



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A BIOMECHANIKY

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

NÁVRH ŘÍDICÍHO SYSTÉMU ROBOTU K ROZMÍSTĚNÍ VÝSTRAŽNÉHO ZNAČENÍ PŘI DOPRAVNÍ NEHODĚ

DESIGN OF THE ROBOT'S CONTROL SYSTEM FOR THE DISTRIBUTION OF WARNING SIGNS IN A
TRAFFIC ACCIDENT

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Kristián Karas

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Michal Bastl

BRNO 2019

Zadání bakalářské práce

Ústav:	Ústav mechaniky těles, mechatroniky a biomechaniky
Student:	Kristián Karas
Studijní program:	Aplikované vědy v inženýrství
Studijní obor:	Mechatronika
Vedoucí práce:	Ing. Michal Bastl
Akademický rok:	2018/19

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Návrh řídicího systému robotu k rozmístění výstražného značení při dopravní nehodě

Stručná charakteristika problematiky úkolu:

Cílem práce je návrh robotu k automatickému rozmístění výstražného systému během autonehody. Robotický diferenciální podvozek je hotov. Cílem práce je podvozek převzít a vytvořit řídicí software řešení, které automaticky dojede do určené vzdálenosti podél vozovky. Během postupu práce se předpokládá testování různé sensoriky například videokamery, senzoru barvy v podvozku a ultrazvukových dálkoměrů.

Cíle bakalářské práce:

- 1) V rešeršní části stručně popište cíle práce a výsledky, kterých chcete dosáhnout.
- 2) Využijte dostupný podvozek v mechatronické laboratoři a naprogramujte Arduino Mega k obsluze sensoriky a ovládání pohonu. Naprogramujte komunikaci s nadřazeným systémem.
- 3) Vytvořte program na Raspberry Pi k ovládání robotického podvozku pomocí komunikace s arduinem. Naprogramujte potřebné zpracování vizuálních dat z kamery, případně z dalších senzorů.
- 4) Ověřte výsledky řešení na funkční demonstrační úloze automatického rozmístění výstražného systému v určené vzdálenosti.

Seznam doporučené literatury:

HOWSE, J.: OpenCV computer vision with Python: learn to capture videos, manipulate images, and track objects with Python using the OpenCV library. Birmingham, England: Packt Publishing, c2013.

CORKE, Peter I. Robotics, vision and control: fundamental algorithms in MATLAB. Berlin: Springer, 2011. Springer tracts in advanced robotics, v. 73. ISBN 9783642201448.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2018/19

V Brně, dne

L. S.

prof. Ing. Jindřich Petruška, CSc.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

Abstrakt

Hlavní téma této bakalářské práce je návrh modelu řízení autonomního robota pro rozmístění výstražných prvků při dopravní nehodě. Práce se zabývá dohromady třemi modely řízení. Dále jsou v práci představeny komponenty robota a komunikace dvou řídicích systémů.

Klíčová slova

Robot, řídicí systém, sériová komunikace, neuronová síť

Abstract

The main topic of this bachelor thesis is the design of the autonomous robot control model for the positioning of warning elements in a traffic accident. The thesis deals with three management models. Also the work introduces components of robot and communication of two control systems.

Keywords

Robot, control system, serial communication, neural network

Bibliografická citace:

KARAS, Kristián. Návrh řídicího systému robotu k rozmístění výstražného značení při dopravní nehodě. Brno, 2019. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/117195>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav mechaniky těles, mechatroniky a biomechaniky. Vedoucí práce Michal Bastl.

Prohlášení

„Prohlašuji, že svou bakalářskou práci na téma *Návrh řídicího systému robotu k rozmístění výstražného značení při dopravní nehodě* jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **20. května 2019**

.....
Kristián Karas

Poděkování

Chtěl bych na tomto místě poděkovat vedoucímu mé bakalářské práce za pomoc při zpracování tohoto projektu, protože pokaždé když jsem potřeboval konzultaci tak si na mě udělal čas a dokázal mi dobře poradit. Taky bych chtěl ještě poděkovat mému spolubydlicímu Ondřeji Neumanovi za rady při práci na projektu a nakonec i celé moji rodině, která mě podporovala společně s mojí přítelkyní.

V Brně dne: **20. května 2019**

.....
Kristián Karas

OBSAH

1	Úvod	10
2	Platforma, Komponenty	12
2.1	Robustní kolový podvozek	12
2.2	Řídící elektronika, motory a ostatní komponenty.....	12
3	Sériová komunikace	16
4	Návrh Algoritmu	18
4.1	Algoritmus pracující infračervenými senzory	20
4.2	Algoritmus pracující s obrazem z kamery	21
4.3	Algoritmus s neuronovou sítí.....	29
5	Demonstrační úloha a porovnání Algoritmů	36
6	Další vývoj a možné uplatnění	38
7	Závěr.....	40

Seznam obrázků

Obr. 2-1 Doručovací robot Starship[29].....	11
Obr. 2-1 Robotický podvozek	12
Obr. 2-2 Arduino MEGA 2560 [2].....	13
Obr. 2-3 Raspberry PI 3 [3].....	13
Obr. 3-5 Kamera a pouzdro kamery s pomocnou nakloněnou rovinou [4].....	14
Obr. 2-6 DC motor Pololu 2827 [5]	14
Obr. 2-7 Li-pol akumulátor Tattu [7]	14
Obr. 2-8 Ultrazvukový senzor [8]	15
Obr. 2-9 Modul pro Arduino – infračervený optický snímač [27]	15
Obr. 2-10 Komponenty robota	15
Obr. 3-1 Rozdělení digitální komunikace [14].....	16
Obr. 3-2 Základní druhy sériové komunikace [16]	17
Obr. 4-1 Pulzně šířková modulace [23].....	18
Obr. 4-2 Schéma systému s IR senzory.....	20
Obr. 4-3 Infračervené senzory pro detekci krajní čáry silnice	20
Obr. 4-4 Schéma systému.....	21
Obr. 4-5 Ukázka prahování	22
Obr. 4-7 Úprava binární fotografie pomocí funkce edge().....	23
Obr. 4-8 Vizualizace vytvořených bodů.....	23
Obr. 4-9 Přímka vytvořená metodou nejstrmějšího gradientu	25
Obr. 4-10 Přímka vytvořená pomocí metody průměrné přímky	26
Obr. 4-11 Schéma regulace s PID	28
Obr. 4-12 Zjednodušené schéma řídicího algoritmu v Arduinu.....	28
Obr. 4-13 Schéma systému s Neuronovou sítí	29
Obr. 4-14 Schéma programovacích přístupů.....	30
Obr. 4-15 Mapa s označením místa, sběru dat pro NS.....	30
Obr. 4-16 Vzorky č.1 – Robot jede přímo.....	31
Obr. 4-17 Vzorky č.2 – Robot otočený více na levou stranu	31
Obr. 4-18 Vzorky č.3 - Robot otočený více na pravou stranu.....	31
Obr. 4-19 Rozdělení dat pro proces učení NS	32
Obr. 4-20 Ukázka upravené fotografie.....	33
Obr. 4-21 ELU aktivační funkce [22].....	33
Obr. 4-22 Model NS	34
Obr. 4-23 Chyba predikce	35
Obr. 5-1 Místo demonstrační úlohy.....	36
Obr. 6-1 Ilustrativní obrázek - Aplikace robota na silnici [26]	38
Obr. 6-2 Ilustrativní obrázek - koncept využití robota [25]	39

1 ÚVOD

Tato práce se zabývá představením návrhu modelu řízení, robotického podvozku, který by mohl sloužit v případě dopravní nehody jako rychlý pomocník pro označení místa nehody výstražnými prvky. Mnohdy v takových situacích je potřeba reagovat co nejrychleji, aby se minimalizovalo riziko vzniku následné srážky dalších vozidel a předešlo se zranění.

Myšlenka je tedy taková, že v případě silniční nehody, kdy řidič je povinen označit místo dopravní nehody výstražnými prvky (trojúhelník či jiné světelné a reflexní prvky), aby upozornil na tuto nehodu okolní účastníky dopravního provozu, by řidič mohl využít robotickou platformu, která je schopná odjet od dopravní nehody po krajnici silnice do předem nastavené vzdálenosti, kde díky výstražným prvkům, kterými by pojízdná platforma byla vybavena, signalizovala vznik této dopravní nehody.

Situaci při silničních nehodách v některých případech komplikuje ještě i špatné počasí a snížená viditelnost, což zvyšuje riziko následné srážky vozidel a je proto nutné označit místo nehody co nejrychleji. A v případech nehod, kdy řidič jede sám v autě, je to ještě komplikovanější, protože daný člověk je na všechno sám a musí v krátké době stihnout označení silniční nehody, pomoc zraněným a v případě závažných zranění také zavolat záchrannou zdravotní službu.

CÍLE PRÁCE

V úvodu byla představena základní myšlenka, která byla inspirací tohoto projektu, avšak nutno podotknout, že tento systém, aby správně fungoval v běžném provozu je komplexní. Proto se tato práce zabývá pouze představením potřebného hardwarového zařízení pojízdného modelu, principy komunikace dvou řídicích zařízení a návrhu základních algoritmu pro sledování bílé krajní čáry silnice.

Tyto algoritmy budou poté testovány na demonstrační úloze na vhodném úseku silnice.

Dohromady budou představeny 3 řídicí algoritmy. Všechny 3 tyto algoritmy zpracovávají data ze senzorů, kterými je robotický podvozek vybavený. Tyto senzory budou představeny společně s ostatními komponenty ve 3. kapitole.

Závěrem budou shrnuté získané zkušenosti při práci na tomto projektu a zhodnocena realizovatelnost tohoto systému v praxi.

Shrnutí cílů:

- Představení potřebných komponentů
- Komunikace systému Arduino Uno a Raspberry Pi
- Popis algoritmů pro řízení podvozku
- Testování algoritmu na demonstrační úloze

INSPIRATIVNÍ PŘÍKLAD AUTONOMNÍHO ROBOTA

Tento relativně malý robot od firmy Starship slouží jako městský doručovatel malých balíčků. Robota lze potkat už i v Evropě a to v Estonsku, Německu a Nizozemí. Tento robot pracuje především v restauracích, ve kterých si přes mobilní aplikaci můžete objednat jídlo a při doručení musíte potvrdit přes aplikaci, že už robot přijel k vám, aby se mohl otevřít zamykací systém robota. V případě pokusu o krádež zboží jsou dvířka robota zamknuta a při manipulaci s celým robotem se spustí varovný alarm.

Robot vybaven kamerami, GPS, IMU jednotkou (ele. zařízení zahrnující akcelerometr, gyroskop, magnetometr), mikrofonom a reproduktorem pro komunikaci s lidmi. Pro zachování nízké ceny robota zde však není LIDAR (Light Detection And Ranging), který se používá u autonomních vozidel pro vytváření 3D mapy okolí.

Při řízení využívá robot i algoritmus s neuronovou sítí, kterou robot používá pro rozpoznávání objektů (Object detection) a pro orientaci v prostředí. Je tak schopný zvládnout přejet silnici, vyhnout se překážkám a předvídat pohyby kolemjdoucích [30].



