VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2021 Jan Klouda

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV MIKROELEKTRONIKY
DEPARTMENT OF MICROELECTRONICS

AUTONOMNÍ ŘÍZENÍ BEZPILOTNÍHO LETOUNU NA ZÁKLADĚ ROZPOZNÁNÍ OBJEKTŮ
OBJECT RECOGNITION BASED AUTONOMOUS CONTROL OF AN UNMANNED AERIAL VEHICLE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR
Jan Klouda

VEDOUCÍ PRÁCE
SUPERVISOR
Ing. Jiří Janoušek

BRNO 2021

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Mikroelektronika a technologie**

Ústav mikroelektroniky

Student: Jan Klouda ***ID:*** 214380

Ročník: 3 ***Akademický rok:*** 2020/21

NÁZEV TÉMATU:

Autonomní řízení bezpilotního letounu na základě rozpoznání objektů

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s problematikou rozpoznání objektů v reálném čase pomocí bezpilotního letounu. Navrhněte autonomní řízení bezpilotního letounu na základě zpracování obrazů. Realizujte software umožňující řízení dronu v autonomním režimu, při kterém bude detekovat zvolené objekty, případně se k nim přibližovat. Proveďte experiment spočívající v řízení letounu pomocí jednoduché aplikace v jazyce Python, při kterém provedete detekci stanovených objektů. Vyhodnoťte možnosti použití realizovaného řešení a úspěšnost provedeného experimentu.

DOPORUČENÁ LITERATURA:

Podle pokynů vedoucího práce

Termín zadání: 8.2.2021 ***Termín odevzdání:*** 3.6.2021

Vedoucí práce: Ing. Jiří Janoušek

doc. Ing. Jiří Háze, Ph.D.
předseda rady studijního programu

Abstrakt

Tato práce se zabývá řízením autonomního dronu na základě rozpoznávání obrazu. Cílem je vytvořit program v programovacím jazyce Python, který určí polohu objektu v zorném poli kamery. Autonomní dron poté vykoná předdefinovaný úkon. Práce je rozdělena na teoretickou a praktickou část. V teoretické části jsou popsány metody zpracování obrazu a vyhodnocení objektů, následně také způsob komunikace mezi řídící stanicí a bezpilotním letounem pro řízení letounu. V části praktické je popsána realizace autonomního řízení letounu na základě detekce stanovených objektů v obraze. Poslední část se zabývá vyhodnocením úspěšnosti detekce objektů a přesností řízení vytvořeného řešení.

Klíčová slova

UAV, detekce objektů, dron, Python, autonomní létání.

Abstract

This thesis deals with the control of an autonomous drone based on image recognition. The goal is to create a program in the Python programming language that determines the position of an object in the field of view of the camera. The autonomous drone then performs a predefined operation. The work is divided into theoretical and practical part. The theoretical part describes the methods of image processing and evaluation of objects, the resulting methods of communication between the control station and the drone. The practical part describes the implementation of autonomous control of the aircraft based on the detection of objects in the image. The last part deals with the evaluation of successful detection objects and the control accuracy of the created solution.

Keywords

UAV, image detection, drone, Python, autonomous navigation.

Bibliografická citace

KLOUDA, Jan. *Autonomní řízení bezpilotního letounu na základě rozpoznání objektů* [online]. Brno, 2021 [cit. 2021-05-16]. Dostupné z: https://www.vutbr.cz/studenti/zav-prace/detail/134673. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav mikroelektroniky. Vedoucí práce Jiří Janoušek.

Prohlášení autora o původnosti díla

Jméno a příjmení studenta:	**Jan Klouda**
VUT ID studenta:	**214380**
Typ práce:	**Bakalářská práce**
Akademický rok:	**2020/21**
Téma závěrečné práce:	**Autonomní řízení bezpilotního letounu na základě rozpoznání objektů**

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné 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 závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné 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: 16. května 2021 ----------

podpis autora

Poděkování

Děkuji Ing. Jiřímu Janouškovi za veškeré rady, vedení, za jeho zájem, odbornou pomoc při řešení technických záležitostí, také za čas, který mi věnoval při zhotovování mé bakalářské práce.

V Brně dne: 16. května 2021 ----------

podpis autora

Obsah

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

ÚVOD

V posledních několika letech se bezpilotní letouny staly ústředním prvkem funkcí různých podniků a organizací. Již dnes drony prolomily bariéry v různých průmyslových odvětvích, a podařilo se jim také prorazit v oblastech, kde průmyslová odvětví stagnovala, nebo zaostávala. Disponují širokým využitím od rychlých dodávek po skenování vojenských základen. Bez ohledu na to, zda je na bezpilotní letoun pohlíženo jako na letadlo řízené dálkově pilotem, nebo se jedná o autonomní letouny, schopnost dosažení vzdálených a složitých oblastí s malou, nebo žádnou potřebnou pracovní silou, je využívána nejen v průmyslu a obchodu ale také ve vojenství. Tato bakalářská práce se zaměřuje na zdokonalení, zpřesnění a inovaci bezpilotních letounů ve způsobu jejich ovládání. Hlavní myšlenkou je zde přiblížit o něco více bezpilotní letoun člověku. Rozpoznání objektů je pro člověka přirozená činnost, ale pro počítač jen shluk pixelů. Za použití knihoven získáváme výsledky rozpoznání obrazu. Některé knihovny poznají člověka jen z jeho zápěstí, jiné potřebují pro rozpoznání jeho hlavní rysy jako je obličej, či silueta [1].

Každý z těchto systémů je samozřejmě jinak náročný na hardware, a tak stačí vytvořit vlastní verzi knihovny s předdefinovaným objektem, který má rozpoznávat, a nechat dron se soustředit jen na jeden objekt. Pokud lze z obrazu kamery například vyčíst, že pod dronem je umístěn heliport, na kterém má za úkol přistát, je již jen na komunikaci mezi mikropočítačem a letovou jednotkou, aby dron bezpečně a přesně přistál. GPS (Global Positioning Systém) je v tomto ohledu nepoužitelná, přistávací plochy pro drony bez kamerových systémů a senzorických přijímačů by byly mnohem větší vlivem nepřesnosti systému GPS. Ovšem nejde jen o přistávání, skrze kameru lze rozpoznat i QR (Quick Response) kódy, které mohou ukrývat aktuální příkazy nebo další úkoly pro bezpilotní letoun. Protože dron je ovládán dálkově (při použití s počítačem jako pozemní stanicí), je nutné zajistit stálé připojení s letovou jednotkou. Telemetrii, tedy komunikační kanál mezi počítačem a letovou jednotkou, zde může zajišťovat Wi-Fi (wireless fidelity) modul, nebo pro větší vzdálenost rádiové antény.

Pokud by byl použit mikropočítač, jako centrální řídící bod, který rozpozná, co je potřeba a sdělí řídící jednotce směr a výšku letu, může být umístěn přímo na bezpilotní letoun pro zamezení ztráty řídícího signálu. Systém GPS je zde ale také a zajišťuje hrubou kontrolu nad letounem a spolu s barometrem, akcelerometrem a gyroskopem vytváří stabilně letící letoun i za mírného větru. Letoun zůstává na poslední pozici, kterou získal od počítače a jen čeká na povel změny pozice. Z pohledu bezpečnosti může letová jednotka, nebo přímo počítač kdykoli rozhodnout, že se stala někde chyba, ať už jde o chybu mechanickou (ztráta komponentu, vybitá baterie), či softwarovou (ztráta signálu), a letoun poslat zpět odkud vzlétl, nebo s ním opatrně přistát pomocí GPS a barometru. Pokud chce operátor jakkoli zasáhnout do probíhajícího úkonu, lze letoun ručně přepnout do módu manuálního, nebo jej vypnout.

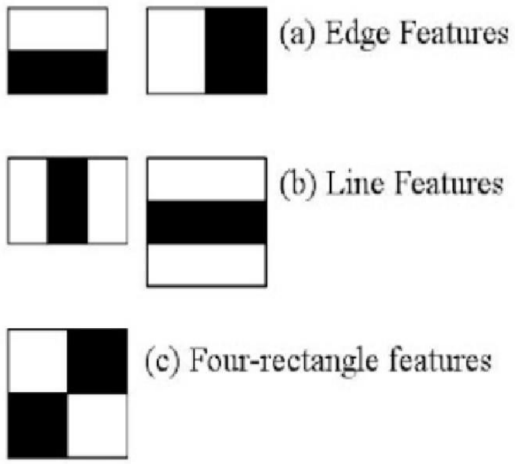
1. ZPRACOVÁNÍ OBRAZU

Základem rozpoznávání a zpracování obrazu je získání digitální formy reálného světa, která je později zpracovávána a vyhodnocována. Pro získání této digitální formy se převádí optické veličiny na elektrický signál. Tento proces lze chápat jako radiometrické měření. Snímání je ovlivňováno více faktory, jako je například odrazivost povrchu, či ozáření snímaného objektu. Vstupní informací při snímání je tedy nejen jas kamery/skeneru, ale i intenzita rentgenového záření, ultrazvuk a tepelné záření. Dále se uvažuje, že vstupním signálem je jasová složka z kamery [2].

Další krok při zpracovávání obrazu je digitalizace, tedy převod analogového signálu na signál digitální. Digitalizace je získána vzorkováním obrazu do matice a kvantováním do úrovní. Vzorkování se řídí Shanonovou větou, ze které plyne, že nejmenší detail v digitálním obraze musí být minimálně dvojnásobkem vzorkovacího intervalu. Proto je tedy zásadní volba rozlišení obrazu. Nízké rozlišení ztrácí informace o detailech a na druhou stranu příliš vysoké rozlišení je náročné k dalšímu zpracování obrazu. Velikost obrazu je uváděna v pixelech, rozlišení v jednotkách body/palec – DPI (dots per inch) [2].

1.1 Systém Haarových kaskád

Detekce objektů pomocí kaskádových klasifikátorů založených na Haarových funkcích je efektivní metoda detekce objektů navržená Paulem Violou a Michaelem Jonesem v roce 2001. Jedná se o detekci založenou na strojovém učení, kde funkce kaskády je trénována z mnoha pozitivních a negativních obrazů a poté se používá k detekci objektů na jiných obrázcích. Každá funkce je jedinou hodnotou získanou odečtením součtu pixelů pod bílým obdélníkem od součtu pixelů pod černým obdélníkem. Pro detekci klasifikátorů kaskády se zde využívá několik funkcí. Na obrázku 1.1 (a) je zde funkce pro detekci rysu hrany, (b) funkce pro detekci rysu čáry, (c) funkce pro detekci rysu šikmé čáry [3].



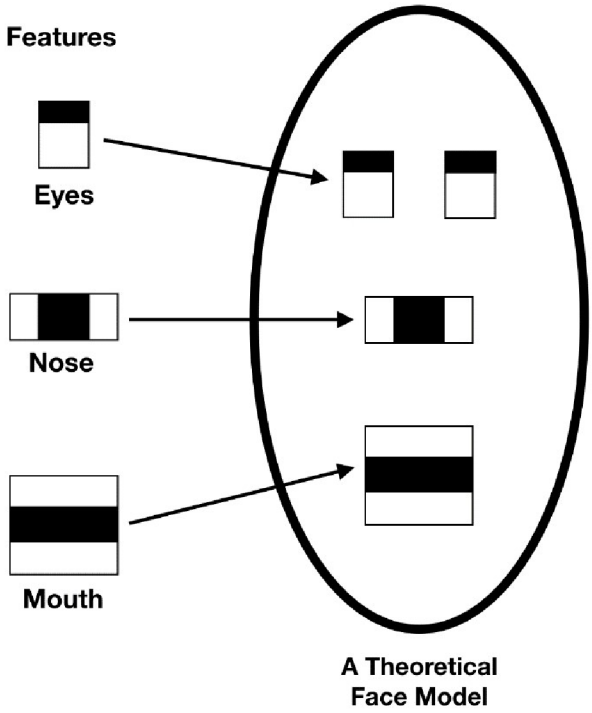
Obrázek 1.1 Funkce pro rozpoznávání Haarových příznaků [3]

Tyto funkce se převádí do matice 3x3, která se postupně násobí s každou stejně velikou maticí vytvořenou z jednotlivých pixelů černobílého obrazu, a tak způsobí zvýraznění některých oblastí (Haarovy příznaky) a jiné naopak vyhladí. Díky tomu je systém Haarových kaskád výborný pro detekci hran a čar. Detekce objektu použitím tohoto systému je však omezena osvětlením. Pokud hledaný objekt je na obraze špatně osvětlen, či část hledaného objektu zakryta, nebo otočena (Haarovy příznaky nejsou jasně viditelné), kaskáda nebude schopna rozpoznat objekt a může se stát, že při rozpoznání obličeje bude stačit jen pootočit hlavou a systém již nebude schopen rozpoznat o jaký objekt se jedná [4].

