

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

VIZUÁLNY SYSTÉM PRE INTERAKCIU
ĽUDSKÉHO UČITEĽA S ROBOTOM
DIPLOMOVÁ PRÁCA

2021
ANGELIKA PIRHALOVÁ

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

VIZUÁLNY SYSTÉM PRE INTERAKCIU
ĽUDSKÉHO UČITEĽA S ROBOTOM
DIPLOMOVÁ PRÁCA

Študijný program: Aplikovaná informatika
Študijný odbor: Aplikovaná informatika
Školiace pracovisko: Katedra aplikovanej informatiky
Školiteľ: Ing. Viktor Kocur, PhD.

Bratislava, 2021
Angelika Pirhalová



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Angelika Fedáková
Študijný program: aplikovaná informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Vizualný systém pre interakciu ľudského učiteľa s robotom
Visual system for interaction of a human teacher with a robot

Anotácia: Toto zadanie je súčasťou projektu interakcie ľudského učiteľa s robotom. Robot pri tejto interakcii manipuluje jednoduchými objektmi na základe pokynov od ľudského učiteľa. Pre tento účel je tak vhodné aby robot dokázal správne detegovať pozíciu jednoduchých, objektov pomocou, učiteľovej ruky a svojho robotického ramena pomocou svojej RGB-D kamery.

Cieľ: Cieľom tejto práce je navrhnúť, implementovať a otestovať systém ktorý na základe vstupných RGB-D dat z Intel RealSense kamery deteguje pozíciu jednoduchých objektov, učiteľovej ruky a robotického ramena. Súčasťou práce bude prehľad existujúcich riešení detekcie objektov v RGB-D snímkach. Navrhnutý algoritmus bude vyhodnotený v kontexte prebiehajúceho projektu interakcie ľudského učiteľa s robotom.

Vedúci: Ing. Viktor Kocur
Katedra: FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry: prof. Ing. Igor Farkaš, Dr.
Dátum zadania: 07.10.2020

Dátum schválenia: 08.10.2020

prof. RNDr. Roman Ďurikovič, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Pod'akovanie: Ďakujem všetkým, ktorí ma pri tvorbe tejto práce podporovali.

Abstrakt

Robot potrebuje na interakciu s reálnym svetom rozpoznať prostredie v ktorom sa nachádza a jednotlivé objekty v ňom. Cieľom našej práce bola 3D objektová detekcia drevených stavebnicových kociek z RGB obrazu kamery robota NICO. Po návrhu využitia spôsobu generovania syntetických dát, sme vygenerovali rôzne datasety, obsahujúce rôzne typy stavebnicových kociek a rôzne počty ich inštancií. Následne sme na týchto dátach natrénovali konvolučnú neurónovú sieť a zhodnotili jej fungovanie na syntetických a reálnych dátach. TODO: doplniť ako dopadli výsledky.

Kľúčové slová: objektová detekcia, generovanie syntetických dát, konvolučné neurónové siete

Abstract

To interact with real world, robot needs to know the environments he is in. He needs to know positions and orientations of objects of his surroundings. Goal of our thesis was 3D object detection of wooden blocks in RGB camera images of robot NICO. After proposal to use synthetic data generation, we generated multiple datasets consisting of different objects and number of their instances. After that we trained a convolutional neural network on these datasets and evaluated it's outputs on synthetic and real data. TODO: add results commentary.

Keywords: object detection, generation of synthetic data, convolutional neural network

Obsah

Úvod	1
1 Úvod do problematiky	2
1.1 Počítačové videnie	2
1.2 Obrazové príznaky	3
1.3 Objektová detekcia	3
1.3.1 Tradičné metódy objektovej detekcie	4
1.3.2 Konvolučné neurónové siete	4
2 Popis problému a stav problematiky	9
2.1 Objektová detekcia v priestore	9
2.2 Špecifikácia zadania našej práce	10
2.3 Vstupy pre 3D objektovú detekciu	10
2.4 Dataset pre tréning CNN	11
2.5 Syntetické dáta	12
2.5.1 Fotorealistické dáta	12
2.5.2 Randomizácia prostredia	13
2.6 Existujúce riešenia	14
2.6.1 Určovanie 3D pozície centroidov objektov pomocou CNN	14
2.6.2 Tradičné metódy pre určenie 6DOF pózy drevených stavebníc	15
2.6.3 Deep object pose estimation - DOPE	15
3 Návrh riešenia problému	17
3.1 Generovanie dát	17
3.2 Využitie randomizácie prostredia a MuJoCo Physics Engine	17
3.3 Využitie NViSII generácie dát	18
3.4 Výber neurónovej siete	19
4 Výsledky experimentov	20
4.1 Dataset jedného obrazu s jednou kockou	20
Záver	22

Zoznam obrázkov

1.1	Reprezentácia globálnych a lokálnych príznakov	3
1.2	Konvolúcia s obrazu 5×5 s nulovým paddingom a s filtrom rozmeru 3×3 s použitím stride jedna.	6
1.3	Obraz rozmerov 3×3 s paddingom jedna	6
1.4	Rôzne aktivačné funkcie	7
1.5	Maxpooling na obraze s veľkosťou 2×2 , veľkosťou okna 2×2 a stride 2	7
2.1	Stavebnicové kocky záujmu našej práce	10
2.2	Ukážka robota NICO nachádzajúceho sa na našej fakulte	10
2.3	Anotovaný obraz	11
2.4	Fotorealistické synteticky vygenerované dáta	13
2.5	Dáta generované princípom randomizácie prostredia	14
2.6	Objekty záujmu pri metóde hľadajúca pozíciu centroidov	15
2.7	Príklad výsledku využitia tradičných metód na reálnych dátach	16
2.8	Príklad výsledku využitia DOPE na reálnych dátach	16
3.1	Príklady vygenerovaných dát s použitím MuJoCo Physics Engine	18
3.2	Príklady vygenerovaných dát s použitím programu založenom na NVISII	19
4.1	Obraz datasetu veľkosti jeden, na ktorom bola sieť trébovaná	20
4.2	Výstup siete pre obraz z datasetu veľkosti jeden, na ktorom bola sieť trébovaná	21

Zoznam tabuliek

Úvod

S automatickými alebo poloaautomatickými robotickými prístrojmi sa v dnešnej dobe stretávame čoraz častejšie. Na to, aby na reálny svet roboty dokázali reagovať a komunikovať s ním je často potrebné, aby dokázali rozpoznávať, čo sa okolo nich deje prostredníctvom senzorov, napríklad kamier. V tejto práci sme sa venovali rozpoznávaniu 3D pozície drevených kociek v RGB obraze kamery robota NICO.