Obr. 1-1 Doručovací robot Starship[29]

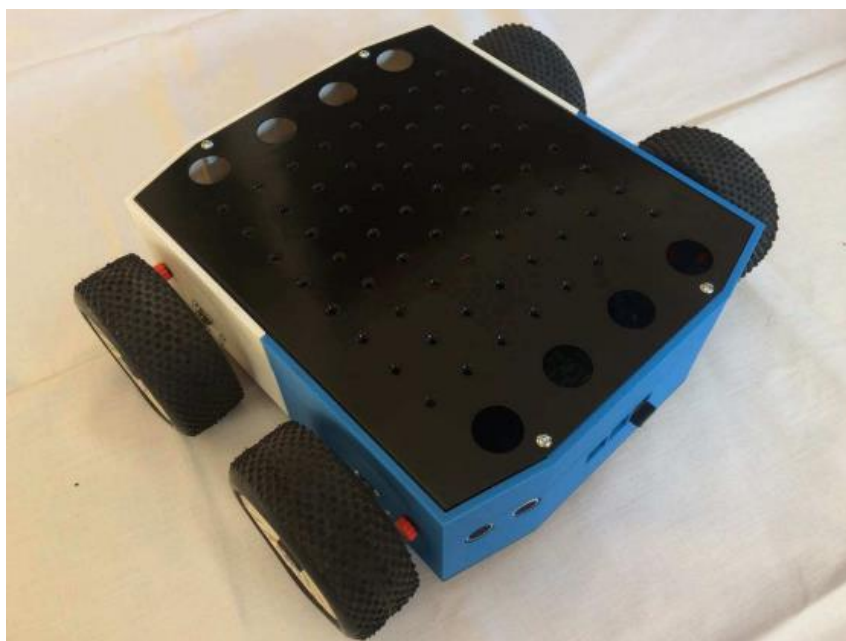
2 PLATFORMA, KOMPONENTY

2.1 Robustní kolový podvozek

Podvozek, který byl k dispozici k této práci, byl převzatý jako hotový funkční model sestavený při bakalářské práci studenta **Bc. Radka Jirgese** v ak. roce **2017/2018** [1].

POPIS PODVOZKU

Tento podvozek byl navržen, hlavně pro indoor aplikace, avšak zvládá i jízdu po mírně nerovném povrchu jako je silnice nebo chodník. Robot je sestaven kombinací plastů a kovu. Důraz je zde kladen především na pevnost a praktičnost tohoto podvozku.



Obr. 2-1 Robotický podvozek

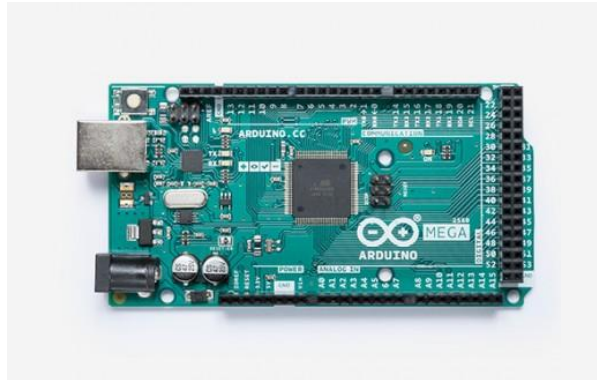
Vrchní a spodní díly jsou vyrobeny z milimetrového plechu, které jsou spojeny plastovým jádrem vyrobeným na 3D tiskárně. Pro hardware komponenty a motory umístěné uvnitř podvozku, byly zkonstruovány a vytisknuty upevňovací prvky, aby vše drželo na svém místě tak jak má.

2.2 Řídící elektronika, motory a ostatní komponenty

K tomuto projektu byly použity dva řídicí systémy, deska Arduino MEGA 2560, která má za úkol obsluhu senzorů a motorů a deska Raspberry PI, která má za úkol zpracování obrazu z Raspberry Pi Kamery. Tyto dvě desky také spolu komunikují a tvoří tak společně řídicí jednotku robota.

ARDUINO

K tomuto projektu byla vybrána deska Arduino MEGA 2560, postavena na známém mikrokontroléru ATmega 2560. Ten je vybavený několika piny, jak digitálními, tak i analogovými, které slouží v tomto projektu pro řízení motorů pomocí PWM signálů [2].



Obr. 2-2 Arduino MEGA 2560 [2]

RASPBERRY PI

Jako hlavní řídicí systém byla zvolena oblíbená deska Raspberry PI 3 Model B 64-bit 1GB RAM. Raspberry PI je známou firmou na poli jednodeskových počítačů, které lze využít pro různé aplikace. V našem projektu byla deska zvolena především díky přehlednému operačnímu systému Raspbian a také díky kamerovému modulu, který je snadno ovládatelný [3].

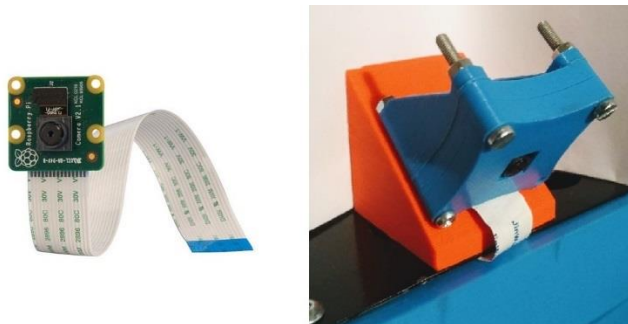


Obr. 2-3 Raspberry PI 3 [3]

RASPBERRY PI KAMERA

Kamerový modul Raspberry PI s rozlišením 5MP, který se připojuje k desce pomocí plochého kabelu se speciálním CSI rozhraním, byl zvolen pro tuto práci, pro snadnou implementaci a jednoduché ovládaní s Raspberry PI. Jednou z nevýhod u této kamery však může být krátký datový kabel a také potřeba dokoupení či vytisknutí pouzdra pro kameru. V tomto projektu se pro uchycení kamery vytisknulo pouzdro na 3D tiskárně, které je umístěné na horní části platformy na pomocné nakloněné rovině ve směru jízdy. Model pouzdra byl převzatý z webové stránky [15], kde je volně k dispozici. Pomocná

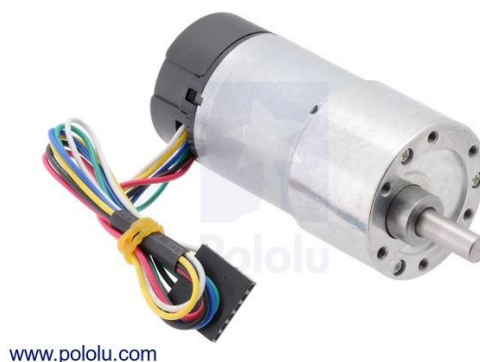
plošina byla navrhnutá v systému Inventor a rovněž vytisknuta na 3D tiskárně. Slouží pro přesnější zachycení krajní čáry silnice na fotografiích, které se poté zpracovávají v řídicím systému, který má za úkol vyhodnotit polohu této krajní čáry silnice na fotografiích [4].



Obr. 2-4 Kamera a pouzdro kamery s pomocnou nakloněnou rovinou [4]

MOTORY

Jako pohon podvozku jsou stejnosměrné motory POLOLU-2827, tyto motory mají převodovku s poměrem 131:1 a pro snímání rotace je zde na zadní straně hřídele magnetický kotouč s hallovým senzorem, který je využitý pro měření ujeté vzdálenosti robota [5].



Obr. 2-5 DC motor Pololu 2827 [5]

AKUMULÁTOR

Zdrojem energie je lithium-iontový tříčlánkový akumulátor s nominálním napětím 11,1V a kapacitou 2300mAh od firmy Tattu [7].



Obr. 2-6 Li-pol akumulátor Tattu [7]

ULTRAZVUKOVÉ SENZORY

Pro detekci případné překážky jsou použity čtyři ultrazvukové senzory, které měří vzdálenost v rozmezí od 2 do 400 cm. Tyto senzory jsou umístěny v přední i zadní části robota [8].



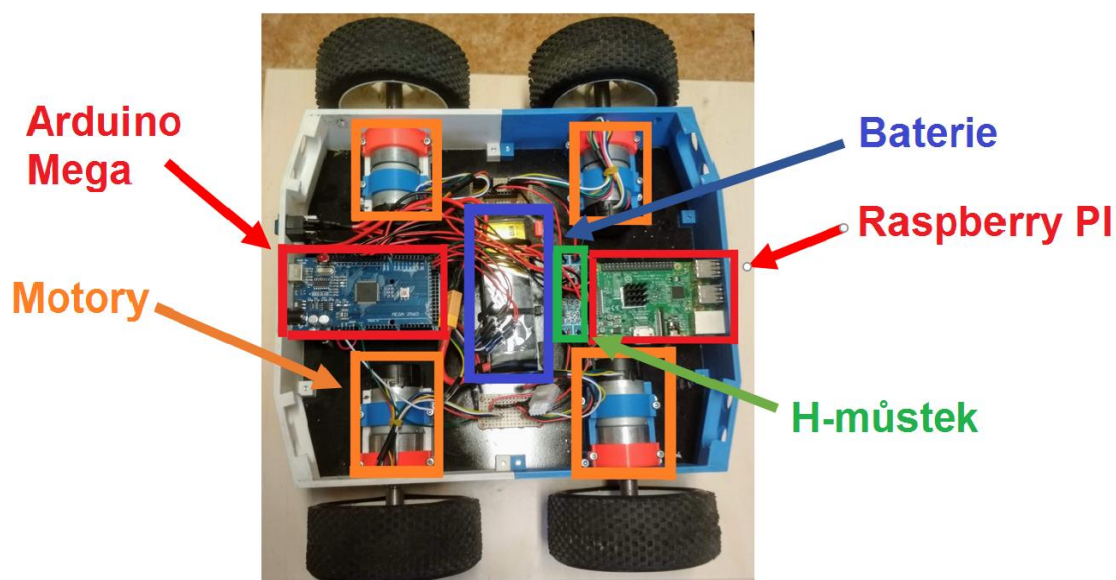
Obr. 2-7 Ultrazvukový senzor [8]

INFRAČERVENÉ OPTICKÉ SNÍMAČE

Pro detekci krajní čáry silnice v prvním algoritmu jsou použity čtyři infračervené optické senzory, které detekují impuls na černém povrchu. Tyto senzory pracují na napětí 3.0 až 5.0 V a jsou umístěny v přední části podvozku robota [27].



Obr. 2-8 Modul pro Arduino – infračervený optický snímač [27]



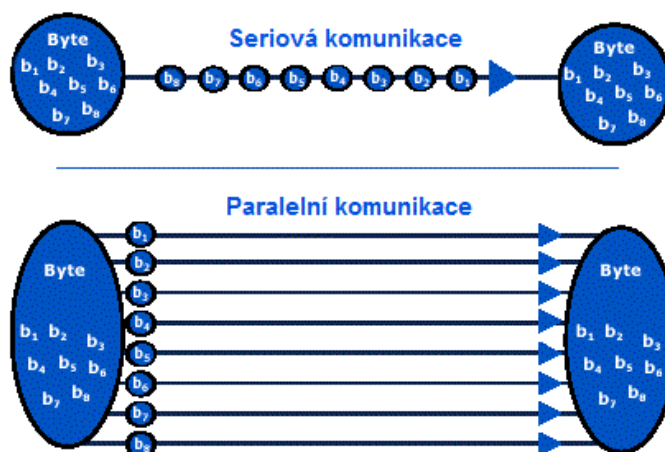
Obr. 2-9 Komponenty robota

3 SÉRIOVÁ KOMUNIKACE

V elektronických zařízeních se rozděluje digitální komunikace na dva základní typy:

- Sériová komunikace
- Paralelní komunikace

Hlavní rozdíl uvedených komunikací je jejich princip činnosti a uspořádání, jakým přijímají či odesílají data.



Obr. 3-1 Rozdělení digitální komunikace [14]

Pro sériové komunikaci dvou zařízení stačí tedy pouze jeden komunikační kanál, avšak v praxi se při této komunikaci setkáme i s více. Tyto další kanály jsou realizované jako pomocné a mají různou funkci, kde záleží na typu sběrnice, který se při komunikaci používá. Některé příklady sběrnice budou představeny dále. U sériové komunikace je hlavní charakter činnosti odesílání nebo přijímání jednoho bit za druhým.