Výhodu Haarovy kaskády je, že systém se nemusí trénovat jako neuronové sítě, ale stačí naučit kaskádu na objekt ručně, tedy z několika obrázků ohraničit hledaný objekt a vytvořit kaskádu relativně malou datovou sadou. Celkové rozpoznávání objektu zahrnuje méně výpočtů, je tedy systém rychlejší, a méně náročný na hardware [4].

1.1.1 Haarovy příznaky

Tyto příznaky jsou typickými znaky pro daný objekt, které se objevují třeba na obličeji díky rozdílnosti světlosti a kontur obličeje. Nejvýznamnější kaskádový klasifikátor (tedy Haarův příznak) na obličeji je třeba nos, ústa a oči viz obrázek 1.2. Funkce rozpozná rozdílnost osvětlení a přiřadí Haarův příznak. Výhoda v práci s příznaky, které popisují jednotlivé pixely namísto pixelů samotných. Tento přístup ke zpracování obrazu zvyšuje rychlost detekce a umožňuje mnohem komplexnější přístup [5][6].



Obrázek 1.2 Příklad použití funkcí pro detekci Haarových příznaků pro obličej [7]

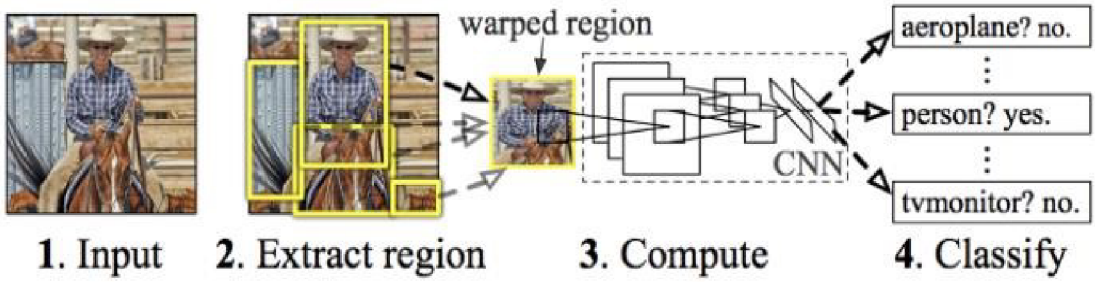
1.2 Detekce objektů založených na hlubokém učení

Využívání neurálních sítí pro detekci objektu odstraňuje nevýhody Haarových kaskád a podobných systémů pro detekci. Rozdíl je v tom, že neurální síť se sama učí, a proto lze sledovaný objekt detekovat z různých úhlů, což Haarova kaskáda neumožňuje. Přesnost při použití neuronových sítí tedy vzrostla. Naučená neuronová síť je tedy schopna detekovat člověka i pokud je vidět na obrázku jen jeho ruka. Pro naučení takovéto neuronové sítě tedy využíváme maticové váhy a postupně je upravujeme pro daný objekt. Pro naučení této sítě a spuštění detekce je zapotřebí mnoho výpočtů. Například knihovna Darknet dokáže pro výpočty neuronové sítě využít GPU (Graphics processing unit) počítače pomocí toolboxu CUDA (Compute Unified Device Architecture), a zvýšit tak rychlost detekci objektu [8].

1.2.1 R-CNN

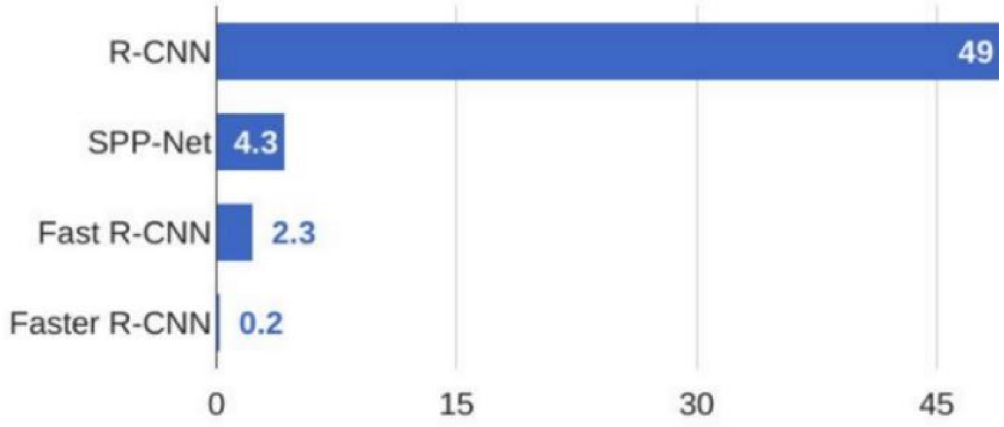
R-CNN (Region-based Convolutional Neural Networks) a varianty Fast R-CNN, Faster R-CNN jsou jedním z prvních detektorů založených na hlubokém učení (deep learning). Využívá algoritmus jako je Selective Search, který vygeneruje mnoho kandidátských oblastí, pomocí greedy algoritmu rekurzivně kombinuje podobné oblasti do větších, a následně použije vygenerované regiony k vytvoření konečných návrhů kandidátských regionů [9].

Kandidátské regiony se odevzdají CNN (Convolutional Neural Networks, tedy konvoluční neurální síť), kde neuronová síť funguje jako extraktor prvků, a za pomoci SVM (Support vector machines, tedy metoda podpůrných vektorů) se klasifikuje přítomnost objektu v kandidátském vektoru. Při vytváření kandidátských regionů se také využívá nejen předpovědi, kde se objekt pravděpodobně nachází, ale také další čtyři hodnoty (hodnoty posunutí), které pomáhají při úpravě ohraničování kandidátských oblastí. Hodnoty posunutí se využívají k tomu, aby se zvýšila přesnost ohraničujícího rámečku kolem objektu. Pokud by se tyto hodnoty nevytvářely, algoritmus by předpověděl přítomnost osoby, avšak tvář osoby v tomto návrhu by mohla být snížena na polovinu. Způsob detekce pomocí R-CNN ilustruje obrázek 1.3 [9].



Obrázek 1.3 Systém R-CNN [9]

R-CNN jsou pro detekci v reálném čase nepoužitelné z hlediska rychlosti. Pro každý obrázek je nutné klasifikovat 2000 kandidátských regionů. Celá akce detekce objektu tak trvá přibližně 47 sekund. Pro zrychlení času na detekci byla tedy vyvinuta verze Fast R-CNN, která zvýšila přesnost a zkrátila čas potřebný k provedení průchodu všemi využívanými systémy. Model se však stále spoléhal na algoritmus, který předával návrhy na kandidátské regiony. Verze Faster R-CNN se posunula blíže k detekci v reálném čase, díky sítěmi regionálních návrhů, odstranila se nutnost Selektivního vyhledávání a verze tak spoléhala jen na RPN (regionální návrhovou síť). Výhoda RPN je, že systém je konvoluční a může předvídat nejen rámečky s potenciálními objekty, ale také jaká je pravděpodobnost, že v daném kandidátském regionu se nachází hledaný obrázek. Faster R-CNN je tedy detekce objektu využívající hluboké učení (deep learning).Obrázek 1.4 ilustruje rychlost detekce jednotlivých typů systému R-CNN [9].

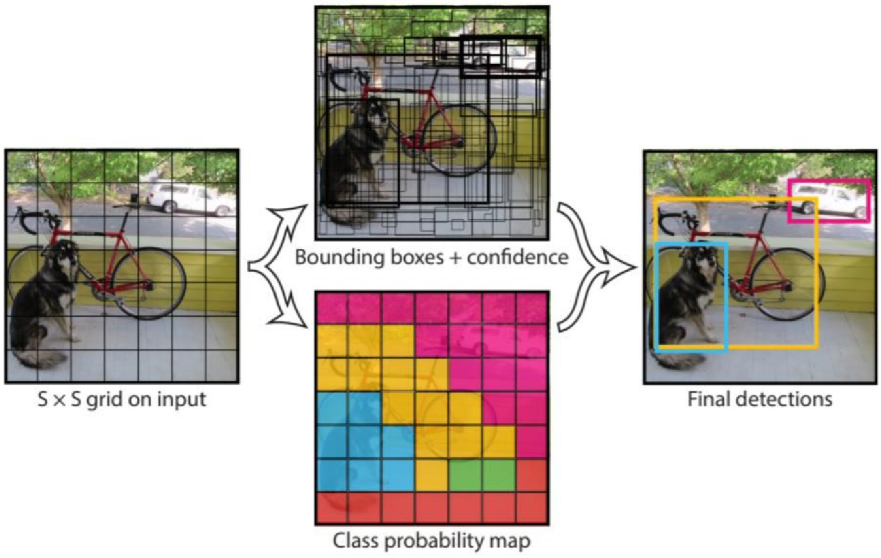


Obrázek 1.4 Porovnání rychlosti verzí R-CNN v sekundách [9]

1.2.2 YOLO

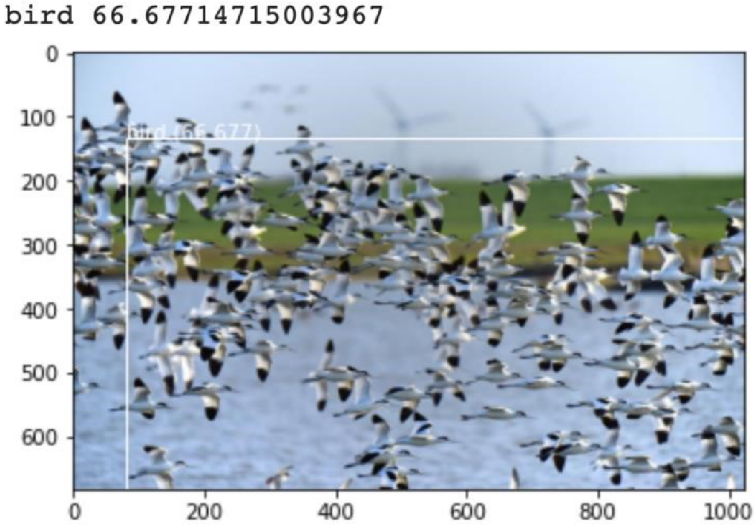
Rozpoznávací systém YOLO (You look only once) pracuje na stejném principu jako systém Haarových kaskád, avšak využívá jiný přístup. Na celý snímek (obraz) aplikuje jednu neuronovou síť, která rozdělí obraz do mnoho rámečků ohraničujících objekty v různých měřítkách. Tyto rámečky se vytváří na základě pravděpodobnosti pro každou oblast. Oblasti obrazu s vysokým počtem rámečků je tedy brána, jako velmi pravděpodobná oblast s výskytem hledaného objektu [9].

YOLO je skvělým příkladem jednostupňového detektoru a přistupuje k rozpoznávání objektu tak, že považuje detekci za regresivní problém. Bere tak vstupní obraz a současně se učí souřadnice a pravděpodobnost ohraničeného rámečku označené třídy. To znamená, že jedna konvoluční síť předpovídá ohraničující boxy a pravděpodobnostní třídy pro tyto boxy současně. Obrázek 1.5 ilustruje způsob, jakým systém YOLO detekuje objekty [10].



Obrázek 1.5 Způsob, jakým systém YOLO detekuje objekty [11]

Fungování tohoto systému lze popsat tak, že pořídíme snímek, a rozdělíme ho na vhodnou čtvercovou matici. Vytvoří několik ohraničujících rámečků (předpověď, jakou pravděpodobností se jedná o objekt) a současně pro každý rámeček vygeneruje hodnoty pravděpodobnosti třídy (jaká je pravděpodobnost že se jedná o hledaný objekt), a hodnoty posunu pro ohraničující rámečky. Algoritmus poté vybere ohraničující rámečky s pravděpodobností třídy vyšší, než je prahová hodnota (tato hodnota se volí uživatelem) a vykreslí ohraničený objekt. Systém YOLO je tedy řádově rychlejší než ostatní detektory s neuronovou sítí, avšak díky tomu, že detekce je jednostupňová, potýká se více s problémy přesností. Další omezení tohoto způsobu detekce je rozpoznání malých objektů, a proto může mít obtíže s detekcí třeba hejna ptáků (detekovat samostatné jedince) příkladem tohoto defektu je obrázek 1.6 [12].