Oblasť počítačového videnia je rozsiahle skúmaná oblasť. Je to najmä vďaka jej širokému využitiu v riešení problémov reálneho sveta. Jednoduché aplikácie počítačového videnia môžeme nájsť kdekoľvek, od priemyselných kamier až po mobilné telefóny. Oblasť objektovej detekcie je jednou z úloh počítačového videnia. Je to proces, kedy sa v obraze snažíme nájsť a určiť pozíciu požadovaných objektov. Na túto úlohu môžeme použiť tradičné metódy počítačového videnia, ako sú napríklad hranové detektory. Druhým prístupom je využiť princípy hlbokého učenia a takzvané konvolučné neurónové siete. Tieto siete majú ale jeden základný problém. Na ich využitie je potrebné mať k dispozícii čo najväčšie množstvo anotovaných dát. Ručný zber dát a ich anotácia je časovo náročná a pri väčších datasetoch a obmedzených zdrojoch môže byť nesplniteľná. Z toho dôvodu sme sa v tejto diplomovej práci venovali testovaniu procesu, v ktorom sme na tréning konvulčnej neurónovej siete pre detekciu drevených kociek využili iba počítačovo generované obrazy.

V kapitole 1 popíšeme základné pojmy a algoritmy v oblasti počítačového videnia a objektovej detekcie. V kapitole 2 následne špecifikujeme problém a rozoberieme existujúce riešenia tohto problému. V kapitole 3 sa venujeme návrhu a špecifikácii nášho riešenia. V kapitole 4 popíšeme a zhodnotíme výstupy nášho postupu na syntetických, ale aj reálnych dátach.

Kapitola 1

Úvod do problematiky

V tejto kapitole zavedieme základné pojmy týkajúce sa počítačového videnia a konvolučných neurónových sietí, potrebné pre pochopenie tejto práce. Hlavná oblasť, ktorej sa týka naša práca, je 3D objektová detekcia v RGB obraze. Preto si popíšeme niektoré tradičné metódy objektovej detekcie a taktiež fungovanie konvolučných neurónových sietí a ich využitie pre riešenie problému 3D objektovej detekcie.

1.1 Počítačové videnie

Pre človeka je vnímanie 3D priestoru reálneho sveta a objektov v ňom prirodzené. Ľudia dokážu od seba rozlišovať jednotlivé objekty a taktiež ich farby, textúry, či veľkosti pomocou malých náznakov svetla, či textúr. Pochopenie ľudského vnímania sveta je veľmi zložitá a o jeho plné syntaktické a následne aj sémantické pochopenie sa snaží odvetvie s názvom počítačové videnie.

Ultimátnym cieľom počítačového videnia je z digitálneho obrazu získať plné sémantické pochopenie zobrazovanej reality. Aj keď sa technika a algoritmy v mnohých oblastiach, ako je napríklad detekcia objektov, či ich klasifikácia v posledných rokoch výrazne posunuli, plné sémantické pochopenie zostáva stále v nedohľadne. Problémom je najmä komplexita informácií, ktorú človek vníma pri sledovaní obrazu. Sú to často informácie vyplývajúce zo súvislostí, ktoré z obrazu nie sú vyčítateľné. [4]

Medzi základné problémy, ktorými sa zaoberá počítačové videnie, patria klasifikácia, detekcia, či segmentácia. Pri riešení týchto problémov sa z obrazu snažíme získať základné informácie potrebné pre pochopenie obsahu daného obrazu, teda aké objekty a kde sa nachádzajú. Pri klasifikácii určujeme triedu daného obrazu, alebo jeho časti. Na rozdiel od toho pri detekcii okrem triedy určujeme aj oblasť, kde sa inštancie daných objektov nachádzajú. Pri segmentácii určujeme triedu každého pixela v obraze. Všetky tieto problémy ale spája využitie obrazových príznakov pri ich riešení.



Obr. 1.1: Reprezentácia globálnych a lokálnych príznakov

1.2 Obrazové príznaky

Obrazové príznaky sú časti alebo vzory obrazov, ktoré napomáhajú identifikovať objekty sa v ňom nachádzajúce. Napríklad o štvorci vieme, že má štyri rohy. Keďže aj obdĺžnik má štyri rohy, tento príznak sám o sebe nie je dostatočným na správne rozlíšenie týchto dvoch objektov. Správny výber príznakov je preto esenciálny pri úspešnom riešení úloh počítačového videnia.

Medzi obrazové príznaky patria napríklad hrany, rohy alebo farebný histogram. Obrazové príznaky môžeme rozdeliť do dvoch hlavných skupín - globálne a lokálne. Medzi globálne príznaky patria napríklad príznaky farby a textúry. Tieto príznaky popisujú obraz ako celok a môžu byť interpretované ako špecifická vlastnosť obrazu zahŕňajúca všetky pixely. Sú zväčša popísané jedným vektorom a ukázali sa byť prospešné napríklad pri hľadaní podobných obrazov v datasete. Na rozdiel od toho lokálne príznaky popisujú jednotlivé zaujímavé oblasti obrazu. V obraze najprv detekujeme zaujímavé oblasti, ktoré by potenciálne mohli obsahovať zásadné informácie o objekte. Následne sa pre každú zo zaujímavých oblastí snažíme vytvoriť príznakový vektor, ktorý je invariantný. To znamená, že pre rovnaké časti obrazu bude vyzeráť daný vektor rovnako. Takýto vektor nazývame príznakový deskriptor. Lokálne príznaky sú oproti globálnym užitočné napríklad pri určovaní jednotlivých inštancií objektov v obraze.

Keďže sa objekty väčšinou nachádzajú v rôznych prostrediach, či svetelných podmienkach, je potrebné, aby boli príznaky čo najrobustnejšie voči zmene kontrastu, šumu, osvetleniu alebo voči rozmazaniu. [2]

1.3 Objektová detekcia

Problém hľadania inštancií objektov vybraných tried v obraze nazývame objektová detekcia. Jej výstup nám poskytuje základné informácie o rozložení objektov v obraze. Poznáme dva typy objektovej detekcie. Prvou je všeobecná objektová detekcia, pri ktorej sa snažíme rozpoznať všetky triedy objektov v obraze, a tým simulovať schopnosti ľudského videnia. Druhý typ úlohy je aplikačná detekcia objektov, kde je hľadanie

zúžené na inštalácie objektov konkrétnych, vopred určených tried. [9]

Poznáme mnoho prístupov riešenia tohto problému, ktoré ale môžeme rozdeliť do dvoch hlavných skupín. Prvou skupinou sú tradičné metódy počítačového videnia a druhou sú metódy založené na hlbokom učení. Zatiaľ čo sa tradičné metódy spoliehajú na výber správnych príznakov človekom, neurónové siete si vyberajú a extrahujú príznaky bez ľudského zásahu.