U paralelní komunikace dochází k přenosu informace, jak už název napovídá, souběžně po několika komunikačních kanálech. Paralelní komunikace se využívá především na odesílání/přijímání celého bajtu najednou a podle toho se taky musí volit počet paralelních kanálů, podle bitového systému, s kterým dané zařízení pracuje (např. 8-bitové, 32-bitové systémy) [14].

SBĚRNICE PRO SÉRIOVOU KOMUNIKACI

I2C sběrnice

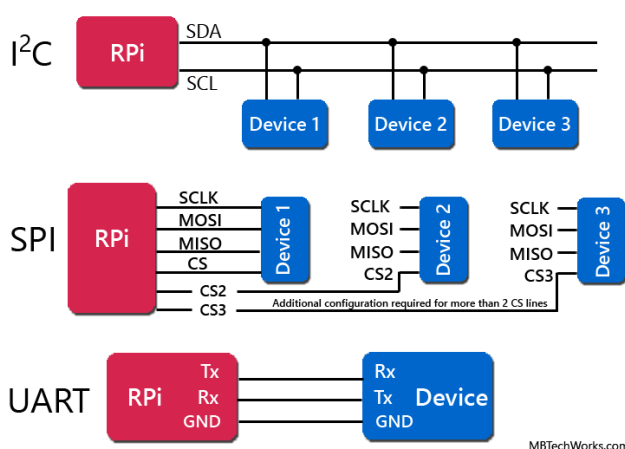
Tato sběrnice byla vytvořena společností Philips Semiconductor. Umožňuje připojení více než jednoho SLAVE zařízení s jedním nebo i více MASTER zařízeními. Pro svou komunikaci využívá minimálně dvou komunikačních kanálů mezi všemi členy komunikace [16].

SPI sběrnice

SPI je sběrnice od firmy Motorola, která se využívá ke komunikaci pouze jednoho MASTER zařízení s více SLAVE zařízeními. Tato sběrnice probíhá pomocí čtyř kanálů a na rozdíl od I2C může SPI pracovat i v duplexním módu (zároveň přijímat i vysílat data), kde rychlost komunikace dosahuje až 1Mbps [16].

UART

Univerzální asynchronní přijímač/vysílač neboli UART je sběrnice, která převádí paralelní data například z CPU na sériovou formu a odesílá ji přímo druhému zařízení (při této komunikaci spolu komunikují pouze 2 zařízení). Tato sběrnice byla použita i v projektu [16].



Obr. 3-2 Základní druhy sériové komunikace [16]

KOMUNIKACE POUŽITÁ V PROJEKTU

V práci, jak už bylo zmíněno výše, byla použita komunikace pomocí zařízení UART. Komunikace probíhá prostřednictvím USB portu mezi Raspbery PI a Arduinem Mega, na kterém se nachází USB-to-serial převodník.

4 NÁVRH ALGORITMU

Při návrhu algoritmu vznikly v konečném součtu tři nezávislé algoritmy pro sledování bílé krajní čáry silnice. Všechny tyto algoritmy budou dále představeny jeden po druhém.

PROGRMOVACÍ JAZYKY PYTHON A C++

Ještě před samotným představením všech algoritmů je třeba zmínit, že všechny kódy psané v nadřazeném systému Raspberry PI jsou psané v Pythonu, což je vysokoúrovňový skriptovací programovací jazyk.

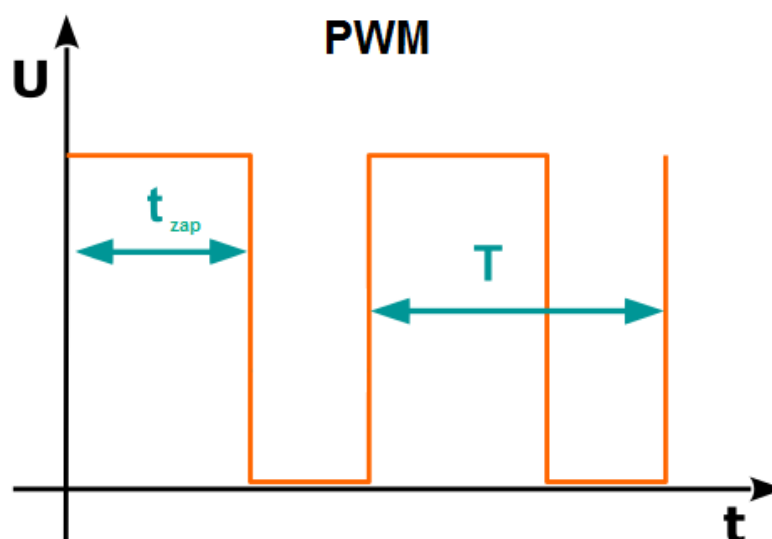
A kódy, které běží na Arduino byly psány prostředím Arduino IDE a jsou psány v programovacím jazyce C++, který je rozšíření základní verze jazyka C.

PULZNĚ ŠÍŘKOVÁ MODULACE

Řídicí signál do motorů robota je vytvářen pulzně šířkovou modulací, neboli PWM. Tato diskretní modulace se využívá pro přenos analogového signálu pomocí dvou diskretních hodnot (log 0 a log 1). Analogový signál je v PWM signálu obsažen jako poměr mezi log 0 a log 1, tzv. střída.

$$s = \frac{t_{zap}}{T} \quad (0)$$

kde: s střída
 t_{zap}doba log 1 (doba zapnutí)
 Tperioda



Obr. 4-1 Pulzně šířková modulace [23]

OPENCV KNIHOVNA

Při zpracování obrazu se pracuje s volně dostupnou, multiplatformní knihovnou OpenCV. Dříve než se stala tato knihovna Open-source knihovnou byla vyvíjena pod firmou Intel. Tato knihovna je jedna z nejoblíbenějších knihoven pro počítačové vidění a zpracování obrazu.

S pomocí této knihovny se v projektu zpracovávají obrazy z kamery.

KNIHOVNA TENSERFLOW

Open-source uvolněný společností Google roku 2015. TF je jedna z knihoven, která umožňuje pracovat a vytvářet neuronové sítě mnohem snáze než si je programovat celé samostatně. V dnešním světě díky TF knihovně a dalším podobným vznikají zajímavé projekty pracující s technikami deep learning, jako příklad je projekt na zkoumání lidských očních vad z fotografií lidské sítnice a nebo projekt na rozpoznávání řeči.

Dalším ještě více oblíbeným pokračovatelem TensorFlow je pak knihovna Keras, která je postavená na TensorFlow a dalších podobných knihovnách [12].

KNIHOVNA KERAS

Tato knihovna se vyznačuje především jednodušší formou, oproti knihovně Tensorflow. Svoji oblíbenost si získala i díky podpoře všech nejznámějších architektur neuronových sítí [12].

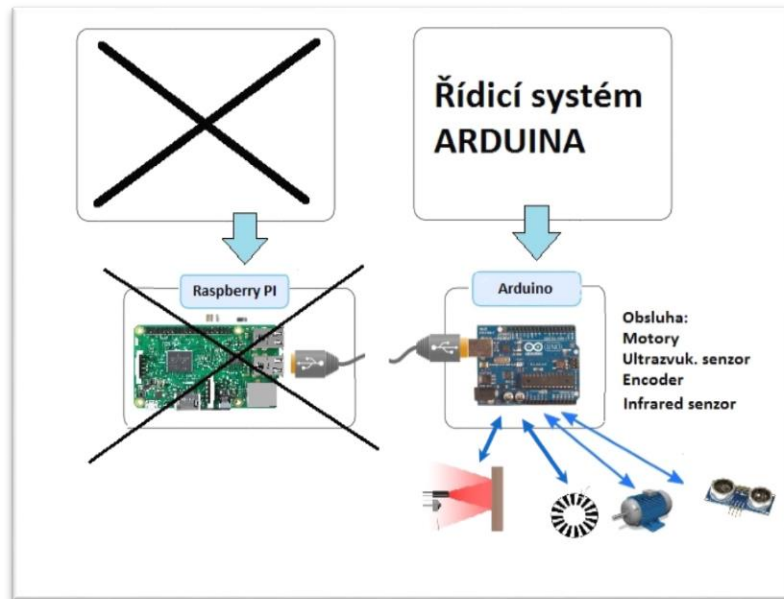
OPTIMIZER

Optimizer je algoritmus, který se snaží najít minimální hodnotu funkce. V případě machine learning hledá minimální hodnotu tzv. 'loss function'. Tato funkce je závislá na vahách jednotlivých uzlů neuronové sítě (NS) a definuje chybu predikce NS. Optimizer trénuje neuronovou síť (mění váhy jednotlivých uzlů) a snaží se tak najít optimální hodnotu vah. Optimizéry můžeme rozdělit na dvě kategorie

1. Optimizéry založené na gradientu funkce
2. Optimizéry založené na Hessové matici (matice druhých derivací funkce)

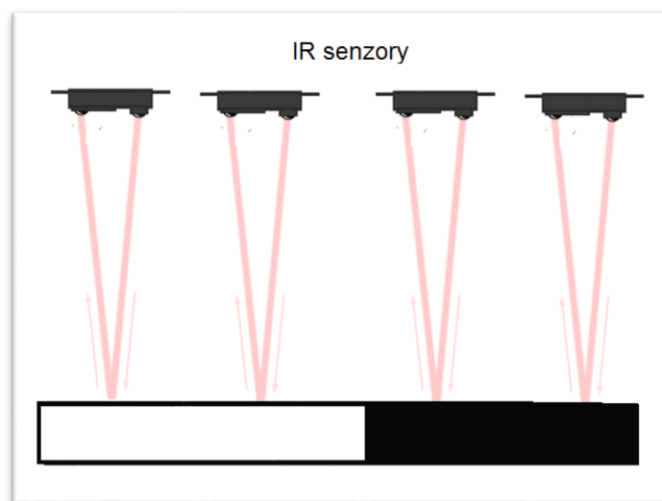
Pro projekt byl použit optimizer ADAM (Adaptive Moment Estimation), který patří do první kategorie. Tento optimizer je díky své rychlosti a efektivnosti jeden z nejpoužívanějších [28].

4.1 Algoritmus pracující infračervenými senzory



Obr. 4-2 Schéma systému s IR senzory

Jak jde vidět výše na schématu, pro tento algoritmus byla použita pouze deska Arduino, která zajišťuje veškeré řízení robota pouze pomocí IR senzorů (z kapitoly 3).



Obr. 4-3 Infračervené senzory pro detekci krajní čáry silnice

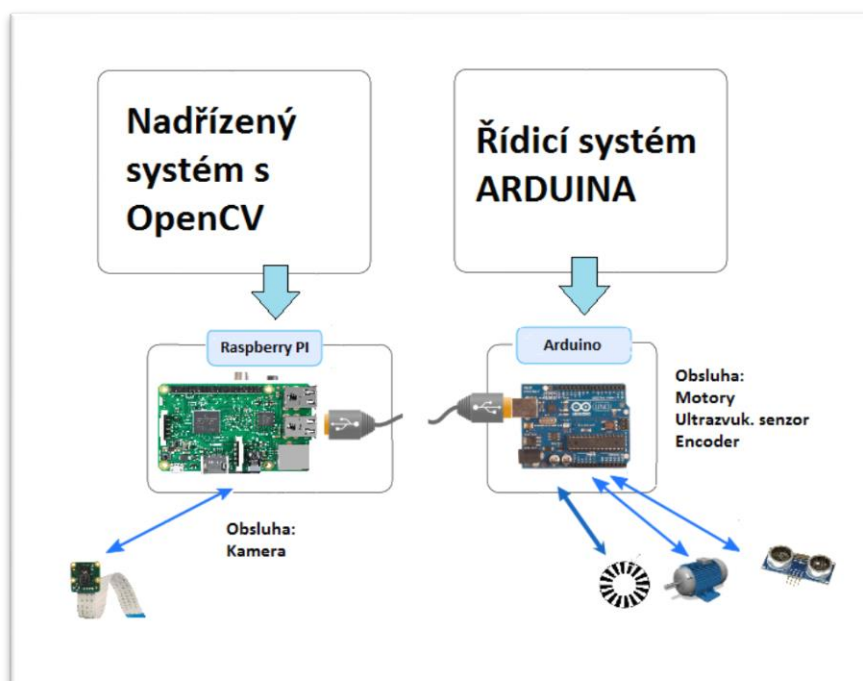
Tento algoritmus je nejjednodušší z těchto tří algoritmů a je postaven na principu, kdy se robot neustále snaží udržet si krajní čáru uprostřed mezi čtyřmi infračervenými senzory.