Obrázek 1.6 Příklad nesprávné detekce hejna ptáků [11]

1.3 OpenCV knihovna

Knihovna OpenCV je open source knihovna, která se využívá k počítačovému vidění a strojovému učení. Byla vytvořena s cílem poskytnout společnou infrastrukturu pro aplikace a urychlit tak používáni vnímání strojů v komerčních produktech. Obsahuje více než 2500 algoritmů od klasických až po nejmodernější pro počítačové vidění a strojového učení. Algoritmy se využívají k detekci základních objektů a také celkovému zpracování videa či obrázku. Tato knihovna je velmi rozšířena a využívána ve společnostech, výzkumech a vládních orgánech. Pro příklad je využívána pro detekci obličejů v Japonsku, monitorování důlního vybavení v Číně, detekci topících se v Evropě, ke kontrole přistávacích drah v Turecku, a mnoho dalších aplikací po celém světě [13].

Rozhraní této knihovny je taktéž rozmanité. Využívá programovací jazyky Python, C++, java, a MATLAB. Podpora na systémech Windows, Linux, Android, Mac OS je samozřejmostí. Dnes je velmi důležité zpracování obrazu v reálném čase. Knihovna OpenCV využívá vektorový prostor a provádí matematické operace s těmito funkcemi. Zpracování obrazu je pro počítač souhrn některých operací za účelem získání vylepšeného obrázku, nebo z něj extrahovat některé užitečné informace. Když mluvíme o základní definici zpracování obrazu, pak „Zpracování obrazu je analýza a manipulace s digitalizovaným obrazem, zejména za účelem zlepšení jeho kvality“. Digitální obraz se zpracovává jako dvourozměrná funkce F(x,y), kde x a y jsou rovinné souřadnice. Každá tato souřadnice nese informaci o intenzitě nebo úrovně šedi v tomto jistém bodě. Jinými slovy, jakou barvu (nebos stupeň šedi) má mít pixel v jistém bodě. Obraz pro počítač není nic jiného než dvourozměrná (v případě barevných obrazů třírozměrná) matice, která je definována funkcí f(x,y) [13][14].

1.4 Technologie QR kódu

QR kód je dvourozměrný čárový kód vyvinutý japonskou společností Denso-Wave v roce 1994. V bakalářské práci byla využita pro detekci stacionárních cílů. Tato technologie je velmi rozšířená. Schopnost předávat informace obsažené v čárovém kódu může být využita v dalších postupech práce. Jedná se o veřejnou bezplatnou technologii. Zkratka „QR“ je z anglického Quick Response, tedy rychlá odpověď. Principem je zakódovaná zpráva do čtvercového QR kódu. Tento vytvořený kód je později natištěn na předmět, papír, či někde vyobrazen. Tento kód lze detekovat a dekódovat za pomocí chytrého telefonu, či zařízení, které disponuje softwarem pro dekódovaní QR kódu. Obvyklým způsobem se QR kód skenuje kamerou nebo fotoaparátem [15].

QR kód se skládá ze sady černých a bílých čtverečních modulů složených do čtvercové matice. Velikost matice je různá od 21 x 21 modulů až po 177 x 177 modulů s nárustem o 4 moduly v každém směru (tedy 40 možných rozlišení QR kódu). Příklady různých velikostí modů jsou vyobrazeny na obrázku 1.7. Použití 25 x 25 modulů je běžné pro webové adresy, 41 x 41 modulů pro odkazy na mapy a 57 x 57 modulů pro odkazy

na detailní kontakty. QR kód zpravidla obsahuje informaci textovou, kterou lze získat za pomocí softwaru pro dekódování QR kódů. Tato technologie disponuje i korekcí chyb, kde je kód s informací upraven takovým způsobem, aby došlo k přenosu informace, i když je povrh QR kódu znečištěn nebo poškozen. Kapacita kódové zprávy obsažené v kódu je vyjádřena v tabulce 1.1 [15].

Obrázek 1.7 Velikosti modulů QR kódů zleva: 25x25 modulů, 41x41 modulů, 57x57 modulů [16]

Tabulka 1.1 Kapacita různých znaků QR kódu [15]

Typ obsahu	Počet znaků
Číslice	7089
Písmena a číslice	4296
osmibitové data	2953
kandaži (japonské znaky)	1817

2. KOMUNIKACE S BEZPILOTNÍM LETOUNEM

Komunikace mezi mikropočítačem a letovou jednotkou je zásadní pro funkčnost celého bezpilotního letounu, a to z nutnosti přenášení instrukcí pro řízení bezpilotního letounu, které se vytvoří na základě zpracování obrazu. Největší předností bezpilotního letounu je hlavně to, že systém běží bez zásahu člověka, a není zapotřebí pilot pro pilotování. Letová jednotka sama o sobě může plnit jednoduché mise, jako je kroužení nad určitou oblastí, či let podle přednastavené trasy. Tyto mise jsou však během letu neměnné a pro většiny aplikací omezující, proto je zapotřebí komunikace s mikropočítačem. Na základě této komunikace lze vytvořit plně autonomní letoun, kde mikropočítač zpracovává informace o prostředí, ve kterém se letoun nachází, a dokáže letové jednotce sdělit požadované příkazy pro splnění mise. Letoun tak nemá přednastavenou trasu a mikropočítač ho během letu řídí podle toho, jak je v daném okamžiku žádoucí [17].

Komunikace mezi dronem a centrálním řídícím bodem (počítačem) pro řízení autonomního letounu může být vytvořena různými způsoby. Zde se budeme zabývat moduly pro přenos telemetrických dat, tedy o komunikaci, a to buď na krátké, či na dlouhé vzdálenosti. Každý z uvedených způsobů má své místo ve využití v praxi, a jde pak pouze o to, co očekáváme od bezpilotního letounu. Příkladem může být tak pozemní stanice, která se skládá z počítače s ovládacím softwarem, a zajišťuje tak zdroj povelů pro letoun. Pokud však je požadavek na bezpilotní letoun, který nevysílá žádný signál, je zde možnost připojení sériové linky, třeba USB (Universal Serial Bus) přímo do počítače nebo mikropočítače. Takové připojení bylo zapotřebí zrealizovat tak, že bezpilotní letoun nesl tento mikropočítač stejně jako třeba baterii nebo letovou jednotku. Je tedy snaha o to, zajistit tak výkonný mikropočítač o malé hmotnosti [17].

Reálné řízení (odezvu na příkazy počítače) realizuje letová jednotka. Jednotka za pomoci GPS, elektronického kompasu, akcelerometru, gyroskopu a barometru zajišťuje pohyb dronu takovým způsobem, jak mu definuje počítač. Pokud jde jednoznačně o rychlou komunikaci (mezi letovou jednotkou a počítačem) a letoun se bude pohybovat do několika desítek metrů od pozemní stanice, volí se přenos telemetrie přes Wi-Fi síťové připojení. S přenosovou rychlostí až 150 Mbit/s a snadnosti připojení, nemá v bezdrátových telemetriích konkurenci. Pro větší dosah se volí přenos telemetrie radiovými moduly, které informace přenáší na frekvenci 433 MHz nebo 868 MHz. Pokud však chceme bezpilotní letoun řídit na vzdálenosti vyšší než 10 km, jsou zde řešení v podobě komunikace přes satelitní síť, či mobilní sítě. Tyto řešení jsou zpravidla velmi drahé a využívají se v nezbytně nutných aplikacích [17].

2.1 MAVLink

MAVLink (Micro Air Vehicle Link) je protokol používaný pro komunikaci, pro přenos řídících signálů, informací o stavu letounu a dalších konfiguračních zpráv mezi drony a

pozemní řídící stanicí. Publikace informací probíhá hybridně, a to publish-subscribe (k odesílání datových toků) a point-to-point (pro konfigurační protokoly, protokol mise, nebo parametrů). Pro přenos informace se využívají takzvané dialekty, tedy sady zpráv podporované konkrétním systémem MAVLink [18].

Komunikace přes protokol MAVLink má výhodu, že nevyžaduje další rámování, takže je vhodný pro aplikace s velmi omezenou šířkou pásma komunikace. Podpora mnoha programovacích jazyků jako je Python, C, C++, a další je samozřejmostí. Protože se jedná o velmi rozšířený způsob komunikace s letovou jednotkou, protokol je podporován i na velmi mnoho operačních systémech od systému Android přes Windows až po mikrokontrolery jako je ATMega. Protože umožňuje až 255 souběžných systémů (letounů, vozidel) v síti najednou, jedná se o velmi široký způsob využití této komunikace. Protokol je postaven pro hybridní sítě, kde vysokorychlostní datové toky ze zdrojů dat (letoun) proudí do datových jímek (pozemní stanice), a jsou kombinovány s přenosy, které zajišťují zaručené doručení. Pro většinu odesílatelů telemetrie není znám jediný příjemce, místo toho letová jednotka a pozemní stanice potřebují stejný datový tok [18].

2.1.1 Topic Mode (punlish-subscribe)

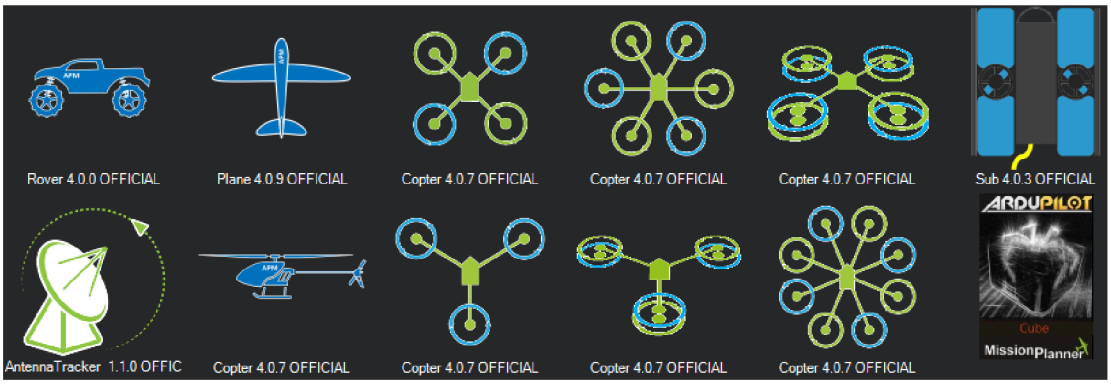
V tomto módu protokol nevydává cílový systém a ani ID (identifikace využívaná ve výpočetní technice) komponentu, tím se ušetří šířka pásma přenosu. Tímto stylem letová jednotka předává informace o své poloze, výšce, a tak dále. Tento typ režimu může přijímat více uživatelů najednou, a letová jednotka tedy nezjistí kdo signál zachytil [19].

2.1.2 Point-to-Point Mode

Režim Point-to-Point využívá ID cíle a cílové komponenty. Tedy letová jednotka má své číslo, přes které komunikuje pozemní stanic. Informaci dostane jen určitá letová jednotka. Tento mód zajišťuje garantované doručení a je využíván na zasílání parametrů, misí a příkazů pro letovou jednotku [19].

2.2 Firmware ArduPilot

ArduPilot je otevřený softwarový systém autopilota. Od roku 2009 je software vyvíjen týmem profesionálů, a dalších členů komunity. Je schopen ovládat jakýkoli přednastavený systém vozidla od dronu, po vrtulník, přes motorové čluny, až po pozemní vozidla. Základní systémy vozidla znázorňuje obrázek 2.1. Softwarová sada se instaluje jako firmware do mnoha produktů a značek, jako je Pixhawk, Durandal, F4By a další. Pomocí tohoto firmwaru nahraného do letové jednotky se letová jednotka stává plnohodnotným zařízením pro tuto bakalářskou práci. Zajišťuje komunikaci dronu, řízení, stabilizaci a veškerou funkční podporu pro autonomní řízení [20][21].