1.3.1 Tradičné metódy objektovej detekcie

Väčšina týchto prístupov je založených na ručne človekom vybraných príznakoch. Ich výhodou je, že pri trénovaní nepotrebujeme veľké množstvo dát. Určenie vhodných príznakov však môže byť zložité. Zatiaľ čo pri jednoduchých geometrických objektoch sú tieto metódy pomerne presné a veľmi rýchle, pri zložitejších objektoch, ako sú napríklad zvieratá je určiť vhodné príznaky náročné. Pozrime sa na dva prístupy, ktoré výrazne ovplyvnili históriu objektovej detekcie.

1. Viola Jones detektor - prvý detektor ľudských tvárí v reálnom čase, ktorý v svojej dobe rýchlosťou výrazne prekonal ostatné algoritmy. Svoju rýchlosť dosiahol najmä vďaka využitiu integrálnych obrazov, algoritmický výber Haárových filtrov a viackrokovej detekcii.
2. Histogram orientovaných gradientov - je príznakový deskriptor. V svojom období prispel najmä k riešeniu problému škálovej a tvarovej invariantnosti pri objektovej detekcii. Je založený na rozdelení obrazu do uniformnej mriežky. Každéj bunke tejto mriežky sú následne priradené počty nájdených gradientov v stanovených smeroch.

1.3.2 Konvolučné neurónové siete

Základom metód založených na hlbokom učení je výber a extrakcia obrazových príznakov bez zásahu človeka. Väčšina metód je založená na konvolučných neurónových sieťach (CNN). Problémy, ktoré tieto siete riešia s vysokou presnosťou, sú napríklad objektová detekcia, klasifikácia či segmentácia obrazu. Ich pozitívom je vysoká presnosť výsledkov a robustnosť, teda schopnosť adaptácie na rôznorodé dáta. Ich nevýhodou je potreba veľkého množstva dát pre učenie a taktiež čas potrebný na natrénovanie siete. Na kvalitné natrénovanie CNN sú často potrebné desaťtisíce, či státisíce obrazov a hodiny až dni času trénovania na kvalitných strojoch.

Vstupom konvolučných neurónových sietí je vo väčšine prípadov obraz, povedzme RGB obraz s rozmermi $X \times Y \times 3$. Výstup siete sa líši v závislosti od problému a implementácie riešenia. Pri klasifikačných problémoch to môže byť jedno číslo, určujúce triedu obrazu. Pri segmentácii to môže byť matica hodnôt, ktorá pre každý pixel

určuje triedu daného pixelu. Pri objektovej detekcii sa výstupy líšia v závislosti od implementácie. Zväčša sa ale skladajú z množiny obsahujúcej označenia tried a pozícií inštancií jednotlivých objektov. Veľkosť tejto množiny môže byť vopred pevne daná, alebo variabilná.

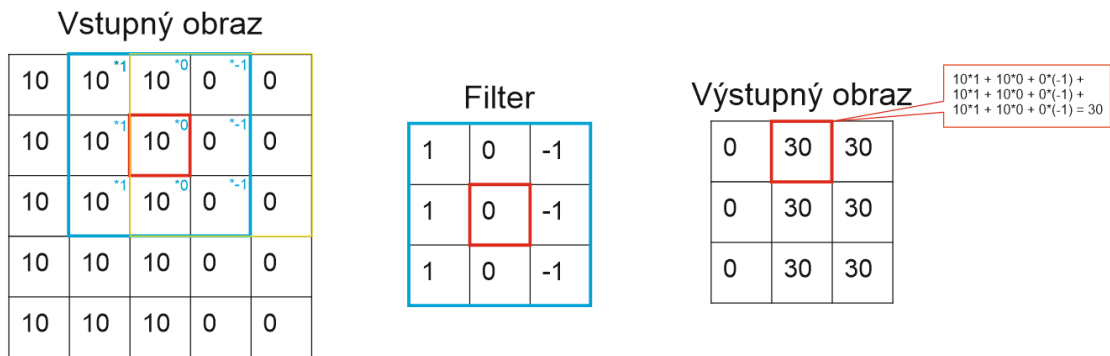
Konvolučné neurónové siete sa skladajú z viacerých vrstiev, a to: konvolučná vrstva, nelineárna vrstva, poolingová vrstva a plne prepojená vrstva. Jednotlivé vrstvy majú v sieti rozličné úlohy. Vrstvy s parametrami sú konvolučná a plne prepojená vrstva. Tieto parametre sieť v priebehu tréningu upravuje, a tak zlepšuje svoju schopnosť určiť správny výsledok.

V nasledujúcom odseku popíšeme fungovanie a význam jednotlivých typov vrstiev CNN.

- Konvolučná vrstva - táto vrstva nám výrazne pomáha pri znížení počtu parametrov siete. Pri plne prepojenej sieti by bol každý neurón siete prepojený s každým pixelom obrazu. Ak ale bude každý neurón pozeráť iba na istý výrez obrazu, výrazne tým znížime počet parametrov. Je to vďaka tomu, že sa každý neurón pozerá iba na časť obrazu, miesto celého. Tieto vrstvy sú založené na matematickej maticovej operácii s názvom konvolúcia.