Kdykoliv nastane stav, při kterém krajní čára není uprostřed 4 senzorů tak robot okamžitě reaguje a posílá PWM signály do pravých a levých motorů tak, aby se robot natočil správným směrem a měl krajní čáru zase uprostřed.

Také tento model obsluhuje ultrazvukové senzory v přední části robota, které mají za úkol v případě zaznamenání velké překážky, okamžitě zastavit robota. Robot může pokračovat v jízdě, jakmile je překážka před ním odstraněna.

A posledním senzorem, který je obsluhován systémem v Arduinu je enkoder jednoho z motoru, díky čemuž robot dokáže měřit ujetou vzdálenost a také odjet do požadované vzdálenosti, která mu je určena. Enkodér dává 2077 impulsů na jednu otáčku. Kolo robota má 5,5 cm a tím pádem za jednu otáčku robot ujede zhruba 35 cm.

4.2 Algoritmus pracující s obrazem z kamery



Obr. 4-4 Schéma systému

Na schématu výše můžete vidět, že systém zahrnuje tentokrát už dvě zařízení. Pro každé z těchto zařízení byl sestaven kód s příkazy pro sériovou komunikaci, díky kterému tato dvě zařízení spolu komunikují přes UART – USB sběrnice.

NADŘÍZENÝ SYSTÉM RASPBERRY PI S OPENCV

V počítačovém prostředí je fotka reprezentována jako tři vrstvy pixelů, kdy každá matice reprezentuje jednu ze základních barevných složek (RGB složky), avšak pro naši aplikaci si vystačíme jen s jednou složkou, díky čemuž se ušetří také výpočetní čas při zpracování obrazu. A proto jako jeden z prvních procesů je převedení fotografie z RGB složek na jednu složku ve stupních šedi.

1. První instrukce algoritmu je zachycení fotografie silnice.
2. Převedení z RGB složek na jednu složku ve stupních šedi.

3. **Prahování**, neboli proces zpracování fotografie, kdy se ve scéně hledá objekt či oblast, která má stejnou úroveň intenzity a zároveň rozdílnou od intenzity pozadí.

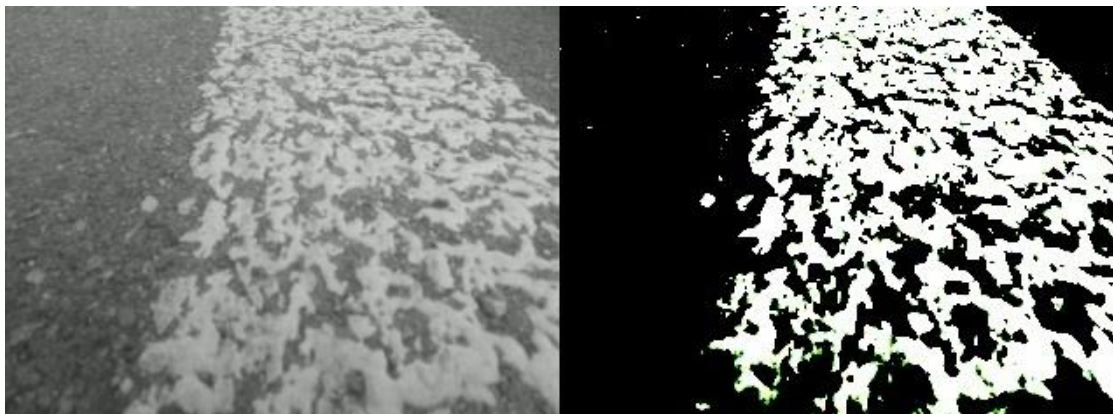
$$g(x, y) = \begin{cases} \log 1, & f(x, y) < T \\ \log 0, & f(x, y) \geq T \end{cases} \quad (1)$$

kde: $g(x, y)$ výstupní obraz
 $f(x, y)$ vstupní obraz
 T hodnota prahu

Proces prahování tedy provede změnu hodnot každého pixelu fotografie na 0, jestli je původní hodnota pixelu menší než nastavený prah nebo na 255 pokud původní hodnota byla větší.

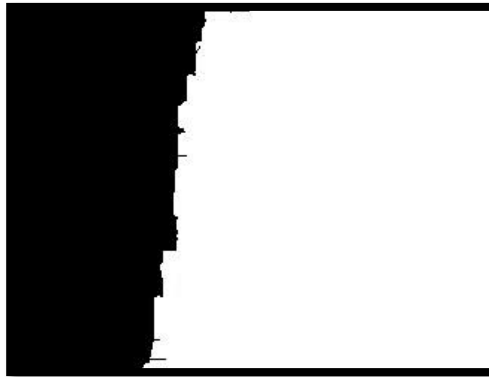
Prahování se provede pomocí funkce z knihovny OpenCV s názvem Threshold.

```
cv2.threshold(IMG, PRAH, cv2.THRESH_BINARY)
```



Obr. 4-5 Ukázka prahování

4. Binární fotografii po prahování se dále upraví pomocí vlastní funkce **edge()**, která ve dvou vnořených cyklech projde jeden řádek pixelů po druhém. Kdykoliv tato funkce narazí na výraznou hranu (5 bílých pixelů za sebou), tak tuto hranu zvýrazní. Zvýraznění probíhá tak, že se změní hodnota všech pixelů, které se nacházejí od hledané hrany až do konce tohoto řádku, na log 0.

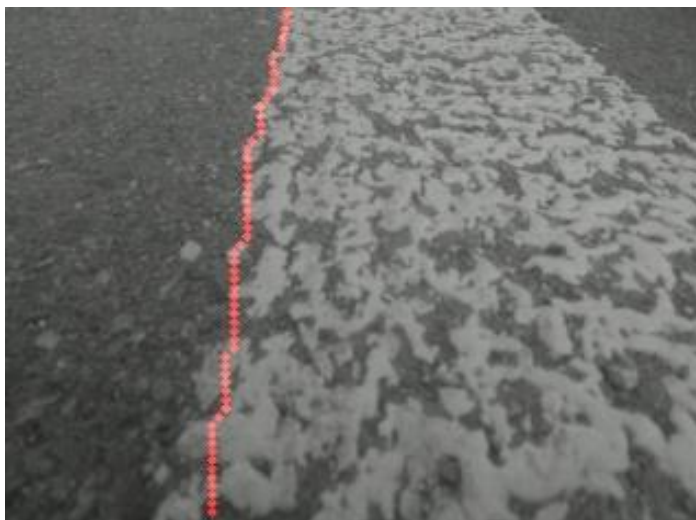


Obr. 4-6 Úprava binární fotografie pomocí funkce edge()

5. Z této upravené fotografie získáme body, které se nacházejí na hranici světlé a tmavé části obrázku pomocí vlastní funkce **points_of_edge()**, která má v sobě podobně jako funkce edge() dva vnořené cykly. Ta projde opět jeden řádek pixelů za druhým a v každém řádku se uloží x-sová souřadnice hledané hranice. Poté co cyklus projde 4 řádky se tyto x-sová souřadnice zprůměrují a uloží se jako jeden bod do pole Points [] se souřadnicemi (x, y), kde souřadnice y se určuje podle řádku, který se zrovna procházel.

$$x_N = \frac{1}{4} \cdot \sum_{i=1}^4 x_i \quad (2)$$

kde: x_i x-sová souřadnice i-tého řádku
 x_N x-sová souřadnice n-tého bodu z pole Points []



Obr. 4-7 Vizualizace vytvořených bodů

6. Dále v tomto algoritmu je potřeba získat rovnici přímky, která by nám charakterizovala umístění levého okraje postranní čáry ve fotografii. Pro tento proces byly testovány dvě metody:

- **Metoda nejstrmějšího gradientu**

U této metody se hledá rovnice přímky, která by nejlépe (s nejmenší chybou) odpovídala hledané přímce, která se nachází na hranici tmavé a světlé části obrázku 4-7.

$$y_{ref} = m \cdot x_i + b \quad (3)$$

kde: y_{ref} ... referenční hodnota na osy y hledané přímky v bodě x_i
 m směrnice hledané přímky
 b průsečík s osou y hledané přímky
 x_i souřadnice i-tého bodu na ose x

To znamená, že hledáme směrnici přímky m a průsečík s osou y b . Pro získání těchto dvou parametrů, je potřeba definovat chybu, která se v průběhu iteračního procesu bude minimalizovat.

$$error = \sum_{i=1}^N (y_i - y_{ref})^2 \quad (4)$$

kde: $error$...chyba aproximace
 y_i souřadnice i-tého bodu na ose x
 N počet bodů

Chyba je definována jako kvadratická funkce a proto, abychom našli minimum této funkce je třeba ji derivovat podle hledaných proměnných.

$$\frac{\partial error}{\partial m} = \frac{\partial \sum (y_i - (m \cdot x_i + b))^2}{\partial m} = error \cdot x_i \quad (5)$$

$$\frac{\partial error}{\partial b} = \frac{\partial \sum (y_i - (m \cdot x_i + b))^2}{\partial b} = error \cdot 1 \quad (6)$$

Díky derivaci chyby víme, směr nejstrmějšího stoupání této chyby, avšak my chceme chybu zmenšit, a proto je potřeba tyto derivace odečíst od původních hodnot našich proměnných.

$$m_{new} = m_{old} - \frac{\partial error}{\partial m} \quad (7)$$

$$b_{new} = b_{old} - \frac{\partial error}{\partial b} \quad (8)$$



Obr. 4-8 Přímka vytvořená metodou nejstrmějšího gradientu

- **Metoda průměrné přímky**

Podobného výsledku jako u metody nejstrmějšího gradientu lze docílit i jednodušší metodou. U této metody bylo vybráno 10 náhodných bodů z pole `Points[]`, které leží na hranici světlé a tmavé části obrázku 4-7 (tyto body jsou vizualizovány na obrázku 4-8), a z těchto bodů se pak vytváří 5 přímek (každá přímka prochází pouze 2 náhodně vybranými body). Z těchto 5 přímek se poté vypočítá průměrná hodnota parametru **m** a **b**.