Obrázek 2.1 Příklad možných ovládacích systémů pomocí firmwaru ArduPilot

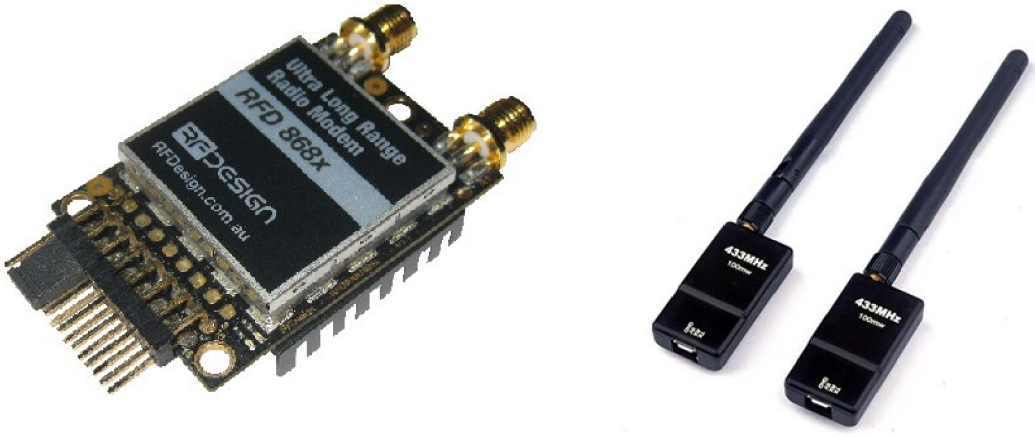
2.3 Přenos telemetrických dat

Telemetrická data, jsou na dálku přenášená data, v případě bezpilotního letounu může jít o výšku, rychlost, napětí baterie, posílání příkazů letové jednotce v podobě XML souborů (MAVLink knihovna), a mnoho dalších informací. Využívají se především k bezpečnosti, přehledu a lepším letovým výkonům letounu. Existují základní dva typy telemetrie. Telemetrie nezávislá, kde informace o letounu sbírá vysílač přímo z letounu, a vysílá je zpět do přijímače s obrazovkou. Dalším způsobem je telemetrie polo nezávislá, kde vysílač i přijímač se starají o přenos dat, a informace lze zobrazovat i na zařízeních jako jsou třeba displeje rádiích. Telemetrie integrovaná je pro klasické užití nejjednodušší. Informace přenáší vysílač, a data přijímá přímo rádio. Pokud jde o využití telemetrie pro účely bezpilotního letounu, využívá se polo nezávislá telemetrie [17].

2.3.1 Rádiová komunikace pro přenos telemetrie

Radiová telemetrie je jedním z nejjednodušších způsobů pro komunikaci mezi počítačem a letovou jednotkou. Jedná se o levnou, malou a lehkou jednotku přenášející informace s dosahem 300 m až několik kilometrů, při použití propojovací antény na zemi. Používá se zde open source firmware, který byl navrhnut pro dobré fungování s pakety MAVLink a integrován do letových firmwarů ArduPilot [22].

V České republice se používá dle legislativy frekvence 433 MHz a 868 MHz, avšak existují i rádia s frekcencí 915 Mhz, která nejsou v České republice povolena. Další modifikací modulů jsou moduly s dalekým dosahem, u kterých výkon dosahuje až do 10 mW. Tyto moduly jsou schopné komunikace až na 50 km. Na obrázku 2.2 je příklad modulů telemetrie vlevo 868MHz, vpravo 433 MHz [23][24].



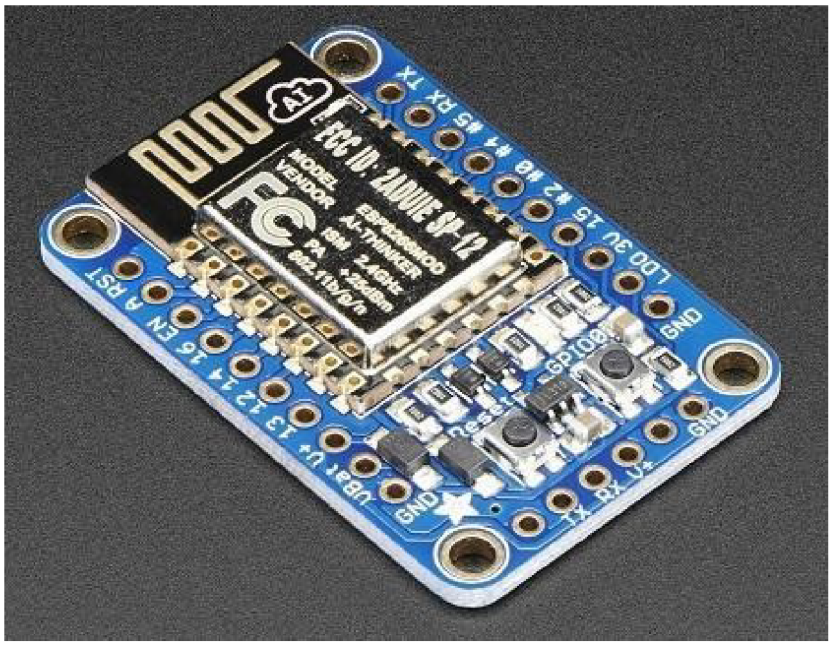
Obrázek 2.2 Telemetrické moduly, vlevo 868 MHz, vpravo 433 Mhz [25][26]

2.3.2 Přenos telemetrie 2,4 GHz (Wi-Fi)

Telemetrie přenášena 2,4GHz funguje na principu Wi-Fi, který pracuje na linkové úrovni ISO/OSI (Mezinárodní normalizace Propojení otevřených systémů). K letové jednotce je připojen modul, který vysílá síť Wi-Fi (2,4GHz). Na tento signál se lze poté připojit například pomocí počítače s Wi-Fi připojením, a zasílat pakety s příkazy (protokol MAVLink) přímo do letové jednotky [27][28].

Letová jednotka komunikuje s bezdrátovým modulem pomocí sériové komunikace. Dosah tohoto systému je velmi závislý na výkonu vysílače (tedy na modulu), také na počasí a překážkami mezi vysílačem a přijímačem signálu. Obvykle se uvádí dosah do několik desítek metrů. Pokud je počítač připojen k letové jednotce přes modul Wi-Fi, lze nejen zasílat data letové jednotce, ale také získat potřebné informace o aktuálním stavu letounu. Komunikace je obousměrná a probíhá přes adresy IPv4 (Internet Protocol version 4) [27][28].

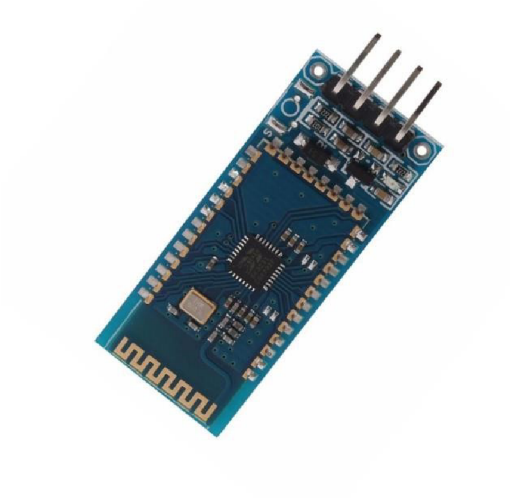
Snadno dostupné a levné moduly této telemetrie jsou moduly ESP. Nejčastějším typem je model ESP8266. Do tohoto modulu lze snadno nahrát MAVLink firmware pro správnou komunikaci modulu s letovou jednotkou. Obrázek 2.3 ilustruje příklad telemetrického modulu ESP8266 [27][28].



Obrázek 2.3 Modul ESP8266 [28]

2.3.3 Bluetooth telemetrie

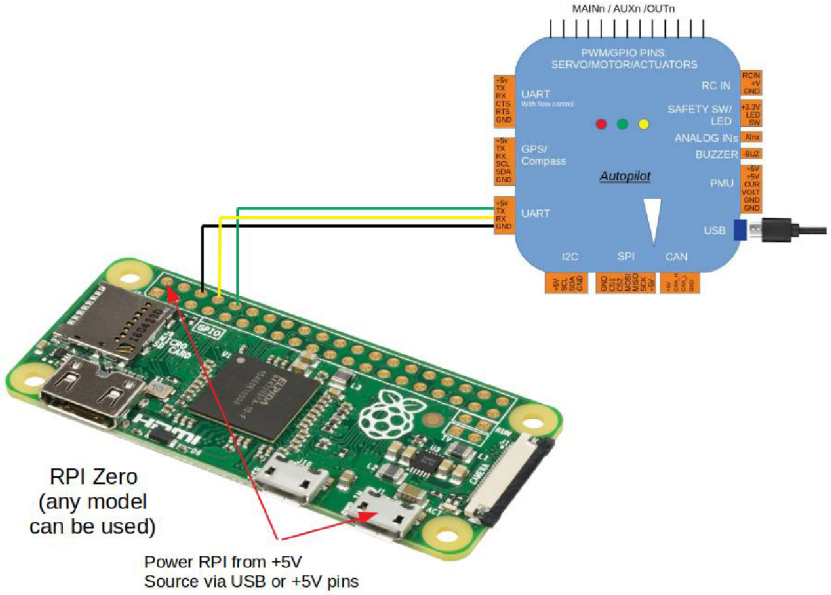
Pro přenos telemetrie jsou moduly Bluetooth velmi málo oblíbenou možností. Pracovní frekvence, tedy přenosová frekvence, je stejná jako u modulu Wi-Fi. Zde se již nevyužívá adresace přes IPV4 adresu, ale Bluetooth device Adress, neboli BD_ADDR. Podobně jako W-Fi pracuje na linkové úrovni ISO/OSI, musí však řešit více aplikační protokol, protože každý typ připojeného zařízení musí pro přenos definovat speciální komunikační protokol. Přenos signálu se udává 24 Mbit/s a je tedy horší alternativou modulu Wi-Fi. Výhodou tohoto modulu je nízká cena, a nevýhodou krátký dosah (maximálně desítky metrů). Na obrázku 2.4 je příklad modulu bluetooth telemtetrie [29].



Obrázek 2.4 Modul telemetrie Bluethooth [30]

2.3.4 Telemetrické připojení přes sérovou linku

Získávání telemetrie z letové jednotky pomocí sériové linky je kontaktní metoda. Při připojení musí být pozemní stanice (přijímací telemetrii) přímo spojena sériovou linkou s letovou jednotkou. Toto připojení využívám v bakalářské práci jen při použití mikropočítače Raspberry Pi jako pozemní stanice, protože mikropočítač je součást dronu, a připojení pomocí sériové linky je snadno realizovatelné. Aby mikropočítač komunikoval s letovou jednotkou, je zapotřebí propojit správné piny obou zařízení, viz obrázek 2.5. Protože letová jednotka může mít i více telemetrických výstupů, musíme nastavit její výstup a vstup na požadovanou rychlost a typ protokolu používaným pro komunikaci. Poté jen stačí povolit sériovou komunikaci na Raspberry Pi a připojení lze nyní vytvořit. Pro komunikaci s letovou jednotkou se používá protokol MAVLink [31].



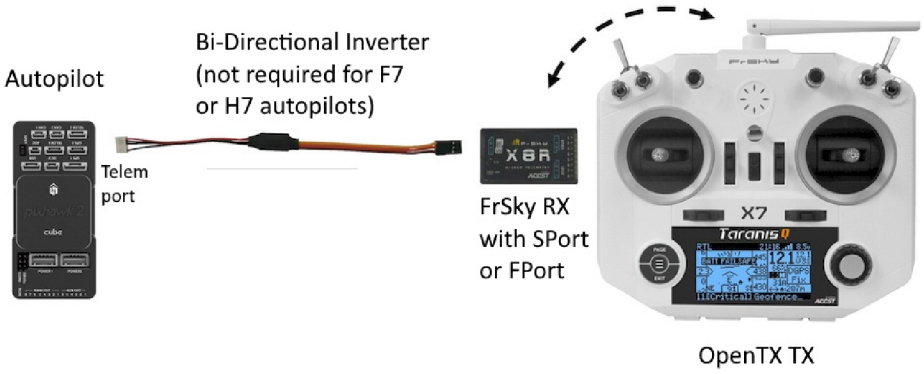
Obrázek 2.5 Propojení Raspberry Pi s letovou jednotkou pomocí sériové linky [31]

2.3.5 Další způsoby telemetrie

Telemetrie a komunikace s letovou jednotkou může být realizována mnoha způsoby. Příkladem je i satelitní telemetrie, která je 10x dražší než telemetrie 2,4 GHz, avšak teoretický dosah této telemetrie je až na druhou stranu země. Satelitní telemetrie využívá mobilní a jiné vysílače a dokonce i internet. Při použití internetu se využívá připojení TCP/IP (Transmission Control Protocol), při použití satelitního připojení se využívá technologie Iridium short burst data, poskytovanou společností Rock Seven Mobile. K využití satelitního připojení je nutnost doprovodného počítače (k letové jednotce), síťový modem a datový tarif u Amazon web Services. Aktivovaný satelitní komunikační modul připojený k mikropočítači (třeba Raspberry Pi) se nazývá UV Radio Room a pozemní stanice UV Hub, která využívá webové služby. Obrázek 2.6 znázorňuje modul satelitní telemetrie [32][33].