Majme vstupný obraz rozmerov 9×9 a filter rozmerov 3×3 . Konvolúcia je proces, pri ktorom postupne priestorovo posúvame filter po obraze a rátame skalárny súčin filtra s časťou obrazu, ku ktorej je momentálne zarovnaný. Pri konvolúcii násobíme oblasť prislúchajúcu momentálnemu zarovnaniu filtra hodnotami filtra a súčiny spočítavame. Pre každé zarovnanie filtra teda získame jednu výslednú hodnotu. Filter, ktorý nazývame aj kernel, má zväčša nepárne rozmery. Často sú používané rozmery 3×3 alebo 5×5 . Je to vďaka tomu, že skalárny súčin takýchto filtrov vieme interpretovať ako zhustenie informácie o hodnotách v okne do jednej hodnoty. Na obrázku 1.2 máme príklad takejto konvolúcie. Filter budeme postupne prikladať k nášmu vstupnému obrazu, ako vidíme na obrázku modrou farbou. Následná pozícia filtra po posunutí je zobrazená žltou farbou. Z pôvodného obrazu nám vznikne zmenšený obraz rozmerov 3×3 . V prípade, že chceme zachovať informáciu o okrajových pixeloch, alebo chceme zachovať rozmer vstupu, pridávame okolo obrazu okraj, nazývaný padding, ktorý zväčša pozostáva z núl. Príklad paddingu veľkosti 1 vidíme na obraze 1.3. Tým sa stredový pixel filtra prikladá aj k okrajovým pixelom a keďže sa umelo zväčší rozmer vstupu, zachová sa tým aj rozmer výstupu. Taktiež je možné filter posúvať o viac ako jeden pixel. Posun, ktorý filter v každom kroku vykoná, sa nazýva stride. Pri tréňovaní sú parametrami jednotlivé filtre. Sieť sa teda učí hodnoty jednotlivých filtrov.

V prípade neurónových sietí máme často viacrozmerne vstupy, napr. pri RGB



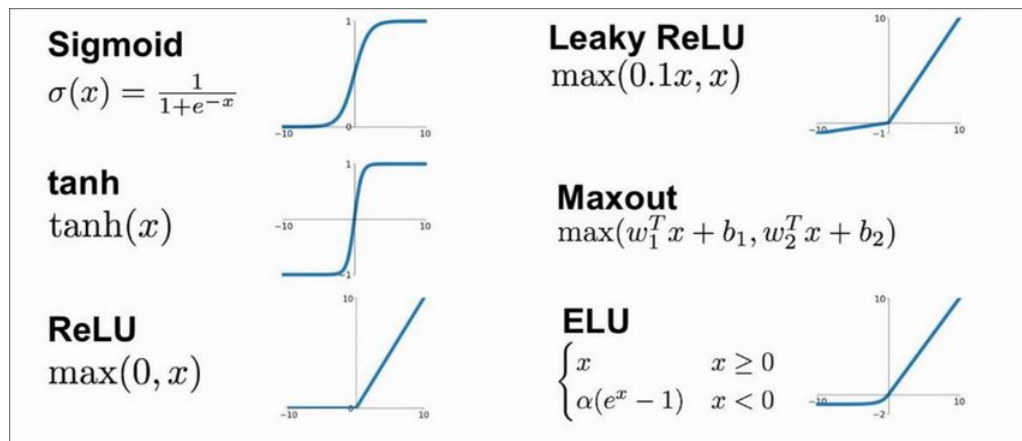
Obr. 1.2: Konvolúcia s obrazu 5×5 s nulovým paddingom a s filtrom rozmeru 3×3 s použitím stride jedna.



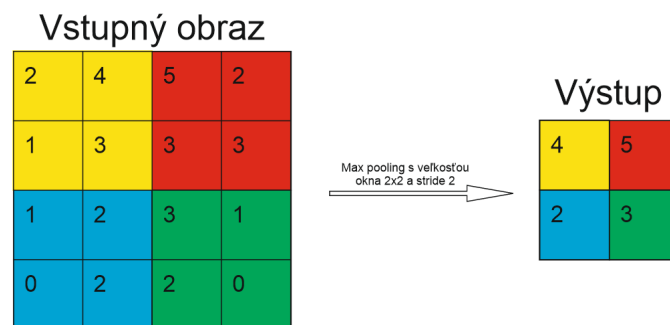
Obr. 1.3: Obraz rozmerov 3×3 s paddingom jedna

obrazoch sú vstupy trojrozmerné. V tomto prípade sa využíva priestorová konvolúcia, pri ktorej sa hĺbka obrazu zhoduje s hĺbkou konvolučného filtra. V tom prípade násobíme a sčítame hodnoty cez všetky dimenzie a výsledkom je teda obraz dvojrozmerný, dimenzia hĺbky sa stráca. Často sa však v jednej vrstve siete využíva viacero filtrov, ktorých výstupy keď poskladáme na seba, získame hĺbku obrazu veľkosti počtu filtrov.

- Nelineárna vrstva - keďže je neurónová sieť v svojej podstate množina lineárnych klasifikátorov poskladaných na seba, ich zloženie by sme dokázali definovať jedným iným lineárnym klasifikátorom, ktorý by bol ich kombináciou. Sieť by sa tak nedokázala učiť zložitejšie príznaky a klasifikovať lepšie ako obyčajný lineárny klasifikátor. Z tohto dôvodu pridávame do neurónovej siete nelineárnu vrstvu, ktorá obsahuje aktivačnú funkciu, ktorá do siete prináša nelinearitu. Poznáme rôzne aktivačné funkcie, ktoré majú svoje výhody a nevýhody. V minulosti sa často používali funkcie sigmoid a tanh. V dnešnej dobe je však častejšie využívaná Rectified Linear Unit (ReLU). Prvou výhodou je jednoduchšia definícia funkcie a gradientu. Druhou výhodou je minimalizácia problému strácajúceho sa gradientu pri spätnej propagácii siete. Sigmoid a tanh majú vo väčšine definičného oboru,



Obr. 1.4: Rôzne aktivačné funkcie

Obr. 1.5: Maxpooling na obraze s veľkosťou 2×2 , veľkosťou okna 2×2 a stride 2

okrem hodnôt blízkyh nule veľmi malé gradienty blízke nule. Preto sa pri spätnej propagácii gradienty postupne znižujú, až sa často stratia, teda klesnú na nulu. Tento problém zabraňuje správne učeniu siete. Keďže má ReLU v kladnej časti definičného oboru lineárny gradient, tento problém je výrazne menší. Aj keď má jeho záporná časť nulový gradient, v praktickej implementácii je tento problém zväčša zanedbateľný. [1] Poznáme aj rôzne iné aktivačné funkcie, ktoré ale nie sú až tak často využívané. Ukážky grafov aktivačných funkcií môžeme vidieť na obrázku 1.4.