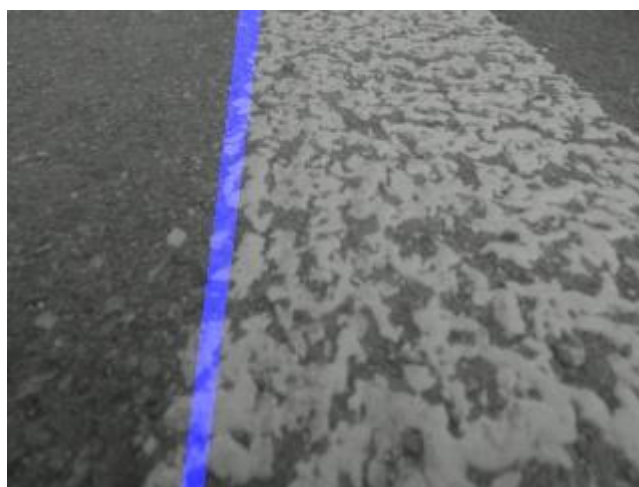
$$m_{new} = \frac{1}{N} \cdot \sum_{i=1}^N m_i \quad (9)$$

kde: m_{new} ...hledaná směrnice přímky
 m_i směrnice přímky i-té náhodné přímky
 N počet náhodných přímek

$$b_{new} = \frac{1}{N} \cdot \sum_{i=1}^N b_i \quad (10)$$

b_{new} ...průsečík hledané přímky s osou y
 b_i průsečík s osou y i-té náhodné přímky

$$y = m_{new} \cdot x + b_{new} \quad (11)$$



Obr. 4-9 Přímka vytvořená pomocí metody průměrné přímky

7. Jakmile se získají parametry této přímky lze dopočítat:

$$\varphi_{new} = \tan^{-1}(m_{new}) \quad (12)$$

kde: φ_{new} úhel svírající hledaná přímka s osou y

8. Tenhle úhel se poté porovnává s referenčním úhlem, který byl získán v při testování tohoto algoritmu (úhel, který odpovídá jízdě přímo) a pomocí PID regulace se dopočítá řídicí hodnota, která se posílá sériovou komunikací do podřízeného systému Arduino.

$$e = \varphi_{new} - \varphi_{ref} \quad (13)$$

kde: φ_{ref} referenční úhel

e odchylka od skutečné hodnoty

$$P = K_p \cdot error \quad (14)$$

kde: Pproporcionální člen
 K_pkonstanta proporcionálního členu

$$I = I + error \cdot K_I \quad (15)$$

kde: Iintegrační člen
 K_Ikonstanta integračního členu

$$D = K_D \cdot \left(\frac{\Delta e}{\Delta t} \right) \quad (16)$$

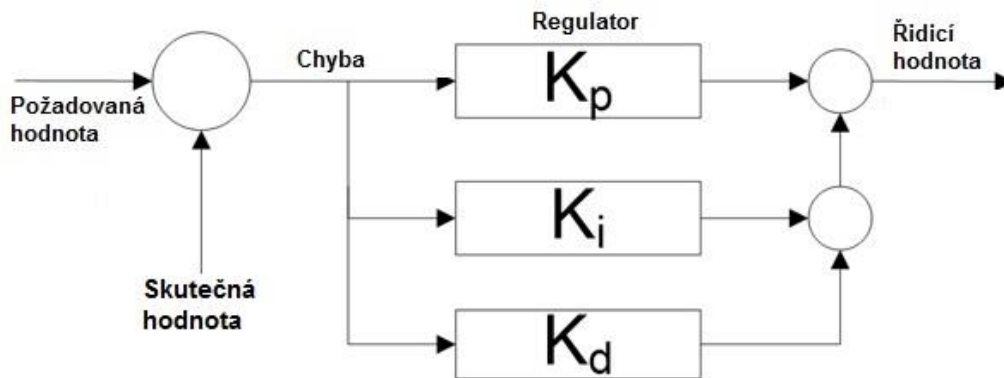
kde: Dderivační člen
 K_Dkonstanta derivačního členu
 Δe změna odchylky za dobu Δt
 Δt doba jednoho cyklu řídicí smyčky

$$\Delta e = e_i - e_{i-1} \quad (17)$$

$$\Delta t = t_i - t_{i-1} \quad (18)$$

$$Y_{out} = I + D + P \quad (19)$$

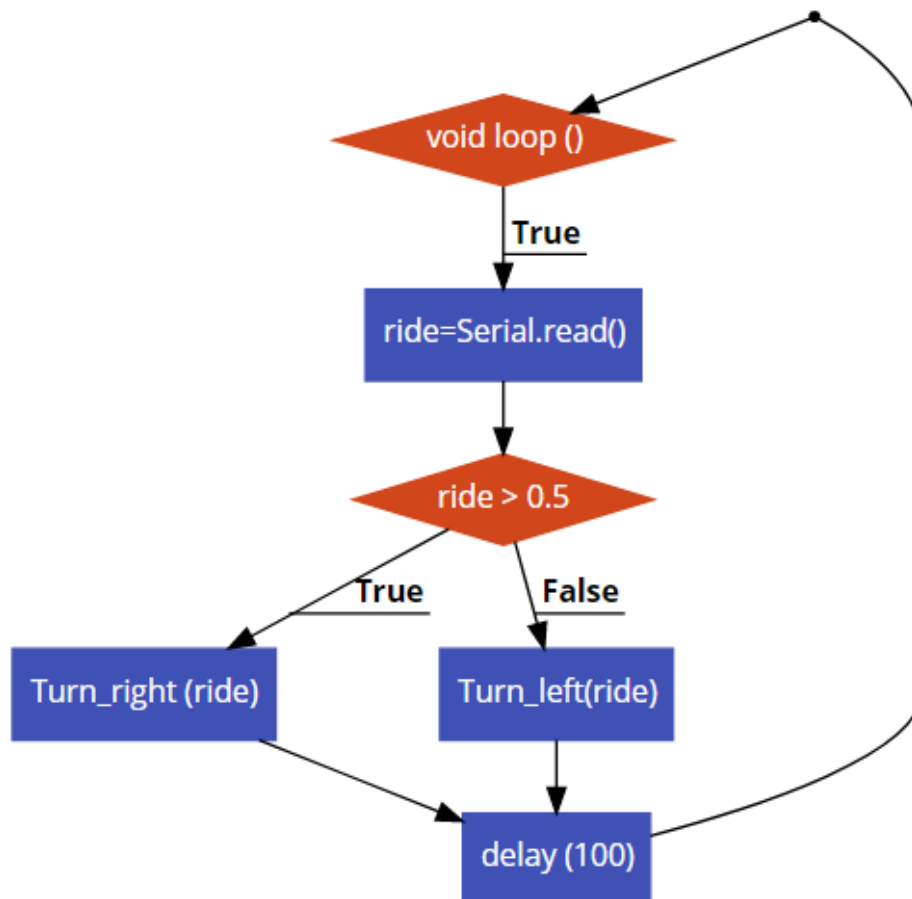
kde: Y_{out}řídicí hodnota odeslaná arduinu



Obr. 4-10 Schéma regulace s PID

ŘÍDICÍ SYSTÉM ARDUINA

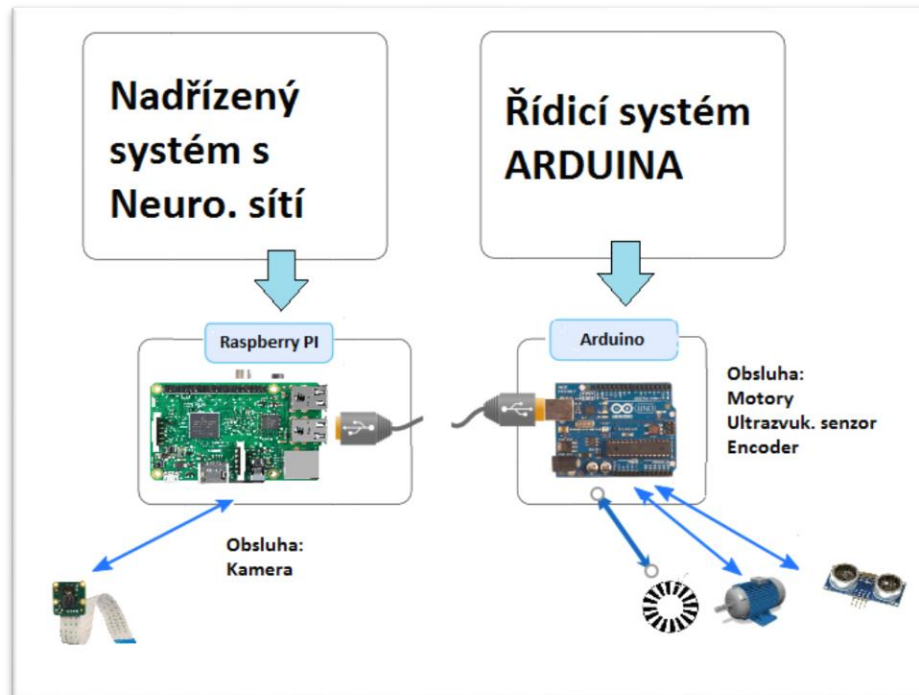
Jako vstupní hodnota tohoto systému je řídicí hodnota, která byla odeslána z nadřízeného systému Raspberry PI a tato hodnota určuje poměr stříd PWM signálů, které se posílají do pravých a levých motorů. Kvůli absenci natáčejících předních kol, je řízen pouze tímto poměrem stříd.



Obr. 4-11 Zjednodušené schéma řídicího algoritmu v Arduinu

Dále také Arduino obsluhuje ultrazvukové senzory a enkodér, kterým měří ujetou vzdálenost podobně jak už bylo zmíněno v kapitole 4.1.

4.3 Algoritmus s neuronovou sítí



Obr. 4-12 Schéma systému s Neuronovou sítí

NEURONOVÁ SÍŤ

Pro přiblížení tohoto algoritmu bude první představeno ve zkratce, co vlastně taková neuronová síť (NS) je a čím se liší od předchozích algoritmů. Neuronová síť je vlastně podoblastí strojového učení (machine learning), které se vyznačuje úplně jiným přístupem k řešení problému, než tradičního programování. U tradičního programování je výstupní hodnota algoritmu závislá pouze na vstupních hodnotách a na 'pravidlech', které byly zadány do kódu programátorem.

U Strojového učení je proces převrácený a aby se získala 'pravidla', které tentokrát nezadává programátor je potřeba v případě NS s učitelem nejprve tuto NS učit (supervised neural networks). To znamená, že pro určení 'pravidel', jak má naučená NS pracovat, je potřeba nejprve získat vhodná data, která dostatečně charakterizují řešený problém. Data společně se správným označením (labelem) jsou pak vstupními hodnotami při trénování NS.



Obr. 4-13 Schéma programovacích přístupů

PROCES UČENÍ NEURONOVÉ SÍTĚ

Na proces učení je tedy nutné si připravit data předem. V tomto projektu byly použity fotografie z kamery Raspberry PI a jako označení (label) byla u každé fotografie hodnota reprezentující poměr stříd PWM signálů.

Bylo pořízeno přes 1500 fotografií, které se získaly při jízdě platformy po krajní čáře silnice s externím ovládním. Pro tento proces bylo použito jako ovládací periferie obyčejný potenciometr, který byl připojený na desku Arduino, které posílalo PWM signály do 2 h-můstku, které dále ovládaly pravé a levé motory podle snímané analogové hodnoty na potenciometru a zároveň tuto hodnotu odesílalo desce Raspberry PI, která si hodnotu společně s pořízenou fotografií ukládala do své paměti.

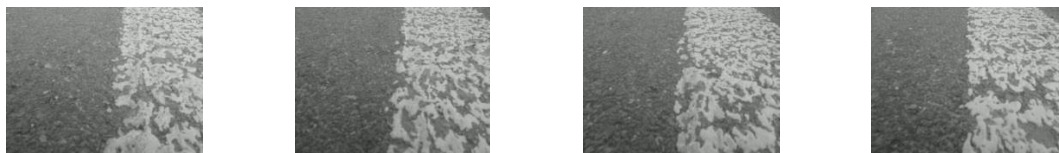
Fotografie pro metodu učení neuronové sítě byly pořízené v Brně na ulici Purkyňova vedle výzkumného centra Ceitec



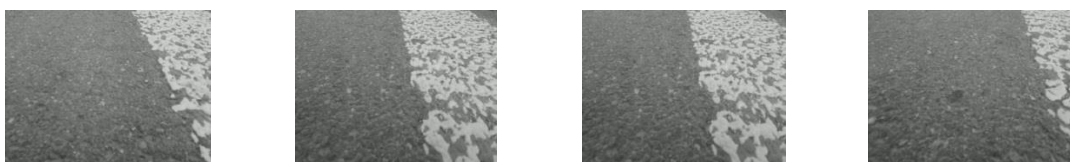
Obr. 4-14 Mapa s označením místa, sběru dat pro NS

Ukázka 3 pořízených fotografií:

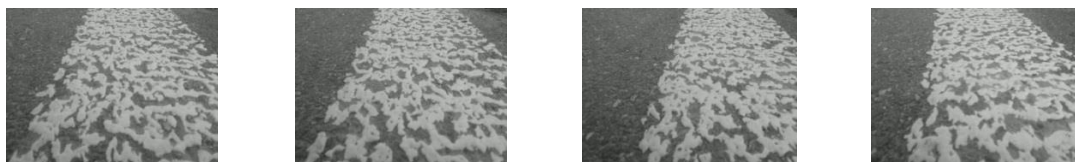
Tyto fotografie se snímali každou vteřinu a společně lablem se ukládali do paměti Raspberry PI ve formě JPG (čas_záznamu.jpg) a label do souboru CSV.