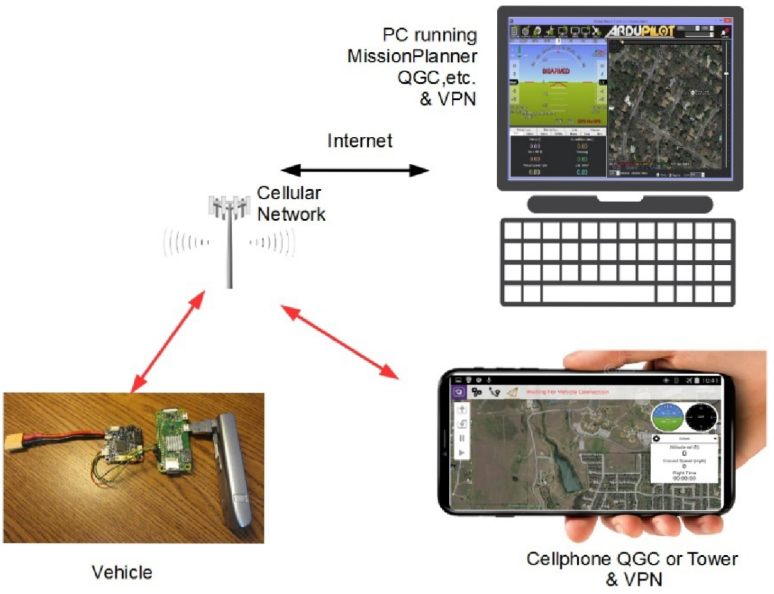
Obrázek 2.6 Satelitní přijímač RockBlock Mk2 [33]

Dalším způsobem je FrSky telemetrie, umožňující zobrazování požadovaných dat přímo na displeji ovladače dronu. Kompatibilní s firmwarem OpenTx a také i starší vysílače jako X9R (s omezenými schopnostmi). Tato telemetrie je využívána při letu bez pozemních stanic, nebo funguje jako záloha pozemní stanice. Příklad připojení této telemetrie je na obrázku 2.7 [34].



Obrázek 2.7 Příklad připojení telemetrie FrSky [35]

Dalším příkladem telemetrie je použití mobilní sítě 3G, 4G a 5G. Za pomoci těchto sítí je dosah telemetrie limitována pokrytím mobilních sítí a také výškou letu dronu, protože vysílače mobilních sítí jsou zaměřeny hlavně na pozemním pokrytí. Tento systém je také spojen s internetem a spojení lze uskutečnit tedy i s počítačem připojeným k síti. Využití sítě 5G se v letectví již vyvíjí a uskutečnili se již první testy řízení pomocí této sítě. Příkladem propojení letové jednotky s mobilní sítí je na obrázku 2.8 [35][36].



Obrázek 2.8 Propojení telemetrie využívající mobilní sítě [35]

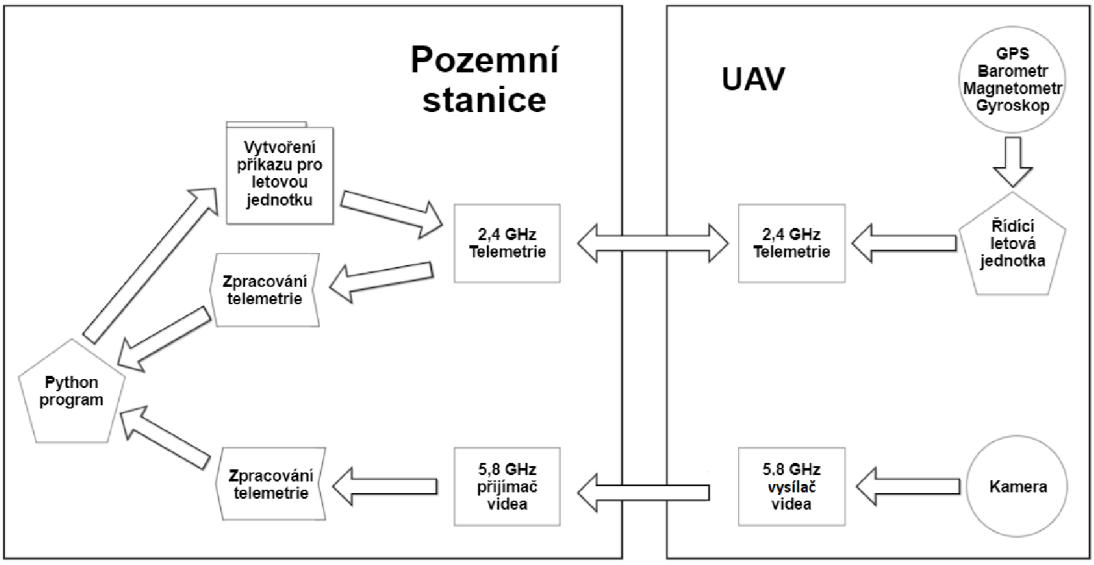
3. AUTONOMNÍ ŘÍZENÍ BEZPILOTNÍHO LETOUNU

Cílem úspěšné realizace autonomního řízení letounu na základě detekce stanovených objektů v obraze je vytvoření systému, který pomocí detekčních algoritmů propojí obrazová data s předem definovanými objekty a přiřadí k nim sady letových instrukcí, které následně odešle do řídící jednotky bezpilotního letounu. Realizace zapojení systému bude popsána v následujících kapitolách.

3.1 Zapojení autonomního systému

Zapojení celého systému popisuje blokové schéma na obrázku 3.1, kde UAV (Unmanned Aerial Vehicle) je bezpilotní letoun řízený pozemní stanicí. UAV se skládá z několik základních bloků. Řídící letová jednotka zde funguje jako hlavní řídící prvek letounu, který zpracovává informace z IMU (internal measurement units) jednotky, která provádí výpočet dat z akcelerometru, gyroskopu, barometru a GPS souřadnic. Řídící letová jednotka také přijímá povely z počítače (pozemní stanice) pomocí připojeného Wi-Fi telemetrického modulu. Letová jednotka z těchto všech informací vytváří PID (proporcionální, integrační a derivační regulátor) regulaci a vysílá PWM (pulse Width Modulation) signály pro regulátory, které provádí řízení jednotlivých motorů UAV. Pro realizaci UAV byla použita letová jednotka Pixhawk PX4 2.4.8 32 bit ARM Flight Controller.

GPS Souřadnice jsou využívány ke stabilizaci letu, magnetometr zajistí stálé natočení požadovaným směrem i při poryvech větru. Barometr, akcelerometr a gyroskop zajistí znalost relativní výšky, náklony, či rychlost letounu. Tyto informace nejen, že letová jednotka zpracovává, ale spolu s informacemi z pozemní stanice je schopna řídit letoun. Naměřené informace dále zasílá pomocí telemetrického modulu zpět do pozemní stanice, pro následné zpracování polohy a výšky. Na letounu se nachází také kamera, ze které se analogový obraz přenáší přes 5,8 GHz spojení do pozemní stanice. Tento obraz se zpracovává pomocí neuronové sítě v pozemní stanici. Pro zpracování obrazu byla z důvodu vyššího výpočetního výkonu zvolena pozemní stanice, kterou zajišťuje počítač, nebo Raspberry Pi. Obraz se zde zpracovává pomocí YOLO neuronové sítě, nebo Haarových kaskád. Ze zaslaných údajů od letové jednotky a zpracovaného obrazu z kamery, vytvoří pozemní stanice příkazy řídící letové jednotce a odešle je zpět pomocí telemetrického spojení. Lze tedy na základě zpracování obrazu z kamery vytvářet a předávat letové instrukce řídící jednotce, která generuje ovládací příkazy pro pohyb letounu daným směrem. Celé zapojení ilustruje obrázek 3.1, kde šipky znázorňují tok dat mezi jednotlivými bloky.



Obrázek 3.1 Blokové schéma zapojení

3.2 Realizace detekce QR kódů

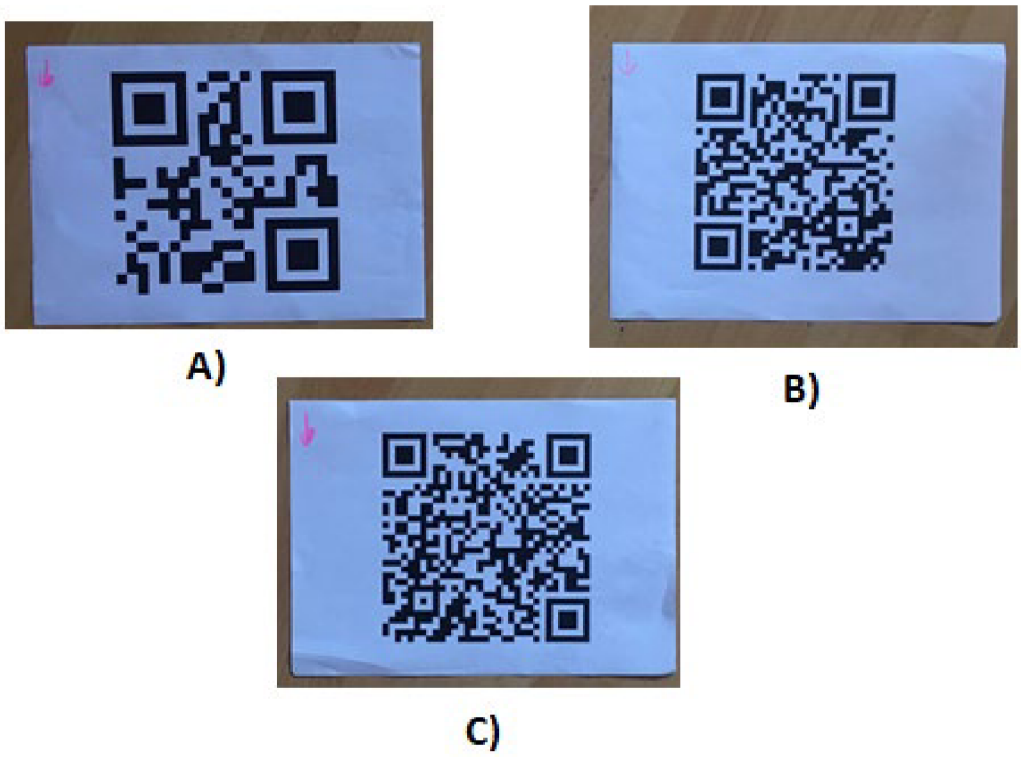
K realizaci dronu detekujícího QR kódy jsem zvolil Haarovy kaskády. Protože při detekci není nutný vysoký výpočetní výkon, je pozemní stanice v tomto případě realizována Raspberry Pi 3B. Celý výpočet detekce je tak spouštěn na 1.2GHz Broadcom BCM2837 64bit čtyřjádrovém procesoru, s využíváním až 1GB RAM. Obraz je pořizován z webkamery Logitech 720p, připojené do USB portu Raspberry Pi. Z důvodu snazšího přenosu telemetrie a hlavně videa, je pozemní stanice součásti UAV, tedy mikropočítač Raspberry Pi je připevněn na nosný rám ovládaného dronu. Připojení Raspberry Pi s letovou jednotkou Pixhawk je realizováno sériovou linkou ilustrovanou na obrázku 2.5. Protože pozemní stanice je součástí UAV, odpadá nutnost telemetrie 2,4 GHz, a 5,8 GHz vysílače a přijímače videa. Výsledný sestavený dron používaný pro testování řízení je vyobrazen na obrázku 3.2 [37].