- Poolingová vrstva - hlavná idea poolingu je zníženie priestorovej veľkosti obrazu, pre zmenšenie vstupu pre ďalšie vrstvy. V oblasti spracovania obrazu by sme mohli pooling prirovnať k zníženiu rozlíšenia obrazu. Pooling vrstva nemá parametre a zachováva hĺbku vstupu. Najčastejšie je využívaný maxpooling. Pri tomto type poolingu posúvame po obraze okno istej veľkosti a do výstupu sa dostane iba maximum z každého tohto okna. Rovnako ako pri konvolúcii môžeme využiť pre obraz padding a rôzne posuny okna, teda rôzny stride. Najčastejšie je využívaný pooling s veľkosťou okna 2×2 , ktorého príklad je zobrazený na obrázku 1.5.

- Plne prepojená vrstva - plne prepojená vrstva je veľmi podobná plne prepojeným vrstvám v klasických neurónových sieťach. To znamená že každý neurón aktuálnej vrstvy je plne prepojený s výstupom predchádzajúcej vrstvy a jeho výstup je prepojený s každým neurónom nasledujúcej vrstvy. Tieto vrstvy majú najväčšie množstvo parametrov v rámci siete a zaberajú pri tréningu najdlhší čas, preto sa počet neurónov zväčša snažíme minimalizovať.

CNN využívané na objektovú detekciu môžeme rozdeliť na dve základné skupiny, podľa ich prístupu k hľadaniu objektov.

1. Dvojkrokové algoritmy - v prvom kroku určujú kandidátske oblasti a v druhom kroku zisťujú, či sa v týchto oblastiach nachádza objekt a spresňujú jeho pozíciu. Jedným zo zástupcov týchto prístupov je RCNN, ktorej architektúra je pomerne jednoduchá. V prvom kroku extrahuje potenciálne oblasti s objektami, pričom v tomto kroku využíva tradičné metódy počítačového videnia. V druhom kroku sú tieto oblasti preškálované do jednotnej veľkosti. V každom z týchto obrazov sú následne extrahované príznaky pomocou CNN. Nakoniec je na týchto príznakov využitá metóda SVM, ktorá určí, aký objekt sa v danej oblasti nachádza. Nevýhodou RCNN je ale pomalé opätovné určovanie príznakov jednotlivých obrazov. Neskôr boli vytvorené vylepšenia tohto algoritmu, ktoré na vytvorenie kandidátskych oblastí aj samotné určenie objektov v týchto oblastiach využívajú neurónové siete.
2. Jednokrokové algoritmy - finálne objekty záujmu hľadajú v jednom kroku. Jedným z najčastejšie využívaných zástupcov tohto prístupu je YOLO, ktorý bol prvým jednokrokovým riešením objektovej detekcie využívajúci CNN. Vstupom CNN je celý obraz. CNN obraz rozdelí na pravidelnú mriežku menších oblastí. V týchto oblastiach vytvorí obdĺžniky, pre ktoré určí triedu a pravdepodobnosť, že sa v obdĺžniku daný objekt nachádza. Tak v jednotlivých oblastiach mriežky dokáže identifikovať aj viacero objektov. Následne podľa pravdepodobností určuje, ktoré z obdĺžnikov najpravdepodobnejšie označujú rovnakú inštanciu objektu, a ktoré rôznu. Aj keď je tento prístup oproti dvojkrokovým algoritmom rýchlejší, zaostáva v presnosti lokalizácie a to najmä pri menších objektoch. [9]

Kapitola 2

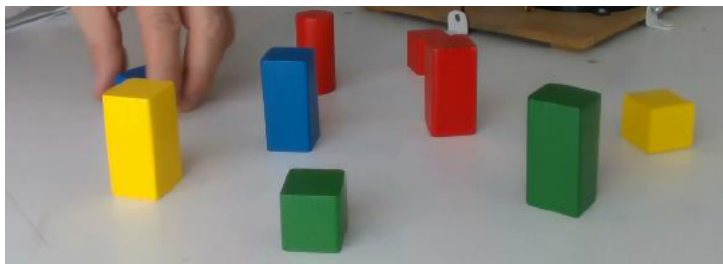
Popis problému a stav problematiky

V tejto kapitole popíšeme problém detekcie objektov pre robotické uchopenie. Popíšeme momentálny stav danej problematiky a konkrétne algoritmy, ktoré tento problém riešia.

2.1 Objektová detekcia v priestore

Pri objektovej detekcii je potrebné veľké množstvo anotovaných dát. Aj keď sú v dnešnej dobe prístupné rozsiahle datasety obsahujúce anotované dáta, konkrétny problém často obsahuje rozdiely voči týmto datasetom. Príkladom môže byť odlišné prostredie v ktorom sa objekty nachádzajú alebo odlišné osvetlenie. Na tieto zmeny v dátach sa CNN nie vždy dokážu bez úprav dobre prispôbiť. Preto nie je vždy možné priamo využiť existujúce a natrénované CNN. Môžeme ale využiť takzvané dotrénovanie (en. fine tuning), pri ktorom je CNN primárne natrénovaná na väčšom datasete, ktorý má podobné dáta. Následne sú niektoré vrstvy dotrénované na našom vlastnom datasete.

Pri základnej objektovej detekcii, popísanej v kapitole 1 určujeme 2D pozíciu objektov v rámci RGB obrazu. Vieme ale, že RGB obraz samotný v sebe nezachytáva hĺbku jednotlivých pixelov, či objektov. Nevieme z neho preto priamo určiť, aké veľké objekty sú, alebo aké sú v obraze vzdialenosti. Taktiež máme v dnešnej dobe na 2D objektovú detekciu k dispozícii veľké množstvo anotovaných dát. Pri určovaní 3D pózy objektu určujeme nielen pozíciu v priestore ale aj orientáciu objektu, takzvaných 6 stupňov voľnosti (6DOF). Konkrétne sú to súradnice na osi x, y, z a taktiež určenie rotácií okolo jednotlivých osí. Pre určenie 6DOF pózy objektu potrebujeme preto viac ako jeho súradnice v RGB obraze. Taktiež pre 6DOF pozíciu objektov nemáme prístupných až také množstvo datasetov a tejto problematike sa oproti určovaniu 2D pozície venuje menšia skupina ľudí.



Obr. 2.1: Stavebnicové kocky záujmu našej práce



Obr. 2.2: Ukážka robota NICO nachádzajúceho sa na našej fakulte

2.2 Špecifikácia zadania našej práce

V našom zadaní je potrebné určiť transláciu a rotáciu farebných drevených stavebnicových kociek. Konkrétne ide o kocky, kvádre a valce vo výraznej červenej, zelenej, žltej a modrej farbe 2.1.

