnosti MEC patria:

- Tesná blízkosť ku koncovému užívateľovi
- Nízka latencia dátového prenosu
- Vhodnosť pre aplikácie v reálnom čase
- Neprerušované prevádzkové schopnosti
- Vzájomná komunikácia s existujúcimi aplikáciami
- Väčšia miera virtualizácie [2]

2.5 ROSMASTER R2

ROSMASTER R2 je mobilné auto s Ackerman riadiacou štruktúrou vyvynutou pre ROS2 systémy. R2 je vybavené s vysoko kvalitnými súčiastkami ako je laserový radar, hĺbková kamera, hlasová interakcia, gumová pneumatika 520, ktorá dokáže realizovať navigáciu mapovania robota, vyhýbanie sa prekážkam, automatickú jazdu, rozpoznávanie ľudských funkcií, ovládanie hlasovej interakcie a ďalšie funkcie. [3]

ROS je operačný systém s otvoreným zdrojovým kódom vhodný pre roboty. Poskytuje služby, ktoré by mal mať operačný systém, vrátane abstrakcie hardvéru, nízkoúrovňového ovládania zariadení, implementácie bežných funkcií, odovzdávania správ medzi procesmi a správy balíkov. Poskytuje tiež nástroje a funkcie knižnice potrebné na získanie, kompliláciu, zápis a spustenie kódu v počítačoch.[3]

Obrázok 2.5: ROSMASTER R2

Hlavné vlastnosti ROS

- Distribuivaná architektúra (každý pracovný proces je považovaný za uzol a je riadený jednotne pomocou správcu uzlov),
- Podpora viacerých jazykov (napr. C++, Python, atď),
- Dobrá škáliteľnosť (môžete napísať jeden uzol, alebo usporiadať veľa uzlov do väčšieho projektu cez roslaunch),
- Otvorený zdrojový kód (ROS sa riadi protokolom BSD a je úplne zadarmo pre individuálne a komerčné aplikácie a úpravy). [3]

2.6 WebSockets

Aplikácie v reálnom čase sa za posledné desaťročie stali oveľa bežnejšími a umožňujú okamžitú komunikáciu a živé aktualizácie, ktoré udržujú používateľov v kontakte a informovanosti. Jednou z kľúčových technológií umožňujúcich tieto možnosti je protokol WebSockets. [4]

WebSockets je komunikačný protokol určený pre plne duplexné komunikačné kanály cez jedno TCP pripojenie. Po nastavení pripojenia WebSockets môžu klient aj server odosielať a prijímať údaje súčasne bez potreby opakovaného otvárania a zatvárania pripojení. Toto trvalé pripojenie znižuje latenciu oproti tradičným požiadavkám HTTP, vďaka čomu je protokol WebSockets ideálny pre aplikácie v reálnom čase. [4]

Výhody WebSockets

Protokol WebSockets ponúka niekoľko silných výhod v porovaní s tradičným protokolom HTTP na požiadavku a odpoveď, najmä ak záleží na rýchlosti odozvy:

- **Nízka latencia**: Keďže pripojenie ostáva otvorené, psrávy sa môžu posielať oboma smermi bez oneskorenia aj pri ďalších požiadavkách,
- **Znížená réžia**: Protokol HTTP zabaluje každú správu do hlavičky a ďalších informácií, kde protokol WebSockets väčšinu z toho odstaňuje a znižuje údaje na odosielanie a spracovanie,
- **Obojsmerná komunikácia**: Spojenie servera s klientom môže medzi sebou kedykoľvek komunikovať bez potreby požiadaviek,
- **Škálovateľnosť**: Jeden WebSockets server dokáže sledovať mnoho aktívnych pripojení naraz, vďaka čomu je vhodný pre aplikácie s veľkým počtom používateľov. [4]

3 Vyhodnotenie

4 Záver

Zoznam použitej literatúry

1. MIRKHAN, Asmaa. *YOLO Algorithm: Real-Time Object Detection from A to Z*. https://kili-technology.com/data-labeling/machine-learning/yolo-algorithm--real-time-object-detection-from-a-to-z, 2023. Tech. spr. Kili.
2. BASUMALLICK, Chiradeep. *What Is Multi-Access Edge Computing (MEC)? Features, Examples, and Best Practices*. https://www.spiceworks.com/tech/edge--computing/articles/multi-access-edge-computing/, 2023-02. Tech. spr. Spiceworks.
3. YAHBOOM. *Welcome to ROSMASTER R2 repository - ROS introduction*. `http://www.yahboom.net/study/ROSMASTER-R2`, 2023. Tech. spr. Yahboom.
4. OBREGON, Alexander. *Building Real-time Applications with Python and WebSockets*. https://medium.com/@AlexanderObregon/building-real-time-applications-with-python-and-websockets-eb33a4098e02, 2024-06. Tech. spr. Medium.

Zoznam skratiek

BSD Berkeley Software Distribution.

CCN Convolutional Neural Network.

FLOPs Floating point operations per second.

MEC Multi-access edge computing.

R-CCN Region-Based Convolutional Neural Network.

ROS Robot Operating System.

SSD Single-Shot Detector.

YOLO You Only Look Once.