Obrázek 3.2 Realizace výsledného dronu

3.2.1 Vytvoření vlastní předlohy pro detekci Haarových kaskád

Jako detekci objektu byla zvolena Haarova kaskáda. Jako předlohu pro detekci byl každý objekt (tedy QR kód) vyfotografován celkem 4, a na těchto trénovacích fotografiích byla neučena výsledná kaskáda. Vytvořil jsem celkově tři kaskády pro detekci třech stejných QR kódů, ale s rozdílným počtem modulů (bodů). Protože je zde detekován pouze QR kód, a není vytvořen dekódovací software, informaci obsaženou v QR kódu program nezjistí. Příklad trénovacích fotografií pro vytvoření třech kaskád je na obrázku 3.3. QR kódy jsou natištěny na papír A4 o rozměrech 210 × 297 mm, jednotlivé kódy jsou tištěny ve velikosti 160 x 160 mm.

Pro naučení neuronové sítě jsem využil programy a soubory s negativními fotografiemi od společnosti Dasaradh Makes. Postup při vytváření zahrnuje vložení pozitivních fotek do dané složky a ruční označení objektů, které chceme detekovat. Po označení se kaskády sami na objekty naučí, za pomocí velkého množství negativních fotografií (řádově tisíce negativních fotografií ve složce) a výsledkem je soubor .xml. Ten se dá použít pro detekci za pomocí knihovny OpenCV. Důvodem využití tohoto systému učení je jeho snadnost, rychlost a přítomnost mnoha negativních fotografií, které obsahují místa venkovního prostředí, kde se dron bude nejčastěji pohybovat. Tyto fotografie vytvoří kontrast mezi definovaným objektem a negativním pozadím obrazu [38].



Obrázek 3.3 Příklad trénovacích předloh QR kódů pro Haarovy kaskády: A) QR kód 21 x 21 modulů; B) QR kód 29 x 29 modulů; C) QR kód 33 x 33 modulů

3.2.2 Vytvoření kódu pro autonomní řízení

Protože se řídící jednotka Pixhawk připojuje pomocí sériového portu, využil se pro připojení příkaz viditelný ve zdrojovém kódu níže. Při připojení se definuje, jaký sériový port se pro připojení využívá a také jakou rychlostí budou zařízení komunikovat. Příklad kódu pro připojení je na obrázku 3.4.

```
12  print('Connecting...')
13  vehicle = connect('/dev/serial0',wait_ready=False,baud=912600)
14  vehicle.wait_ready(True, raise_exception=False)
15  print('connected')
```

Obrázek 3.4 Příkazy zajišťující připojení pomocí sériové linky

Po vytvoření připojení, se v počítačovém kódu definují potřebné neznámé, jako je třeba střed detekované obrazové oblasti získané později z webkamery, místo uložené předlohy Haarových kaskád, a další. Dalším krokem vytvořeného počítačového kódu je vytvoření nekonečné smyčky, ve které se načítají stisknuté klávesy. Podle stisknutých kláves se letové jednotce předávají informace o změně módu letu, nebo se spouští vytvořené podprogramy v počítačovém kódu. Jedním z podprogramů je kontrolovaný vzlet dronu do definované výšky, viz obrázek 3.5.

```python
def arm_and_takeoff(altitude):
    while not vehicle.is_armable:

        print("waiting to be armable")
        time.sleep(1)

    print("Arming motors")
    vehicle.mode = VehicleMode("GUIDED")
    vehicle.armed = True

    while not vehicle.armed: time.sleep(1)

    print("Taking Off")
    vehicle.simple_takeoff(altitude)

    while True:
        v_alt = vehicle.location.global_relative_frame.alt
        print(">> Altitude = %.1f m"%v_alt)
        if v_alt >= altitude - 1.0:
            print("Target altitude reached")
            #vehicle.mode = VehicleMode("LOITER")
            break
        time.sleep(1)
```

Obrázek 3.5 Příklad kódu zajišťující vzlet dronu

Po vzlétnutí dronu do definované výšky, je na operátorovi, jakou další funkci od autonomního řízení chce vykonat. Zda má dron prolétnout definovanou trasu podle souřadnic GPS, letět definovaným směrem, nebo lze dron manuálně řídit. Pro příklad se spustí systém pro detekci, tedy zapojí se Haarovy kaskády. Mikropočítač začne přijímat obraz z webkamery a na tento obraz poté aplikuje kaskádu, dále detekuje v obraze hledaný objekt. Například QR kód viz obrázek 3.6.

```python
    ret, img = cap.read()
    img = cv2.resize(img, (950, 720))
    if (type(img) == type(None)):
        break

    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

    cars = car_cascade.detectMultiScale(gray, 1.1, 3)

    for (x,y,w,h) in cars:
            ax=ay=0
            directionx=directiony=""
            cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255, 0), 2)
            cv2.putText(img, "dx: {}, dy: {}".format(x, y),(20,20), cv2.FONT_HERSHEY_SIMPLEX,0.75, (0,
            centerX=x+w/2
            centerY=y+h/2
            dX= centerx-centerX
            dY= centery-centerY
```

Obrázek 3.6 Příklad kódu aplikace kaskády a detekování objektu v zorném poli kamery

Tato detekce zjistí pozici detekovaného objektu v zorném poli kamery. Po úspěšné detekci, program vyšle definovaný příkaz do letové jednotky, spustí určitý podprogram, nebo se začne na detekovaný objekt přesně přistávat (pokud se třeba jedná o heliport). Při

přesném přistávání jsou využívány souřadnice objektu pro manévrování dronu. Pro manévrování se vypočítají pixely středu detekovaného objektu od středu zorného pole kamery a pokud detekovaný heliport není uprostřed zorného pole, mikropočítač vyšle korekční signál letové jednotce, aby se dron vystředil nad detekovaný objekt. Příkladem tohoto řízení je počítačový kód zobrazený na obrázku 3.7.

```
92              if dX > 30:
93
94                  ay=-gnd_speed
95                  directionx= "vlevo"
96              if -30.0 > dX:
97                  ay=gnd_speed
98                  directionx= "vpravo"
99
100             if dY > 30:
101                 ax=gnd_speed
102                 directiony= "dozadu"
103
104             if -30.0 > dY:
105                 ax=-gnd_speed
106                 directiony= "dopředu"
107
108
109             if ax==ay==0:
110                 set_velocity_body(vehicle, 0, 0, updown)
111             else:set_velocity_body(vehicle, ax, ay, 0)
```

Obrázek 3.7 Příklad kódu zajišťující výpočet pozice objektu v zorném poli kamery

Po vytvoření příkazu a výpočtu odchylky objektu vůči dronu, pokud bereme, že střed zorného pole kamery je i středem dronu (pokud je skutečnost jiná, je nutno zavést více výpočtů pro korekční let), je nutné vytvořit protokol MAVLink s letovými příkazy a odeslat ho do letové jednotky. Příklad vytvoření protokolu MAVLink s příkazem je ve zdrojovém kódu níže na obrázku 3.8.

```
56  def set_velocity_body(vehicle, vx, vy, vz):
57
58      msg = vehicle.message_factory.set_position_target_local_ned_encode(
59              0,
60              0, 0,
61              mavutil.mavlink.MAV_FRAME_BODY_NED,
62              0b0000111111000111,
63              0, 0, 0,
64              vx, vy, vz,
65              0, 0, 0,
66              0, 0)
67      vehicle.send_mavlink(msg)
68      vehicle.flush()
```

Obrázek 3.8 Příklad odeslání příkazu o změně letu protokolem MAVLink

Po dokončení požadovaného úkonu, lze probíhající kód bezpečně vypnout, pomocí klávesy „Esc“. Tato klávesa souží i jako bezpečnostní tlačítko, které okamžitě kdykoli

vypne probíhající kód a přepne režim letu dronu do přistávacího módu, ve kterém dron samostatně přistane pomocí GPS.

3.3 Realizace detekce člověka

Při realizaci tohoto systému byla pro detekci použita YOLO neuronová síť. Pro tuto aplikaci byla nejvhodnější, ale bylo zapotřebí malých úprav na vytvořeném počítačovém kódu a také zvýšit výpočetní výkon. Mikropočítač Raspberry Pi byl nahrazen klasickým počítačem, komunikaci zajišťuje 2,4 GHz telemetrie a přenos videa do počítače zajišťují 5,8 GHz analogové antény. I když je obraz přenášen analogově, tedy nemá vysokou kvalitu, je neuronová síť schopna detekovat požadovaný objekt. Program v jazyce Python je zde spuštěný na počítači a neuronová síť YOLO se aplikuje podobně jako Haarova kaskáda, s tím rozdílem, že výpočet neuronové sítě se provádí na grafické kartě GPU. YOLO neuronové sítě se využívá přes knihovnu Darknet. Do knihovny lze nahrát jakoukoli verzi YOLO. Při testech jsem využíval oficiální verzi yolo-v3. Řízení dronu je stejné jako při detekce QR kódů, takže původní program byl opatřen jen o několik změn [39].

3.3.1 Změny v původním kódu pro zajištění detekce pomocí YOLO

První změnou bylo odstranění použití Haarovy kaskády a nahrazení příkazem pro knihovnu Darknet. Původní program knihovny jsem upravil za účelem získání velikosti a místě středu detekovaného objektu. Tyto informace poté knihovna předá do vytvořeného řídícího programu. Úprava programu v knihovně Darknet zajišťující zjištění pozice i velikost objektu v zorném poli kamery je vyobrazena na obrázku 3.9.

```
76  def bbox2points(bbox):
77      x, y, w, h = bbox
78      xmin = int(round(x - (w / 2)))
79      xmax = int(round(x + (w / 2)))
80      ymin = int(round(y - (h / 2)))
81      ymax = int(round(y + (h / 2)))
82      return xmin, ymin, xmax, ymax
83
84
85  def middle(detections, dy,dx):
86      import cv2
87      en=w=0
88      for label, confidence, bbox in detections:
89          if label == "person":
90              x, y, w, h = bbox
91              dy = y
92              dx = x
93              en=1
94      return dy,dx,en,w
```

Obrázek 3.9 Úprava programu knihovny Darknet

Hlavní úpravou řídícího kódu je získání informace, zda se v zorném poli nachází člověk, tedy použití YOLO neuronové sítě skrze Darknet knihovnu. A dále získání detekce i s velikostmi detekovaného člověka a jeho umístění v pixelech. Práce s těmito informacemi probíhá stejně jako u detekce QR kódů. Je zjištěna výchylka od středu kamery a vytvořeno stejné autonomní korekční řízení, zajišťující následování člověka. Využití knihovny Darknet a získání souřadnic cíle je na obrázku 3.10.

```
115  if frame_resized is not None:
116      image = darknet.draw_boxes(detections, frame_resized, class_colors)
117      image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
118      ax=dx=dy=en=w=0
119      directionx=directiony=""
120      dy,dx,en,w = darknet.middle(detections,dy,dx)
```

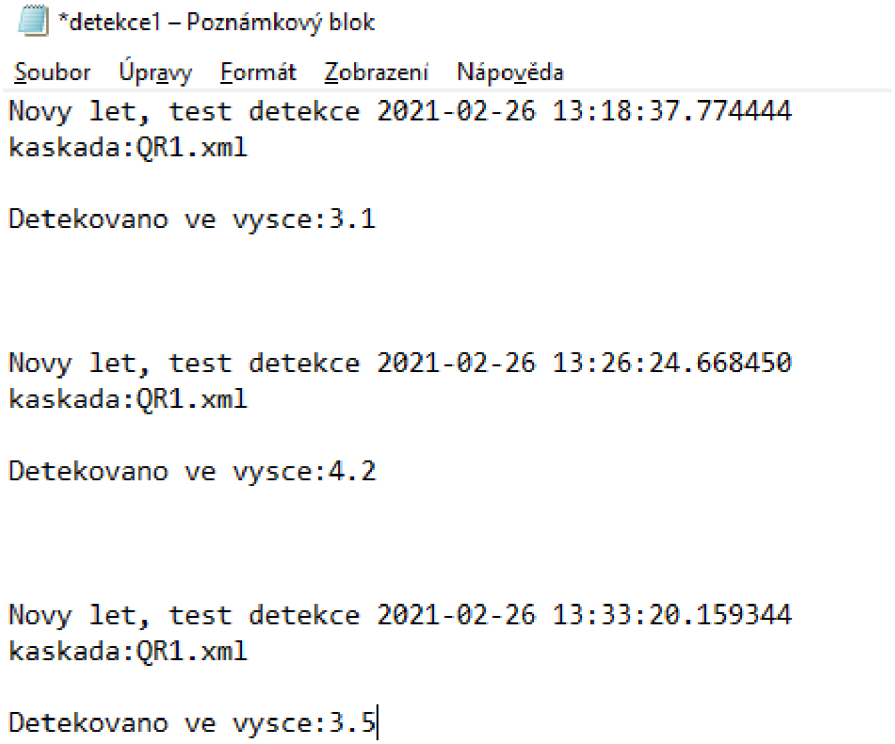
Obrázek 3.10 Využívání knihovny Darknet pro získání detekce, velikosti a umístění cíle v zorném poli kamery

4. VYHODNOCENÍ ÚSPĚŠNOSTI DETEKCE OBJEKTŮ

U každé vytvořené detekce jsou provedeny experimenty pro vyhodnocení detekce. Každý z experimentů probíhal za slunečného dne a plně pod kontrolou. Spuštěné programy obsahují pojistky pro okamžité vypnutí programu po stisku určitého tlačítka, a také byl dron stále připojen na vysílačku. Z vysílačky se v případě nouze mohl vyslat signál pro okamžitou manuální kontrolu nad dronem a vyrušení signálů z pozemní jednotky. Pro vyhodnocení detekce statického objektu jsem použil Haarovy kaskády a pro objekt pohybující jsem použil YOLO neuronové sítě. Haarovy kaskády by sledování člověka vzhledem k přesnosti detekce nebyli schopné, a tak jsou použity pro detekci jen stacionárního cíle, kde detekce je jednoduší a není potřeba složitějších výpočtů, a tedy ani YOLO neuronové sítě.