Obr. 4-15 Vzorke č.1 – Robot jede přímo



Obr. 4-16 Vzorke č.2 – Robot otočený více na levou stranu

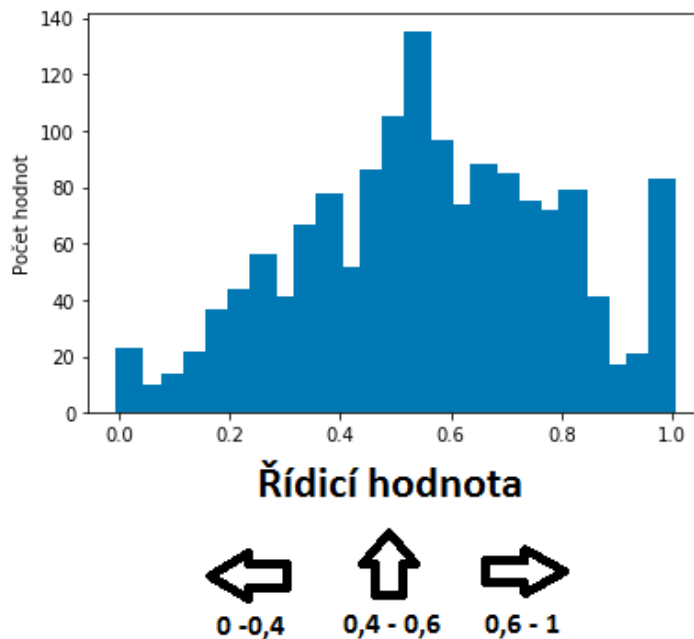


Obr. 4-17 Vzorke č.3 - Robot otočený více na pravou stranu

Samotný proces učení

Proces učení NS probíhal v prostředí **Google Collab**, což je webová aplikace, kterou si lze otevřít v prohlížeči. Velikou výhodou této aplikace je jednoduchá možnost sdílení kódu a také možnost využití zdarma výpočetního prostředku od společnosti Google na početní operace, které vznikají při učení NS.

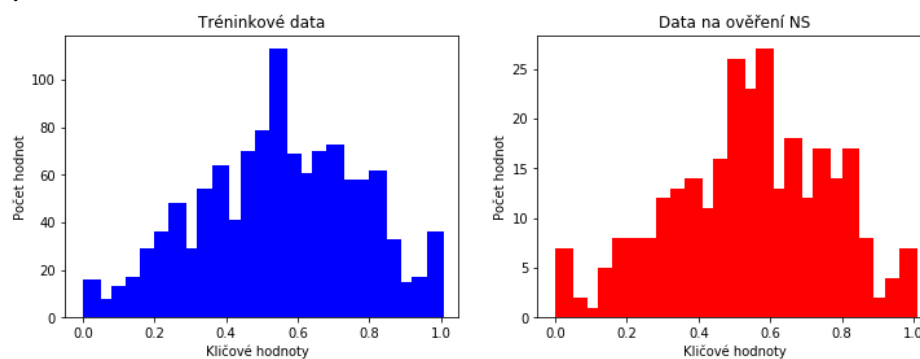
1. Na začátku scriptu se načtou získaná data z mého účtu na GitHubu [17] a importují se všechny potřebné knihovny.
2. Vizualizuje se počet jednotlivých fotografií s řídicí hodnotou (label), aby se zjistil charakter rozdělení.



Graf 1 Vizualizace počtu fotografií s danou řídicí hodnotou

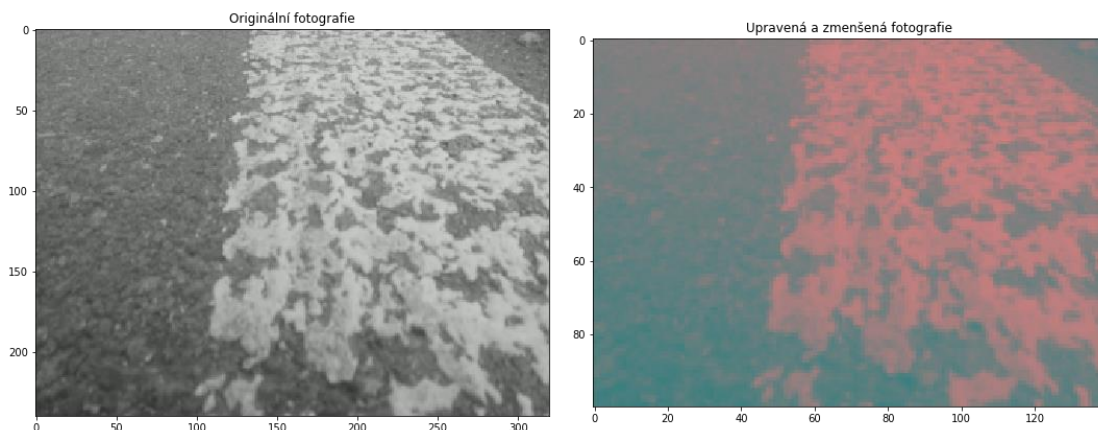
Při práci s těmito daty je potřeba hlídat, aby charakter rozdělení těchto dat odpovídal reálnému problému, který lze u řízení takového systému očekávat. V tomto projektu lze proto vidět, že pro jízdu rovněž odpovídá největší počet fotografií, což odpovídá reálnému problému.

3. Dále se data rozdělí na tréninková data a data potřebná na ověření modelu NS (validation data). Rozdělení je většinou v poměru 4:1



Obr. 4-18 Rozdělení dat pro proces učení NS

4. A ještě těsně před spuštěním iteračního procesu trénování NS se každá nahraná fotografie zmenší a převede z RGB složek na složky YUV, kde Y je jasová složka a U, V jsou barevné složky. Tento proces převádění barevných složek není zcela povinný, avšak je doporučený. Můžeme díky tomuto převodu snížit rozlišení U a V složky a složku Y zachovat původní, protože jasová složka je důležitější, jelikož lépe charakterizuje scénu na obrázku.

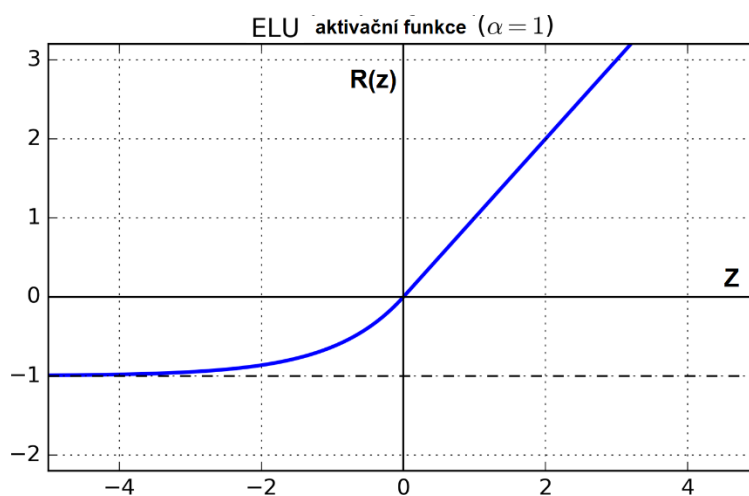


Obr. 4-19 Ukázka upravené fotografie

5. Pro sestavení modelu NS bylo potřeba jako první zvolit aktivační funkci. Nejdříve byla zvolena funkce nazývaná RELU. Ta díky problému zvaný “dying RELU” způsobovala, že se některé uzly neuronové sítě nebyly schopné učit se (měnit hodnotu svých vah) tak jako ostatní uzly. Tato funkce se nahradila funkcí ELU, která tento problém odstranila.

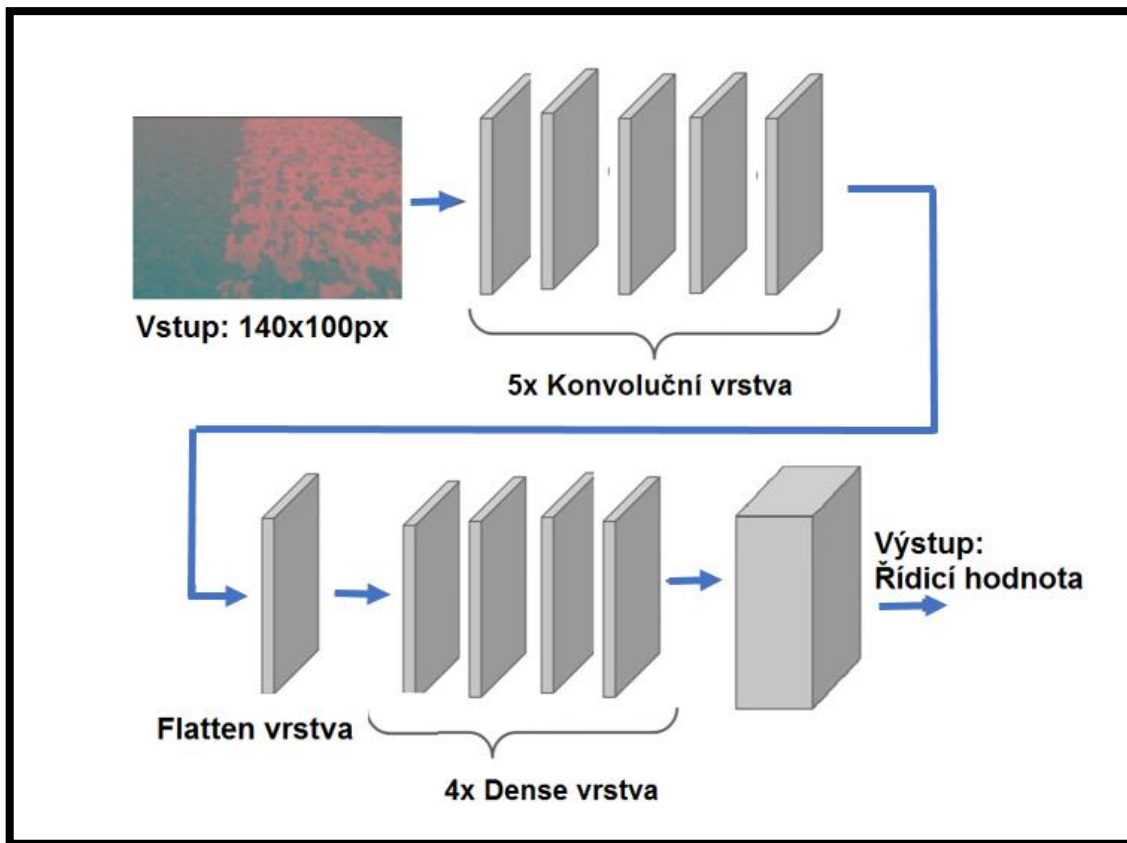
$$R(z) = \begin{cases} \alpha(e^z - 1), & z < 0 \\ z, & z \geq 0 \end{cases} \quad (20)$$

kde: $R(z)$ aktivační funkce
 z souřadnice z
 α konstanta (kladná hodnota)



Obr. 4-20 ELU aktivační funkce [22]

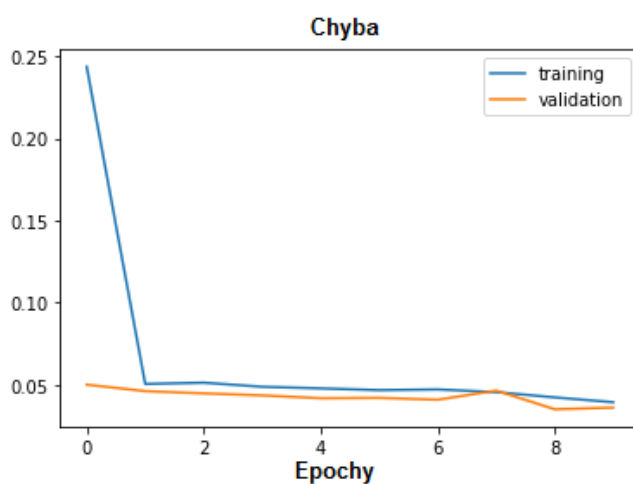
Samotný model neuronové sítě má 5 konvolučních vrstev a 4 dense vrstvy (plně propojené vrstvy) a mezi nimi jednu flatten vrstvu, která převádí výstup z poslední konvoluční vrstvy na jednosloupovou matici.