Pozíciu objektov následne robot využije na interakciu s ľudským učiteľom. Časť zaoberajúca sa interakciou robota nie je súčasťou tejto práce. Naším vstupom sú obrázky z kamery robota NICO.

Keďže má robot kameru zabudovanú v oboch očiach, je možné využitie stereo obrazu. Výstupom našej práce má byť objektový detektor určujúci 6DOF pózu jednotlivých stavebnicových kociek v obraze.

2.3 Vstupy pre 3D objektovú detekciu

Poznáme mnohé prístupy, ktoré sa zaoberajú detekciou objektov v 3D priestore. Vzhľadom na ich vstup by sme ich mohli rozdeliť do viacerých základných skupín. Prvou sú siete, ktoré využívajú jednoduchý RGB obraz. Príkladom takéhoto riešenia je napríklad



Obr. 2.3: Anotovaný obraz

sieť DOPE [7], ktorú využívame aj v našej práci. Ďalším možným vstupom je RGBD obraz. Tento obraz má o jednu hĺbkovú vrstvu navyše. Je zväčša zachytený špeciálnymi RGBD kamerami. Tieto kamery slúžia na princípe infračervených vysieláčov a snímačov, vďaka ktorým určujú hĺbku pre každý pixel RGB obrazu. Príkladom využívajúci práve takýto vstup je napríklad Pixor [8].

2.4 Dataset pre tréning CNN

Pri učení neurónových sietí sa vo veľkom množstve prípadov využíva učenie s učiteľom. To znamená, že je to proces, pri ktorom má neurónová sieť pri učení informáciu o správnej odpovedi pre jednotlivé obrazy. Takéto dáta obsahujúce správnu odpoveď sa nazývajú anotované dáta.

Sú to dáta, ktoré pri objektovej detekcii okrem samotného obrazu obsahujú aj informáciu o pozícii objektu. Pri RGB obrazoch to môže byť obdĺžnik ohraničujúci inštancie objektov v obraze. Tento obdĺžnik nazývame bounding box. Príklad môžeme vidieť na obrázku 2.3. Bounding box môže byť určený pozíciami rohov obdĺžnikov, alebo napríklad ich stredmi a rozmermi. Typ anotácie sa líši v závislosti od implementácie neurónovej siete. Anotácia dvojrozmerných dát je pomerne jednoduchá, dokáže ju vykonávať človek. Na túto úlohu existuje veľké množstvo programov. Tento proces teda zahŕňa označovanie jednotlivých objektov obraz po obraze. Keďže neurónové siete potrebujú väčšinou tisíce až státisíce obrazov, tento proces je časovo aj finančne náročný.

Pri trojrozmerných dátach je informáciou o pozícii objektu zväčša kváder, ktorý ohraničuje objekt v priestore. Môžu to byť kvádre rovnobežné s osami x, y, z , kedy je určená 3DOF póza objektu, to znamená, že sa do úvahy neberú rotácie objektu v priestore. V prípade určovania 6DOF pózy objektov je potrebné nielen určiť kváder, ale aj jeho rotáciu. Keďže určovanie takýchto 3D anotácií je pre človeka takmer nemožné pri 2D obraze zobrazovanom na obrazovke, bolo potrebné k získaniu 3D anotovaných

dát pristúpiť iným spôsobom. Jeden z ďalších prístupov k získavaniu anotovaných dát, je generácia syntetických dát.

2.5 Syntetické dáta

Syntetické dáta sú dáta generované umelo počítačom. Často sa snažia čo najlepšie podobáť dátam reálneho sveta, takéto obrazové dáta nazývame fotorealistické dáta. Aj napriek neustále zlepšujúcim sa fyzikálnym modelovacím nástrojom dáta ešte stále nie sú na úrovni reálnych dát. Generovanie syntetických dát je v dnešnej dobe stále viac využívaný prístup najmä vďaka schopnosti získavania veľkého množstva anotovaných dát v pomerne rýchlom čase oproti ručnej anotácii. Väčšina času pri generovaní syntetických dátach je investovaná do modelovania objektov a prostredí. Po vyhotovení modelu je už generovanie samotných obrazov výrazne rýchlejšie voči manuálnemu prístupu človeka. Taktiež pri generovaní dát vieme, kde sa samotné objekty nachádzajú a teda vieme jednoducho určiť aj 6DOF pózu objektu. Toto bolo pri ručnej anotácii takmer nemožné.

Pri trénovaní syntetických dát sa ale stretávame s rôznymi výzvami, pričom jednou z nich je problém prispôsobenia sa realite, nazývaný reality gap problem. Je to jav, pri ktorom sa sieť po učení na syntetických dátach nedokáže prispôsobiť na dáta reálneho sveta a jej výsledky sú veľmi slabé. Preto táto oblasť bola a je súčasťou rozsiahleho výskumu a vedci došli k viacerým postupom, ako tento problém minimalizovať.

2.5.1 Fotorealistické dáta

V minulosti sa aj napriek snahe čo najviac priblížiť realite stále stretávali s problémami, ktoré fyzikálne systémy nedokázali plne namodelovať. Príkladom môhla byť neschopnosť modelovania dokonalých svetlených vlastností, či problém s modelovaním neriigidných objektov. V dnešnej dobe sa simulátory stále zlepšujú a postupne odstraňujú nedostatky, ktorými sa renderované obrazy líšia od reality. Príklad fotorealistických dát využitých na trénovanie neurónovej siete DOPE môžeme vidieť na obrázku 2.4.

Samotné nastavenie takýchto fyzikálne presných modelov je časovo náročné a náchylné na chyby [5]. Práve súhrn týchto javov spôsobuje následne problém prispôsobenia sa realite. Jedným z riešení adresujúcich tento problém je využitie malého množstva reálnych dát pri trénovaní spolu s veľkým množstvom fotorealistických dát. Táto malá pomoc znižuje náklady na potreby veľkého množstva reálnych dát a zároveň adresuje problém prispôsobenia sa na reálne dáta vďaka ich prítomnosti počas trénovania.