4.1 Vyhodnocení detekce Haarových kaskád

Pro vyhodnocení detekce Haarových kaskád jsem vytvořil tři různé kaskády pro QR kódy o modulech 21 x 21, 29 x 29 a 33 x 33. Cílem experimentů bylo zjistit, v jaké výšce mohou Haarovy kaskády detekovat QR kód o různých modulech, nikoli dekódovat informaci v nich obsaženou. Detekce probíhaly na Raspberry Pi 3B. Mikropočítač byl umístěn přímo na dron a spojen sériovou linkou s letovou jednotkou pro komunikaci. Jako kamera byla použita USB kamera Logitech 720p. Pro kontrolu správné detekce se obraz přenášel pomocí aplikace AnyDesk do mobilu, kde pilot kontroloval správnost detekce. Sice systém nevykazuje vysokou přesnost a spolehlivost, ale při vytvoření správné předlohy a zajištění určitých podmínek, je pro test detekce ideální. Výsledná nejvyšší výška detekovaného QR kódu, spolu s časem a typem použité kaskády, byla zaznamenána automaticky do textového souboru. Příklad výsledů jednotlivých experimentů zapsaných do textového souboru ilustruje obrázek 4.1.

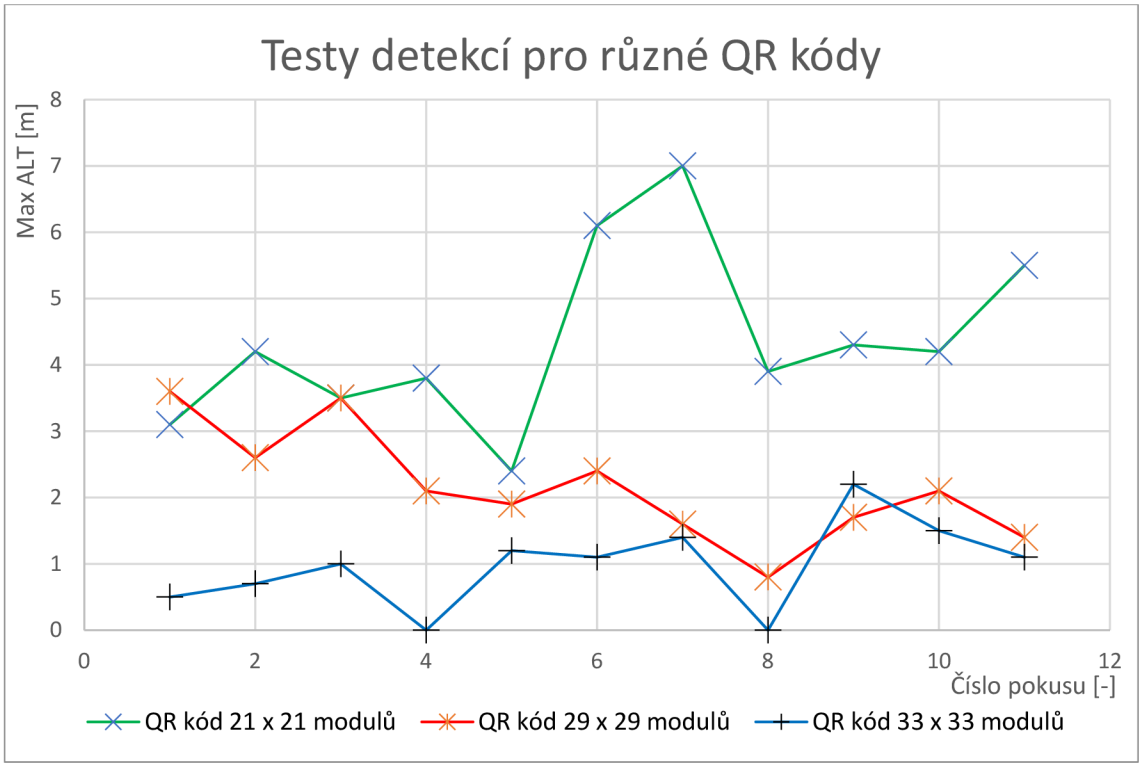


Obrázek 4.1 Příklad zápisu výsledné detekce do textového souboru

Z výsledků vypsaných do textového souboru, byla vytvořena tabulka 4.1 popisující úspěšné detekce a jejich maximální výšku. Z této tabulky byl vytvořen graf, který znázorňuje úspěšnost experimentů vyobrazený na obrázek 4.2.

Tabulka 4.1 Tabulka výsledných hodnot detekce QR kódů

Detekce: QR kód 21 x 21 modulů		Detekce: QR kód 29 x 29 modulů		Detekce: QR kód 33 x 33 modulů	
číslo pokusu	Maximální výška detekce [m]	číslo pokusu	Maximální výška detekce [m]	číslo pokusu	Maximální výška detekce [m]
1	3,1	1	3,6	1	0,5
2	4,2	2	2,6	2	0,7
3	3,5	3	3,5	3	1
4	3,8	4	2,1	4	0
5	2,4	5	1,9	5	1,2
6	6,1	6	2,4	6	1,1
7	7,0	7	1,6	7	1,4
8	3,9	8	0,8	8	0
9	4,3	9	1,7	9	2,2
10	4,2	10	2,1	10	1,5
11	5,5	11	1,4	11	1,1
Průměr	4,4	Průměr	2,2	Průměr	0,93



Obrázek 4.2 Graf výsledků detekce pro QR s rozdílným počtem modulů

Výsledkem je tedy průměrná výška detekce pro každý z QR kódů. Nejúspěšnější detekcí (průměrně ve 4,4 m) je detekce QR kódu s nejméně moduly, a nejméně úspěšná detekce (průměrně ve 0,93 m) je detekce QR kódu s nejvíce moduly. Je to dáno jemností černých a bílých ploch. Kontrast jednotlivých ploch při použití QR kódu s moduly 21 x 21 je větší než u použití více modulů. Detekce je tedy přesnější a výsledkem je, že neuronová síť zvládne rozpoznat tvar kódu na větší vzdálenosti.

Během experimentů docházelo k chybné detekci Haarovy kaskády, nebo k žádnému označení objektu. Hlavním problémem byla kamera, která nedokázala kompenzovat odražené sluneční světlo z papíru s QR kódem do objektivu. Vlivem toho byl obraz přesvícený a ve vyšších výškách (často i nad 4 m, záleží na intenzitě svitu) QR kód již nebyl rozpoznatelný. Kaskáda také označovala místa mimo oblast předpokládané detekce. Toto je dáno velmi rozmanitým pozadím jako je tráva a listí. Tato chyba by se dala kompenzovat vytvořením kaskády s negativními obrázky aktuálního pozadí. Příkladem správně vyhodnocené detekce je obrázek 4.3, příkladem chybné detekce je obrázek 4.4, příkladem chybné detekce vlivem přesvícení obrazu je obrázek 4.5.

Obrázek 4.3 Příklad správné vyhodnocení detekce QR kódu

Obrázek 4.4 Příklad chybné detekce QR kódu

Obrázek 4.5 Příklad přesvíceného obrazu kamery znemožňující detekci

4.2 Vyhodnocení detekce a autonomního řízení pomocí YOLO neuronové sítě

Během letů a testování detekce neuronové sítě byl viditelný rozdíl oproti Haarovým kaskádám. Pro všechny testy byl nastaven jako detekovaný objekt člověk. Neuronové sítě člověka ihned rozpoznaly a to z jakéhokoli úhlu a natočení. Během testů se dron pohyboval do výšky 10 m (tato hodnota byla nastavena jako provozní výška) a výsledkem byla přesná a okamžitá detekce člověka. Příklad detekce člověka během jednoho z letů je na obrázku 4.6.

Obrázek 4.6 Detekce člověka za letu pomocí YOLO neuronové sítě

Protože detekce v provozní výšce byla vždy úspěšná, byla vyhodnocena i funkčnost autonomního řízení. Pro toto vyhodnocení bylo cílem následování člověka. Letové instrukce, které dostává letová jednotka od pozemní stanice byli omezeny jen na let vpřed, vzad a otočení dronu vlevo a vpravo. Tyto instrukce, spolu s definovanými rychlostmi otáčení a rychlosti letu jsou vypočítávány na základě pozice cíle (člověka) od kamery, umístěné na dronu. Systém je tedy plně autonomní a vytváří tak řízení dronu jen pomocí kamery. Obrázek 4.7 zobrazuje jeden z letů, kde je zachycen dron, který následuje člověka před ním. Na obrázku 4.6 je také vidět v levém horním rohu souřadnice polohy člověka v zorném poli kamery a instrukce „vpravo“ a „vpřed“ indukující požadovaný směr letu dronu.

Obrázek 4.7 Test autonomního řízení dronu

Řízení má prozatím své omezení, jako je rychlost dronu (3 m/s), kdy se dron nestihne otočit, nebo cíl je příliš rychlý, nastane ztráta cíle v zorném poli a dron se zastaví. Poté drží výšku a čeká buď na instrukce, nebo na zachycení nového cíle. Aby se cíl neztratil ze zorného pole je nutné zvýšit rychlost otáčení dronu a jeho rychlost letět vpřed a vzad. Maximální hodnoty rychlostí jsou však omezeny stavem baterie, povětrnostními podmínkami a celkovou konstrukcí dronu, jako je počet ramen a výkon motorů. Řízení je plynulé při pohybu cíle vycházkovou chůzí a dosahuje okamžité odezvy na vychýlení cíle ze středu zorného pole. Pro jeho plynulost, byla vytvořena hysterezní smyčka řízení, jako definovaná tolerance mezi středem zorného pole a středem cíle. Ovládání je uzpůsobené pro let v definované vzdálenosti od člověka (okolo 3 m), když se tedy člověk přiblíží k dronu na vzdálenost menší, dron začne letět vzad. Toto řízení je vytvořeno měřením šířky a výšky detekovaného objektu. Obrázek 4.8 ilustruje případ, kdy cíl je v požadované vzdálenosti od dronu a není potřeba letu vpřed.

Obrázek 4.8 Test autonomního řízení dronu v definované vzdálenosti od cíle

Test tohoto řízení byl zaznamenán jako video ukazující funkčnost celého autonomního systému. Rychlost dronu vpřed a vzad bylo nastaveno na 0,7 m/s a rychlost otáčení na 5 °/s.

5. ZÁVĚR

Během bakalářské práce byly vytvořeny Haarovy kaskády pro detekci QR kódů jako stacionárního cíle. Předpokladem je využívání kaskády při přesném přistávání a detekci jednoduchých stacionárních cílů, jako jsou například QR kódy, kde pro detekci stačí kaskády a nemusí se využívat složité neuronové sítě jako je YOLO. Principem bylo zajištění přesného přistávání nebo přijímání informací z prostředí za pomocí kamery. Přesnost a spolehlivost Haarovy kaskády však nebyla pro autonomní řízení pomocí QR kódů dostatečná. Úspěšné detekování QR kódu proběhlo maximálně v 7 m nad objektem. Obraz byl zpracováván přímo na dronu pomocí Raspberry Pi. Výsledkem testovacích letů, které detekovaly QR kódy bylo, že moduly 29 x 29 nejsou pro Haarovy kaskády spolehlivě rozeznatelné. Chybovost detekce způsobovala nerozpoznání kódu a detekci chybného objektu, jako trs trávy nebo rozmanité pozadí na obraze. Použití Haarovy kaskády však může být využito pro detekci jednoduchých objektů jako jsou heliporty, či velké výrazné geometrické obrazce. Vytvořením kaskády na velmi členitý a složitý obrazec je nejen složité, ale i nespolehlivé v praxi. Důvodem je rozmanitost pozadí objektu a potenciální chybná detekce.