Obr. 4-21 Model NS

Jako optimizér NS byl nastaven **adam**, který je popsán v teoretické části na začátku kapitoly.

Počet kroků pro jednu epochu trénování bylo nastaveno 300 a samotný počet epoch 10, avšak chyba predikce byla už i po 2. epoše dostatečně nízká.



Obr. 4-22 Chyba predikce

	Training	Validation
epocha 1	0.24362	0.24362
epocha 2	0.05035	0.05035
epocha 3	0.05114	0.05114
epocha 4	0.04864	0.04864
epocha 5	0.04766	0.04766
epocha 6	0.04667	0.04660
epocha 7	0.04702	0.04702
epocha 8	0.04540	0.04540
epocha 9	0.04217	0.04217
epocha 10	0.03923	0.03923

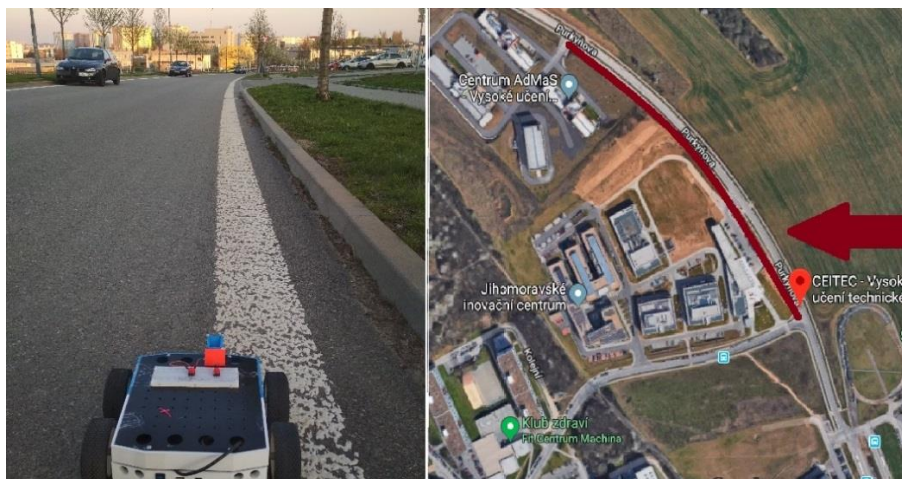
Tabulka 1 Hodnoty Loss function pro tréninková a validation data

- Získaný naučený model je implementován do nadřazeného systému Raspberry PI a jako vstupní hodnotu tohoto modelu se mu zadává každou vteřinu stejně upravená fotografie, jak při trénování a jako výstupní hodnota pak naučený model vrací řídicí hodnotu, které se odesílá přes sériovou komunikaci do desky Arduino.

Řídicí systém v arduinu, který řídící motory a obsluhuje senzory, je stejný jako v kapitole 4.2.

5 DEMONSTRAČNÍ ÚLOHA A POROVNÁNÍ ALGORITMŮ

Všechny tři algoritmy byly testovány na úseku ulice Purkyňová vedle budovy Ceitec.



Obr. 5-1 Místo demonstrační úlohy

Algoritmus pracující infračervenými senzory – kapitola 5.1

Jako první byl testován algoritmus infračervenými optickými senzory, který je řízený samotnou deskou Arduino.

Výhoda tohoto systému spočívá především v tom, že pro svoji práci nepotřebuje kameru ani desku Raspberry PI díky čemuž je tenhle ten systém s porovnání s ostatními levnější. Avšak je potřeba zmínit, že tenhle systém měl velké problémy na úseku silnice, kde byla částečně zničena postranní čára silnice a také při přímém slunečním záření.

Čas jedné smyčky (čas výpočtu): 32ms

- + nejlevnější varianta řízení
- chybné řízení při přímém slunečním záření
- chybné řízení na znečištěné nebo poškozené krajní čáře silnice

Algoritmus pracující s obrazem z kamery – kapitola 5.2

Tento algoritmus měl dvě testované části. V první části se testoval algoritmus, ve kterém byla implementována metoda nejstrmějšího gradientu a poté byla testována i metoda průměrné přímký.

U obou těchto metod byl hlavní problém čas výpočtu, který při snímání kamery s rozlišením 320x240px byl téměř 2s na jeden cyklus smyčky.

Při snížení rozlišení fotografií na 118x96px se čas snížil na 0,7s.

Čas jedné smyčky(čas výpočtu) : 792ms

- horší doba výpočtu
- problémy řízení při velice znečištěné krajní čáře silnice

Algoritmus s neuronovou sítí - kapitola 5.3

Jeden z problémů tohoto algoritmu je především nahromadění velkého objemu dat, s potřebným označením, které se použijí pro trénování NS. Avšak po získání naučeného modelu je samotný algoritmus už snadný. Veliká výhoda tohoto algoritmu je především všestrannost, kterou má tento algoritmus díky technikám machine learning.

Čas jedné smyčky(čas výpočtu) : 428ms

- + Lepší možnosti pro řízení v reálném prostředí než předešlé algoritmy
- Poměrně složité získání dat potřebné pro trénování NS

6 DALŠÍ VÝVOJ A MOŽNÉ UPLATNĚNÍ

Pro další vývoj tohoto projektu by bylo dobré sestavit nový model, menších rozměrů, s rychlejšími motory a s funkcí natáčení předních kol, pro lepší ovladatelnost robota. Také by bylo potřeba zabývat se problémem, který nastává na silnicích bez krajní čáry, tam by bylo potřeba navrhnout nový model řízení, který by si s takovým problémem poradil.

V tomto směru se nabízí nejpravděpodobnější řešení řídicího systému založeném na principech machine learning, do kterých patří i Neuronové sítě. Tyto principy se využívají i při vývoji autonomních automobilů a zvládají náročné aplikace s kterými by si systémy založené na tradičním programování jen těžko poradili.

Při dalším vývoji by bylo potřeba zvážit několik aspektů, s kterými by se robot potýkal v reálných situacích, jako například řízení robota na silnici bez postranní čáry a vyhýbaní se překážkám, na které při rozmisťování výstražných prvků může narazit, díry v silnici nebo odpadky, povalující se na okraji silnice.

Další možnost pro zefektivnění světelné signalizace je kooperace více autonomních robotů při rozmisťování výstražných světel. Také konstrukce robota by měla být dostatečně pevná, aby robot vydržel případné přejetí nepozorného řidiče.



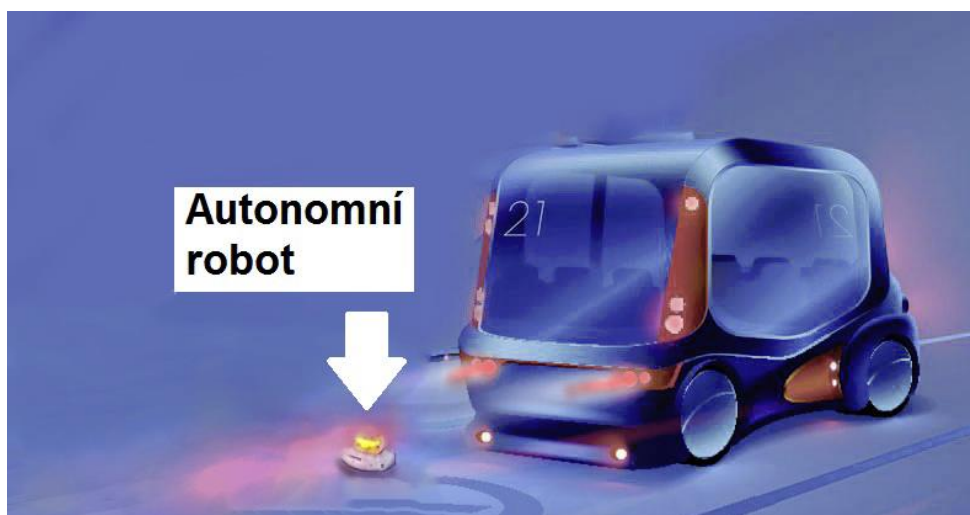
Obr. 6-1 Ilustrativní obrázek - Aplikace robota na silnici [26]

BUDOUCÍ MOŽNOST UPLATNĚNÍ

Je třeba také zvážit, jestli by takový systém měl uplatnění i budoucnosti silničního provozu, kde se předpokládá vývoj nových autonomních vozů, které by mohly sloužit jako osobní, hromadné i nákladní vozy.

V dnešním světě se lze už setkat například s plně autonomními taxi službami ve Phoenixu od společnosti Waymo [21]. A právě v takových systémech může vzniknout problém, jak vlastně označit místo dopravní nehody, když chybí člověk (řidič), který by v případě dopravní nehody mohl včas této místo nehody označit a proto by bylo možné přeprogramovat systém autonomního rozmístění výstražných prvků přímo pro tento problém.

V tomto případě by bylo dobré se zamýšlet i nad výsuvným, sklápěcím systémem, díky kterému by se v případě silniční nehody mohl robot dostat z podvozku autonomního vozu na zem na silnici a označit tak místo vzniklé dopravní nehody.



Obr. 6-2 Ilustrativní obrázek - koncept využití robota [25]

7 ZÁVĚR

Na začátku byla představena základní myšlenka pro vznik tohoto projektu a poté navazovalo představení cílů. Jelikož samotný projekt je však velice rozsáhlý nebylo cílem práce zprovoznit a vytvořit schopného autonomního robota, který by se dál používat v reálných situacích při silničních nehodách. Hlavním cílem bylo přiblížení základních komponentů, sériové komunikace a různých modelů řízení, které by pro takový systém mohly být použity.

Sériová komunikace se v projektu využívá pro komunikaci dvou spolupracujících zařízení, Arduino a Raspberry PI. Deska Arduino má především na starost řízení motorů, obsluhu ultrazvukového senzoru a enkoderu. Raspberry PI pak zpracovává obraz z kamery, která je na vrchní části robota a vytváří řídicí signál, který dále odesílá sériovou komunikací Arduino.

Postupně jsou představeny 3 algoritmy sledování bílé krajní čáry silnice. První z nich je nejjednodušší algoritmus, který pracuje s infračervenými senzory. Druhý a třetí poté využívá k řízení kameru. Tato kamera při řízení neustále zachycuje krajní čáru silnice. U třetího algoritmu je implementována neuronová síť, která byla trénována na datech pořízených na úseku silnice.

Dále se algoritmy odzkoušely na demonstrační úloze a zhodnotily se jejich výhody a nevýhody. Poslední kapitola se pak zabývá tématem dalšího vývoje systému.