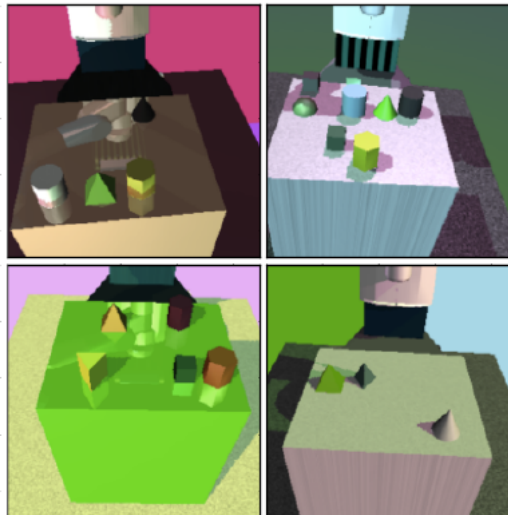


Obr. 2.4: Fotorealistické synteticky vygenerované dáta

2.5.2 Randomizácia prostredia

Mohli by sme povedať, že opačným prístupom ku generovaniu fotorealistických dát, je generovanie dát, ktoré sa minimálne podobajú reálnemu svetu. Randomizácia prostredia, z anglického domain randomization, je proces, kedy generujeme veľké množstvo dát podobajúcich sa realite minimálne. Zatiaľ čo zachováваме tvary objektov, štruktúry či pozadie, sú často nereálne. Ako vidíme na obrázku 2.5, dokážeme na ňom rozlíšiť jednotlivé geometrické tvary, samotný obraz však pripomína realitu iba minimálne.

Prostredie bolo generované z jednoduchých geometrických tvarov, pričom farby a textúry objektov boli v jednotlivých obrazoch náhodne určované. Základ tohto prístupu spočíva v tom, že keď sieť uvidí dostatočné množstvo rozličných dát, obrazy reálneho sveta pri testovaní sa sieti budú javiť iba ako ďalšia variácia toho, čo videla počas tréningu. Tieto prístupy boli testované vo viacerých formách. Pri niektorých prístupoch boli použité iba základné farebné textúry, pri iných boli použité zložitejšie textúry reálnych objektov. Tento prístup bol využitý v článku [5], kde pri tréňovaní využili iba syntetické randomizované dáta. Pri istých objektoch sa im podarilo dosiahnuť presnosť dostačujúcu na robotické uchopenie objektov. Tento úspech bol čiastočne podmienený aj pomerne jednoduchou štruktúrou objektov a pri zložitejších tvaroch výsledky neboli dostatočne presné. Preto neskôr začali výskumníci tento prístup využívať v kombinácii s reálnymi alebo fotorealistickými dátami pre získanie lepších výsledkov.



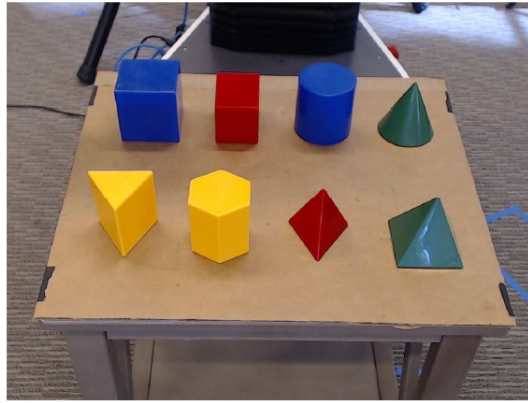
Obr. 2.5: Dáta generované princípom randomizácie prostredia

2.6 Existujúce riešenia

V tejto časti sa pozrieme na existujúce riešenia k rovnakému, alebo výrazne podobnému problému.

2.6.1 Určovanie 3D pozície centroidov objektov pomocou CNN

V tomto článku sa autori venovali využitiu iba synteticky vytvorených dát princípom randomizácie prostredia [5]. Objekty záujmu boli podobne ako v našom prípade drevené stavebnicové kocky, zobrazené na obrázku 2.6. Pri generácii dát vytvárali jednoduché scény zložené najmä zo základných geometrických objektov (kocky, kvádre, ...). Hlavnou časťou riešenia sa ukázala byť vysoká variabilita textúr objektov a pozadia. Následne overovali, či sú dané dáta dostatočné na získanie výsledkov dostatočných pre získanie pozície objektu potrebnej pre robotické uchopenie. V tomto článku na rozdiel od nášho problému určujú iba triedu objektu a pozíciu jeho centroidu, teda neberú do úvahy rotácie objektov. Na získanie centroidu využili pomerne jednoduchú modifikovanú architektúru siete VGG-16. Podarilo sa im tak lokalizovať objekty priemerne do vzdialenosti jeden a pol centimetra. Zatiaľčo pri jednoduchších objektoch, ako je kocka alebo hranol, sa im podarilo dosiahnuť pomerne stabilné výsledky s maximálnou odchýlkou dva centimetre, pri zložitejších objektoch boli výsledky menej stabilné a maximálna chyba pri štvorstene sa vyšplhala na desať centimetrov.



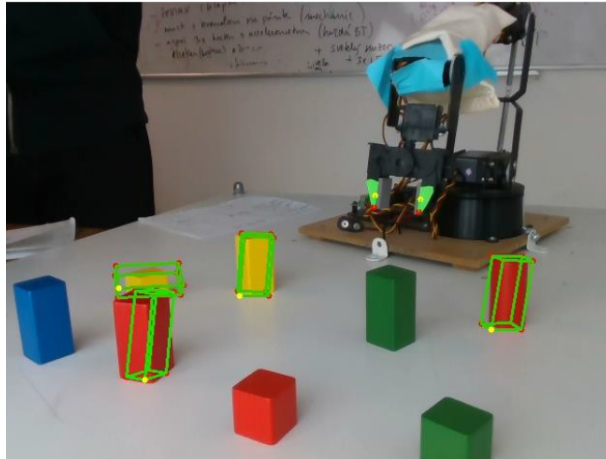
Obr. 2.6: Objekty záujmu pri metóde hľadajúca pozíciu centroidov

