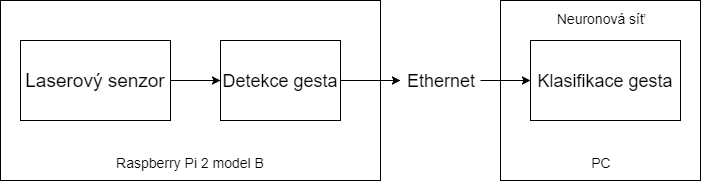
# Dokumentace: Laserový klasifikátor gest pomocí neuronových sítí

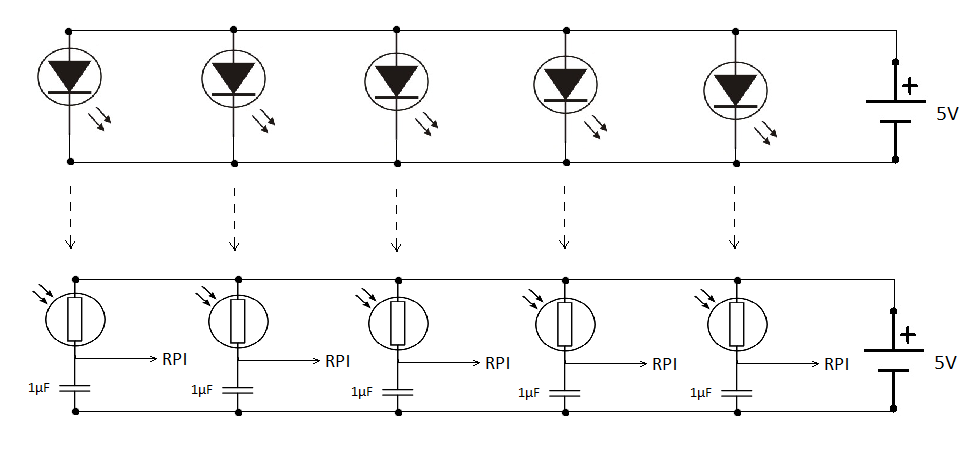
Jméno: Bc. Radek Pazderka

Vedoucí: Ing. Tomáš Goldmann

Laserový klasifikátor se skládá ze dvou hlavních částí. První část se skládá z laserového senzoru a detektoru gesta. Tato část se vykonává na externím hardwaru – na Raspberry Pi 2 model B. Druhá část obsahuje klasifikátor gesta, který přijme od Raspberry Pi přes ethernet. Klasifikace se provádí pomocí neuronových sítí. Výhoda této metody je ta, že se jednoduše a rychle trénuje a dokáže s velkou přesností predikovat, které gesto bylo z Raspberry Pi odesláno. Na obrázku níže je uvedeno blokové schéma celého projektu.



## Laserový senzor

Laserový senzor se skládá z pěti paralelně zapojených laserů na zdroj o pěti voltech. Tyto lasery přesně směřují na fotorezistory, které snímají světlo. Pokud se světelný paprsek laseru přeruší, tak to hned senzor detekuje a informuje o tom Raspberry Pi tím, že změní hodnotu pinu. Hodnoty pinu se pohybují v intervalu <0,1>, kde 0 značí – „Laser nesvítí na fotorezistor“ a 1 značí – „Laser přesně svítí na fotorezistor“. Na obrázku níže je uvedeno zapojení laserů a laserových senzorů.

## Detekce gesta

Gesto se začne detekovat v případě, že je alespoň jeden laser přerušen po dobu 0,6 vteřiny. Pokud gesto překročí tento nastavený časový interval, znamená to, že se má detekovat gesto. Gesto může trvat maximálně 2 vteřiny. Pokud trvá déle, automaticky se ořízne na velikost 2 vteřin (tzn. maximálně 50 hodnot z každého laseru). Snímání pinů z Raspberry Pi se provádí každých 0.04 vteřiny. Poté, co je gesto detekováno, jsou hodnoty předzpracované (viz. níže) a odeslané přes ethernet do PC, kde se klasifikují.

### Předzpracování hodnot

Hodnoty na pinu Raspberry Pi se mohou pohybovat v různých podintervalech intervalu <0,1> např. 0,4 pro – „laser nesvítí“ a 0,8 pro – „laser svítí“. Tento problém je řešený tím, že se hodnoty normalizují do celočíselného intervalu <0,100>, aby následná klasifikace gesta dostávala vždy hodnoty v předem definovaném intervalu. Další výhoda je to, že se nemusí posílat čísla s plovoucí desetinnou čárkou, ale jen celá čísla, na kterých se lépe trénuje neuronová síť.

## Komunikace

Komunikace mezi Raspberry Pi a PC je po ethernetu. Data se nejprve na straně Raspberry Pi zabalí a přes protokol TCP se pošlou do PC. Na Raspberry Pi běží TCP klient a na PC běží server.

## Klasifikace gesta

Klasifikace gesta se provádí pomocí neuronových sítí. Pro klasifikaci jsem využil framework Caffe, který umožňuje práci se strukturou, trénováním a testováním neuronových sítí. Pro tento projekt jsem vytvořil vlastní strukturu sítě, která se skládá ze tří plně propojených vrstev. Do sítě vstupuje vektor 5x50 (5 laserů a maximálně 50 hodnot) a výstupem při klasifikaci je hodnota 1-4, která symbolizuje členství gesta v určité skupině.

### Datová sada

Datová sada se skládá ze 4 různých gest. Každé gesto obsahuje list vzorků. Každý vzorek obsahuje vektor 5x50 hodnot. Pro každé gesto jsem ručně vytvořil 20 vzorků, které se mi posílaly do PC z Raspberry Pi. Pro trénování neuronové sítě je 80 vzorků velice malé číslo a bylo potřeba rozšířit datovou sadu. Hrozilo by velice rychlé přetrénování sítě. Rozšíření spočívalo v tom, že jsem každému vzorku náhodně přidával nebo ubíral jednotlivé hodnoty v rozmezí -5 … 5. Dále jsem různě posouval vektor doprava a doleva. Tím jsem rozšířil datovou sadu z 80ti vzorků na 2045 vzorků. Výslednou datovou sadu jsem uložil do databáze LMDB, kterou framework caffe podporuje.

### Trénování

Trénování spočívalo v rozdělení datové sady na dvě části. První pro trénování (cca 4/5 datové sady) a druhou pro testování (cca 1/5 datové sady). Po vytvoření datových sad bylo možné začít trénovat. Trénování proběhlo na 2000 iteracích. Časová náročnost trénování na grafické kartě NVIDIA GeForce GT 650M trvala necelé 2 minuty.

### Testování

Testování sítě probíhalo při trénování na každých 100 iteracích. Od iterace 500 byla úspěšnost sítě na 100 %. Což s největší pravděpodobností znamenalo to, že se síť přetrénovala. Proto jsem vybral iteraci 300, která dosáhla úspěšnosti 99,97 %. Rychlost se pohybovala okolo 1ms

### Závěr

Za velkou výhodu řešení považuji flexibilitu v natrénování různých gest. Vysokou rychlost trénování neuronové sítě (~2min) a klasifikace (~1ms). Výslednou přesností, která dosáhla 99,97% na testovací sadě. Praktická úspěšnost bude nižší. Ale v případě, že bychom dostatečně zvětšili datovou sadu, bylo by možné se k praktické hranici 99% úspěšnosti dostat.