Seznam použitých zdrojů

Bibliografické citace

- [1] JIRGES, R. Návrh robustního kolového robotického podvozku. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2018. XY s. Vedoucí bakalářské práce Ing. Michal Bastl.
- [2] ARDUINO MEGA 2560 REV3, [online], ARDUINO, [cit. 17.5.2019], Dostupné z: <https://store.arduino.cc/arduino-mega-2560-rev3>
- [3] RASPBERRY PI 3 MODEL B+, [online], RASPBERRY PI, [cit. 17.5.2019], Dostupné z: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>
- [4] Oficiální Raspberry Pi kamera V2. Rpiishop.cz [online]. [cit. 2019-03-31]. Dostupné z: <http://rpishop.cz/raspberry-pi/329-raspberry-pi-kamera-modul-v2.html>
- [5] 131:1 Metal Gearmotor 37Dx73L mm with 64 CPR Encoder, [online], Pololu, [cit. 17.5.2019], Dostupné z: <https://www.pololu.com/product/2827>
- [6] Arduino H-m]stek modul L9110S, [online], ARDUINO-SHOP.CZ, [cit. 17.5.2019], Dostupné z: <https://arduino-shop.cz/arduino/1233-arduino-h-mustek-modul-l9110s1458419853.html>
- [7] GENS ACE LIPO TATTU FPV serie - 2300mAh 11,1V 3S1P (45C), [online], PECKAMODELÁT, [cit. 17.5.2019], Dostupné z: <https://www.peckamodel.cz/ta-45c-2300-3s1p-gens-ace-lipo-tattu-serie-3s-2300-mah-45c>
- [8] Měřič vzdálenosti ultrazvukový pro Arduino Raspberry, [online], ARDUINO-SHOP.CZ, [cit. 17.5.2019], Dostupné z: <https://arduino-shop.cz/arduino/846-arduino-meric-vzdalenosti-ultrazvukovy-1500636000.html>
- [9] TCS230 Color Sensor Recognition Module for Arduino, [JPEG], MPJA.COM, [cit. 17.5.2019], Dostupné z: <http://www.mpja.com/TCS230-Color-Sensor-Recognition-Module-for-Arduino/productinfo/31582+MP/>
- [10] OpenCV [online]. California: OpenCV team, 2019 [cit. 2019-04-05]. Dostupné z: <https://opencv.org/license.html>
- [11] Edge Detector for Grayscale Images [online]. United State: MathWorks, 2015 [cit. 2019-04-05]. Dostupné z: <https://se.mathworks.com/matlabcentral/fileexchange/51124-shannon-edge-detector-for-grayscale-images>
- [12] The What's What of Keras and TensorFlow [online]. United State: upGrade Blog, 2018 [cit. 2019-04-07]. Dostupné z: <https://www.upgrad.com/blog/the-whats-what-of-keras-and-tensorflow/>
- [13] *Google maps* [online]. USA: Google, 2019 [cit. 2019-04-21]. Dostupné z: <https://www.google.com/maps/place/CEITEC>
- [14] PARALLEL COMMUNICATION OVERVIEW [ONLINE]. US: B+B SMARTWORKS, 2018 [CIT. 2019-04-21]. DOSTUPNÉ Z: <http://www.bb-elec.com/Learning-Center/All-White-Papers/Serial/Parallel-Communication-Overview.aspx>
- [15] PI NOIR CASE AND TRIPOD MOUNT [ONLINE]. US: PINSHAPE, - [CIT. 2019-04-21]. DOSTUPNÉ Z: [HTTPS://PINSHAPE.COM/ITEMS/29947-3D-PRINTED-PI-NOIR-CASE-AND-TRIPOD-MOUNT](https://pinshape.com/items/29947-3d-printed-pi-noir-case-and-tripod-mount)

- [16] RASPBERRY PI I2C / SPI / UART COMMUNICATIONS [ONLINE]. MBTECHWORKS, 2018 [CIT. 2019-04-21]. DOSTUPNÉ Z: [HTTPS://WWW.MBTECHWORKS.COM/HARDWARE/RASPBERRY-PI-UART-SPI-I2C.HTML](https://www.mbtechworks.com/hardware/raspberry-pi-uart-spi-i2c.html)
- [17] GITHUB REPOSITORY [ONLINE]. SAN FRANCISCO: GITHUB [CIT. 2019-05-03]. DOSTUPNÉ Z: [HTTPS://GITHUB.COM](https://github.com)
- [18] WHAT IS ENCODER?, VERA BASIC INTRODUCTION [ONLINE]. US, 2019 [CIT. 2019-05-12]. DOSTUPNÉ Z: [HTTP://ENCODER.COM/BLOG/COMPANY-NEWS/WHAT-IS-AN-ENCODER/](http://encoder.com/blog/company-news/what-is-an-encoder/)
- [19] IR SENSOR MODULE FOR ARDUINO / RASPBERRY PI [ONLINE]. EU: CDISCOUNT, 2019 [CIT. 2019-05-15]. DOSTUPNÉ Z: [HTTPS://WWW.CDISCOUNT.COM/BRICOLAGE/ELECTRICITE/XCSOURCE-IR-INFRAROUGE-OBSTACLE-PREVENTION-MODULE/F-1661416-XCS4894479459614.HTML](https://www.cdiscount.com/bricolage/electricite/xcsource-ir-infrarouge-obstacle-prevention-module/f-1661416-xcs4894479459614.html)
- [20] QUADCOPTER NAVIGATION IN THE FOREST USING DEEP NEURAL NETWORKS [ONLINE]. SWISS: IDSIA, 2016 [CIT. 2019-05-15]. DOSTUPNÉ Z: [HTTPS://WWW.YOUTUBE.COM/WATCH?V=UMRdt3zGgPU](https://www.youtube.com/watch?v=UMRdt3zGgPU)
- [21] TAXI BEZ TAXIKÁŘE [ONLINE]. US: AKTUÁLNĚ.CZ, 2018 [CIT. 2019-05-15]. DOSTUPNÉ Z: [HTTPS://ZPRAVY.AKTUALNE.CZ/EKONOMIKA/AUTO/JIZ-V-PROSINCI-SI-V-AMERICANE-BUDOU-MOCI-OBJEDNAT-TAXI-KTERE/R~851845B0E7EF11E8B7C1AC1F6B220EE8/](https://zpravy.aktualne.cz/ekonomika/auto/jiz-v-prosinci-si-v-ameriane-budou-moci-objednat-taxi-ktere/r~851845b0e7ef11e8b7c1ac1f6b220ee8/)
- [22] INTRO TO OPTIMIZATION IN DEEP LEARNING [ONLINE]. PAPERSPACE, 2018 [CIT. 2019-05-17]. DOSTUPNÉ Z: [HTTPS://BLOG.PAPERSPACE.COM/VANISHING-GRADIENTS-ACTIVATION-FUNCTION/](https://blog.paperspace.com/vanishing-gradients-activation-function/)
- [23] POROVNÁNÍ PWM REGULÁTORŮ VÝKONU [ONLINE]. ČR, 2018 [CIT. 2019-05-20]. DOSTUPNÉ Z: [HTTPS://NAVODY.ARDUINO-SHOP.CZ/TECHNIKUV-BLOG/POROVNANI-PWM-REGULATORU-RYCHLOSTI.HTML](https://navody.arduino-shop.cz/technikuv-blog/porovnani-pwm-regulatoru-rychlosti.html)
- [24] DALECKÝ, F. METODY ROZPOZNÁVÁNÍ OBJEKTŮ SNÍMANÝCH KAMEROU PRO ÚČELY ROBOTICKÉ MANIPULACE. BRNO: VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ, FAKULTA STROJNÍHO INŽENÝRSTVÍ, 2013. 41 s. VEDOUcí BAKALÁŘSKÉ PRÁCE ING. JIŘÍ KOVÁŘ.
- [25] FUTURE OF BUSES [ONLINE]. GERMANY: WISSENSFORUM, 2019 [CIT. 2019-05-20]. DOSTUPNÉ Z: [HTTPS://WWW.VDI-WISSENSFORUM.DE/EN/EVENT/FUTURE-OF-BUSES/](https://www.vdi-wissensforum.de/en/event/future-of-buses/)
- [26] ROAD FLARE BATTERY STATUS CHECK [ONLINE]. US: VIKTOR-M PHOTO, 2019 [CIT. 2019-05-20]. DOSTUPNÉ Z: [HTTP://VICTORMPHOTO.COM/EDITOR/?](http://victormphoto.com/editor/)
- [27] VEEROBOT INFRARED (IR) [ONLINE]. US: AMAZON, 2019 [CIT. 2019-05-21]. DOSTUPNÉ Z: [HTTPS://WWW.AMAZON.IN/VEEROBOT-INFRARED-PROXIMITY-OBSTACLE-DETECTING/DP/B0115NCT4U](https://www.amazon.in/veerobot-infrared-proximity-obstacle-detecting/dp/B0115NCT4U)
- [28] TYPES OF OPTIMIZATION ALGORITHMS [ONLINE]. TOWARDS DATA SCIENCE, 2017 [CIT. 2019-05-22]. DOSTUPNÉ Z: [HTTPS://TOWARDSDATASCIENCE.COM/TYPES-OF-](https://towardsdatascience.com/types-of-)

OPTIMIZATION-ALGORITHMS-USED-IN-NEURAL-NETWORKS-AND-WAYS-TO-
OPTIMIZE-GRADIENT-95AE5D39529F

- [29] STARSHIP'S ROBO-COURIERS HAVE COMPLETED 50,000 DELIVERIES [ONLINE].
ENGADGED, 2019 [CIT. 2019-05-23]. DOSTUPNÉ Z:
[HTTPS://WWW.ENGADGET.COM/2019/04/10/STARSHIP-ROBOTS-50000-DELIVERIES/](https://www.engadget.com/2019/04/10/starship-robots-50000-deliveries/)
- [30] HOW NEURAL NETWORKS POWER ROBOTS AT STARSHIP [ONLINE]. MEDIUM, 2018
[CIT. 2019-05-22]. DOSTUPNÉ Z:
[HTTPS://MEDIUM.COM/STARSHIPTECHNOLOGIES/HOW-NEURAL-NETWORKS-
POWER-ROBOTS-AT-STARSHIP-3262CD317EC0](https://medium.com/starshiptechnologies/how-neural-networks-power-robots-at-starship-3262cd317ec0)

Seznam symbolů a zkratek

Zkratky:

FSI	...	Fakulta strojního inženýrství
VUT	...	Vysoké učení technické v Brně
RGB	...	Barevné složky červená, zelená, modrá
NS	...	Neuronová síť
TF	...	TensorFlow
CSV	...	Comma separated values, tabulkový soubor
GT/s	...	Gagi bajt za sekund (přenosová rychlost)
MB/s	...	Maga bajt za sekundu (přenosová rychlost)
IR	...	Infrared, infračervené
YUV	...	Barevné složky
PWM	...	Pulzně šířková modulace
RPI	...	Raspberry PI
CPU	...	Centrální procesorová jednotka

Symbols:

U	...	napětí	[Ω]
I	...	proud	[A]

Seznam příloh

- Příloha 1 – Zdrojový kód Arduina z kapitoly 5.1. je uložen na přiloženém CD
- Příloha 2 – Zdrojový kód Arduina z kapitoly 5.2. je uložen na přiloženém CD
- Příloha 3 – Zdrojový kód RPI z kapitoly 5.2. je uložen na přiloženém CD
- Příloha 4 – Zdrojový kód Arduina z kapitoly 5.3.(proces učení) je na přiloženém CD
- Příloha 5 – Zdrojový kód RPI z kapitoly 5.3. (proces učení) je na přiloženém CD
- Příloha 6 – Zdrojový kód Arduina z kapitoly 5.3. je uložen na přiloženém CD
- Příloha 7 – Zdrojový kód RPI z kapitoly 5.3. je uložen na přiloženém CD
- Příloha 7 – Zdrojový kód z kapitoly 5.3.pro model NS je uložen na přiloženém CD