Pro vytvoření autonomního řízení byl naprogramován skript využívající neuronové sítě YOLO, které jsou velmi rozšířené a obsahují desítky možných detekcí různých objektů. Předpokladem pro použití těchto neuronových sítí je schopnost detekce cíle i za pohybu a mírné změny obrysu objektu, například stojící člověk a člověk v pohybu. V těchto dynamických momentech je složitost neuronových sítí YOLO výhodou, protože zvládnou detekovat požadovaný cíl s velkou přesností. Výsledná detekce v provozní výšce (do 10 m) byla natolik přesná, že detekovala cíl (člověka) z jakéhokoli úhlu. Testováním proto nepodléhala přímo detekce jako samotná, ale vyhodnocení autonomního letu za pomocí této neuronové sítě. Vytvořený dron odesílá obraz do pozemní stanice, kde je zpracováván a na základě pozice a velikosti detekovaného objektu je vytvořeno autonomní řízení vysílané do letové jednotky. Pro testování tohoto vytvořeného řízen bylo nastaveno sledování člověka. Pokud se v zorném poli objevil člověk, neuronová síť ho ihned detekovala. Informace o velikosti a souřadnice jeho pozice v pixelech předala řídícímu programu, který vyhodnotil, kam má dron letět pro dosažení optimálního natočení a vzdálenosti od cíle. Celý systém funguje plynule za předpokladu příznivého počasí, a že se cíl nebude pohybovat rychleji než dron. Omezení rychlosti dronu je způsobeno jak konstrukcí, tak i aktuálním stavem komponentů. Pro testovací lety byla nastavena rychlost otáčení a letu dronu na rychlost odpovídající rychlosti chůze člověka. Systém tak dokáže následovat člověka v určité vzdálenosti plynule.

Dalším postupem této práce může být selektivita detekovaných objektů. V této fázi systém nerozpozná různé osoby, takže řízení může být nestabilní při detekování více cílů najednou. Jako selektivní řešení bych zvolil označení a definovaní jednotlivých cílů. Poté

by problém při detekci více cílů mohl být řešen jednotlivě pro každého člověka a zařídit tak sledování specifického cíle.

LITERATURA

[1] Drone technology uses and applications for commercial, industrial and military drones in 2021 and the future. Businessinsider [online]. 2021, Jan 12, 2021 [cit. 2021-5-23]. Dostupné z: https://www.businessinsider.com/drone-technology-uses-applications

[2] Digitalizace a zpracování obrazu [online]. [cit. 2020-10-18]. Dostupné z: http://home.zcu.cz/~holota5/publ/DigZprO.pdf

[3] Cascade Classifier [online]. [cit. 2020-10-18]. Dostupné z: https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html

[4] What's the Difference Between Haar-Feature Classifiers and Convolutional Neural Networks? [online]. 2018 [cit. 2020-10-18]. Dostupné z: https://towardsdatascience.com/whats-the-difference-between-haar-feature-classifiershttps://towardsdatascience.com/whats-the-difference-between-haar-feature-classifiers-and-convolutional-neural-networks-ce6828343aeband-convolutional-neural-networks-ce6828343aeb

[5] IMAGE PROCESSING FOR UAV AUTONOMOUS NAVIGATION [online]. Brno, 2020 [cit. 2020-10-18]. Dostupné z: https://conf.feec.vutbr.cz/eeict/EEICT2020/author/downloadFile/689/1471/1

[6] Detekce obličejů pomocí příznakového rozpoznání [online]. 2014 [cit. 2020-10-18]. Dostupné z: https://dspace.vsb.cz/bitstream/handle/10084/106109/MER0035_FEI_B2647_26R0 25_2014.pdf?sequence=1&isAllowed=y

[7] Face Detection Using OpenCV With Haar Cascade Classifiers [online]. 2019 [cit. 2020-10-23]. Dostupné z: https://becominghuman.ai/face-detection-using-opencv-with-haarhttps://becominghuman.ai/face-detection-using-opencv-with-haar-cascade-classifiers-941dbb25177cascade-classifiers-941dbb25177

[8] Haar Cascades vs Mask R-CNN [online]. 2018 [cit. 2020-10-18]. Dostupné z: https://www.reddit.com/r/learnmachinelearning/comments/8d0puz/haar_cascades_vs_ mask_rcnn/

[9] R-CNN, Fast R-CNN, Faster R-CNN, YOLO - Object Detection Algorithms [online]. 2018 [cit. 2020-10-18]. Dostupné z: https://towardsdatascience.com/r-cnn-fast-r-cnnhttps://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365efaster-r-cnn-yolo-object-detection-algorithms-36d53571365e

[10] YOLO object detection with OpenCV [online]. 2018 [cit. 2020-10-18]. Dostupné z: https://www.pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv/

[11] Object Detection Using YOLOv3 [online]. 2020 [cit. 2020-10-18]. Dostupné z: https://medium.com/@larkspurvc718/object-detection-using-yolov3-f7c75515ddc

[12] You Only Look Once: Unified, Real-Time Object Detection [online]. 2019 [cit. 2020-10-23]. Dostupné z: https://arxiv.org/pdf/1506.02640.pdf
[13] About [online]. [cit. 2020-10-18]. Dostupné z: https://opencv.org/about/
[14] OpenCV – Overview [online]. 2019 [cit. 2020-10-18]. Dostupné z: https://www.geeksforgeeks.org/opencv-overview/
[15] Co je to QR kód. QRcodes [online]. [cit. 2021-5-9]. Dostupné z: http://qrcodes.cz/qrkody-qrcode.php
[16] QR Code Size Setting Instruction. QRstuff [online]. 2011 [cit. 2021-5-9]. Dostupné z: https://blog.qrstuff.com/2011/11/23/qr-code-minimum-size
[17] Úvod do telemetrie [online]. [cit. 2020-10-18]. Dostupné z: https://www.rc-zoom.cz/uvod-do-telemetrie/zoom.cz/uvod-do-telemetrie/
[18] MAVLink Developer Guide [online]. [cit. 2020-10-23]. Dostupné z: https://mavlink.io/en/
[19] Protocol Overview [online]. [cit. 2020-10-23]. Dostupné z: https://mavlink.io/en/about/overview.html
[20] Autopilot Hardware Options. Ardupilot [online]. [cit. 2021-5-16]. Dostupné z: https://ardupilot.org/copter/docs/common-autopilots.html
[21] About. Ardupilot [online]. [cit. 2021-5-16]. Dostupné z: https://ardupilot.org/index.php/about
[22] SiK Telemetry Radio [online]. [cit. 2020-10-23]. Dostupné z: https://ardupilot.org/copter/docs/common-sik-telemetry-radio.html
[23] DragonLink RC Systems [online]. [cit. 2020-10-23]. Dostupné z: https://ardupilot.org/copter/docs/common-dragonlink-rc.html
[24] VR TELEMETRY 868 MHZ. Lasernavigation [online]. [cit. 2021-5-16]. Dostupné z: https://www.lasernavigation.it/vr-telemetry/
[25] [online]. 2019 [cit. 2020-10-23]. Dostupné z: https://www.banggood.com/Newhttps://www.banggood.com/New-Upgraded-V2_0-3DR-Radio-Telemetry-433MHZ-915MHZ-Data-Transmission-Module-For-Android-Smartphone-APM-Pixhawk-PX4-p-1304758.html?ID=510651&cur_warehouse=CNUpgraded-V2_0-3DR-Radio-Telemetry-433MHZ-915MHZ-Data-Transmission-Module-For-Android-Smartphone-APM-Pixhawk-PX4-p-1304758.html?ID=510651&cur_warehouse=CN
[26] RFD 868x Modem. Rfdesign [online]. [cit. 2021-5-16]. Dostupné z: http://store.rfdesign.com.au/rfd-868x-modem/
[27] ESP8266 WiFi Module. Px4 User guide [online]. 2021 [cit. 2021-5-16]. Dostupné z: https://docs.px4.io/master/en/telemetry/esp8266_wifi_module.html
[28] ESP8266 wifi telemetry [online]. [cit. 2020-10-23]. Dostupné z: https://ardupilot.org/copter/docs/common-esp8266-telemetry.html
[29] Bluetooth Telemetry radio [online]. [cit. 2020-10-23]. Dostupné z: https://ardupilot.org/copter/docs/common-mission-planner-

bluetoothhttps://ardupilot.org/copter/docs/common-mission-planner-bluetooth-connectivity.htmlconnectivity.html
[30] BT06 Bluetooth Serial Port Wireless Data ModuleMk2 [online]. [cit. 2020-10-23]. Dostupné z: https://sunhokey.cn/collections/bluetooth-module/products/bt06https://sunhokey.cn/collections/bluetooth-module/products/bt06-bluetooth-serial-port-wireless-data-modulebluetooth-serial-port-wireless-data-module
[31] Communicating with Raspberry Pi via MAVLink. Ardupilot [online]. [cit. 2021-5-16]. Dostupné z: https://ardupilot.org/dev/docs/raspberry-pi-via-mavlink.html
[32] Telemetry (landing page). Ardupilot [online]. [cit. 2021-5-16]. Dostupné z: https://ardupilot.org/copter/docs/common-telemetry-landingpage.html
[33] RockBLOCK Mk2 [online]. [cit. 2020-10-23]. Dostupné z: https://www.rock7.com/products/rockblock-iridium-9602-satellite-modem
[34] FrSky Telemetry. Ardupilot [online]. [cit. 2021-5-16]. Dostupné z: https://ardupilot.org/copter/docs/common-frsky-telemetry.html
[35] UAVcast 3G/4G Cellular Telemetry. Ardupilot [online]. [cit. 2021-5-16]. Dostupné z: https://ardupilot.org/copter/docs/common-uavcast-telemetry.html
[36] Using commercial 5G technology for telemetry in flight testing. Aerospacetestinginternational [online]. 2019 [cit. 2021-5-16]. Dostupné z: https://www.aerospacetestinginternational.com/features/commercial-5g-technology-used-for-telemetry-in-flight-tests.html
[37] Raspberry Pi 3 Model B 64-bit 1GB RAM. Rpishop [online]. [cit. 2021-5-16]. Dostupné z: https://rpishop.cz/raspberry-pi-3b/283-raspberry-pi-3-model-b-64-bit-5060214370028.html
[38] Make your own Haar Cascade on Windows | Quick & Simple | Face Detection. Rpishop [online]. 2016 [cit. 2021-5-16]. Dostupné z: https://www.youtube.com/watch?v=Dg-4MoABv4I
[39] Darknet. GitHub [online]. 2020 [cit. 2021-5-25]. Dostupné z: https://github.com/AlexeyAB/darknet

SEZNAM SYMBOLŮ A ZKRATEK

Zkratky:

GPS	Global Positioning System
QR	Quick Response
Wi-Fi	wireless fidelity
GPU	Graphics processing unit
CUDA	Compute Unified Device Architecture
R-CNN	Region-based Convolutional Neural Networks
CNN	Convolutional Neural Networks
SVM	Support vector machines
RPN	Regionální návrhová síť
YOLO	You look only once
MAVLink	Micro Air Vehicle Link
USB	Universal Serial Bus
ID	Identifikace ve výpočetní technice.
ISO/OSI	Mezinárodní normalizace Propojení otevřených systémů
IPv4	Internet Protocol version 4
BD_ADDR	BlueTooth Device Address
TCP/IP	Transmission Control Protocol/Internet Protocol
UAV	Unmanned Aerial Vehicle
IMU	internal measurement units
PID	Proporcionální, integrační a derivační regulátor
PWM	Pulse Width Modulation
QR	Quick Response

Veličiny:

Dpi	Dots per inch
Mbit/s	Megabit za sekundu
MHZ	Megahertz
Km	Kilometr
m	Metr
mm	Milimetr
mW	Miliwatt
GHz	Gigaherzt
m/s	Metrů za sekundu
°/s	Stupně za sekundu