2.6.2 Tradičné metódy pre určenie 6DOF pózy drevených stavebníc

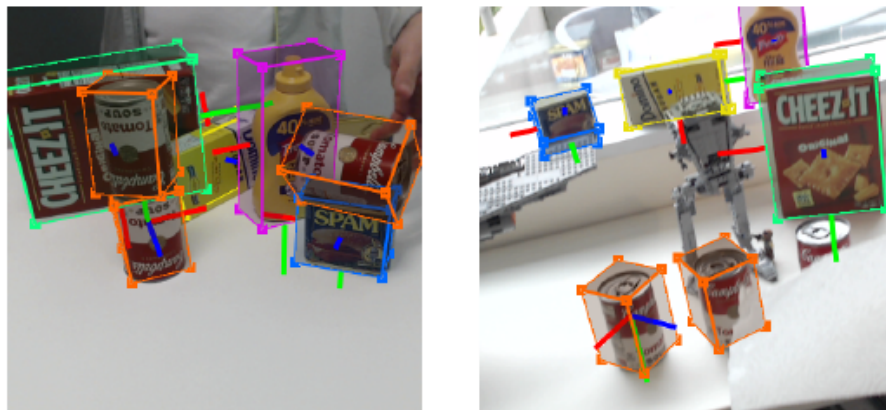
V tejto práci bolo cieľom určovanie 3D pozície objektov pomocou RGB obrazu. Tento problém bol v minulosti na našej fakulte riešený využitím tradičných metód počítačového videnia. Vstupom bol v tom čase obraz z RGBD intenal realsense d435i kamery. Prvým krokom tejto metódy bolo hľadanie objektov pomocou farebnej segmentácie RGB obrazu, pričom farby hľadaných kociek boli vopred známe a v implementácii pevne nastavené. Z RGBD dát bol následne vytvorený point cloud, teda množina bodov v priestore. Tento krok bol možný vďaka hĺbkovej informácii o každom pixeli. Následne boli hľadané hlavné roviny jednotlivých kociek pomocou metódy RANSAC. Následne boli určené body hlavnej roviny kocky a vedľajších stien. Nakoniec bola vytvorená projekcia daných bodov pre získanie bounding boxu. Okrem toho, že táto metóda bola vzhľadom na využitie farieb veľmi náchylná na svetelné podmienky a vysoko závislá od použitej kamery, nepodarilo sa jej dosiahnuť dostačujúce výsledky pre rozsiahlejšie využitie v ďalších projektoch. Príklad výstupu tohto postupu je ukázaný na obrázku 2.7.

2.6.3 Deep object pose estimation - DOPE

DOPE je jedným z riešení zaoberajúcich sa určovaním 6DOF pózy objektu z RGB obrazu [7]. V tomto článku sa autori zaoberajú využitím fotorealistických dát a dát generovaných využitím metódy randomizácie prostredia na natrénovanie CNN pre určovanie 6DOF pózy bez dotrénovania na reálnych dátach. Autori ukazujú, že využitie oboch skupín dát dosahuje výrazne lepšie výsledky ako využitie iba jednej zo skupín dát. Konkrétne tento prístup testovali na objektoch domáceho využitia z datasetu YCB. CNN na vstupe dostane RGB obraz a má dva výstupy. Prvým výstupom je 9 máp vieryhodnosti (en. belief maps), 1 pre každý vrchol bounding boxu a posledná pre



Obr. 2.7: Príklad výsledku využitia tradičných metód na reálnych dátach



Obr. 2.8: Príklad výsledku využitia DOPE na reálnych dátach

centroid objektu. Druhým výstupom je 8 vektorových polí (en. vector fields), ktoré pre každý vrchol určujú smer od neho k centroidu objektu. Po tom, ako sú CNN vyrátané mapy vieryhodnosti a vektorové polia, je z nich potrebné extrahovať samotné inštancie objektov. Na rozdiel od iných prístupov, na extrakciu objektov je využitý pomerne jednoduchý algoritmus. Najprv v mapách vieryhodnosti vyhľadáva lokálne minimá nad istým prahom. Následne pažravý algoritmus priradzuje jednotlivé body k centroidom na základe vektorov jednotlivých vrcholov vzhľadom na všetky centroidy. Vrchol následne priradí najbližšiemu centroidu v istom uhlovom rozmedzí daného vektoru. Na záver po určení inštancií objektov sú využité jednotlivé vrcholy, vlastnosti kamery, ale aj rozmery objektu na získanie finálnej translácie a rotácie objektu. Výsledky siete môžeme vidieť na obrázku 2.8. Vďaka výstupu siete vyhovujúcemu požiadavkám nášho problému sme sa toto riešenie rozhodli využiť v našich experimentoch.

Kapitola 3

Návrh riešenia problému

V tejto kapitole si popíšeme jednotlivé kroky postupu riešenia problému. Najprv popíšeme spôsoby generovania dát. Následne popíšeme využité algoritmy na získanie 3D pózy objektov.

3.1 Generovanie dát

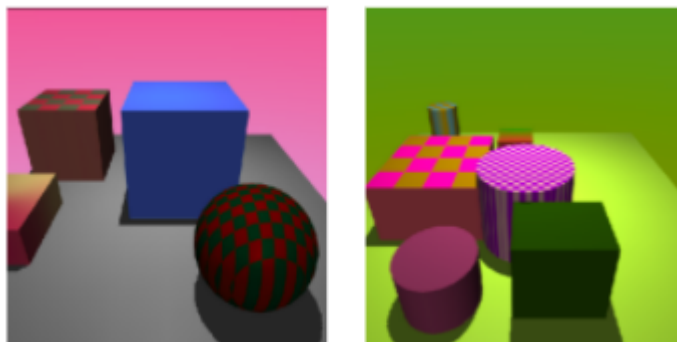
Pre náš konkrétny prípad sme nemali k dispozícii žiadny anotovaný dataset. K dispozícii sme mali niekoľko videí natočených RGB intel real sense d435i kamerou. Tieto dáta však nemali anotácie, teda neboli určené jednotlivé pozície objektov v snímkoch videa. Počas našej práce sa taktiež zmenil robot pre daný projekt, preto dané zábery neboli využiteľné.

Keďže je ručná anotácia veľkého množstva 3D dát v našich podmienkach nemožná, rozhodli sme sa na tréning CNN využiť generáciu syntetických dát popísanú v kapitole 2. Konkrétne ide o randomizáciu prostredia. Tento prístup je omnoho jednoduchší a taktiež rýchlejší na prípravu scén.

3.2 Využitie randomizácie prostredia a MuJoCo Physics Engine

Pri generovaní jednoduchých dát sme sa inšpirovali článkom využívajúci vstavaný modul na renderovanie obrazov od MuJoCo Physics Engine's [6]. Modely sa dajú generovať jednoducho pomocou vytvorenia xml súboru obsahujúceho informácie o scéne. Naša jednoduchá scéna bola zložená zo stola - kvádra a jednoduchých geometrických útvarov uložených na stole. Súbor scény vo formáte xml generujeme pomocou jednoduchého programu v jazyku python.

Ako prvý krok náhodne vyberieme počet objektov na stole (4 až 7) a ich typy - kocka, kváder, guľa alebo valec. Následne vygenerujeme textúru pre každý objekt,



Obr. 3.1: Príklady vygenerovaných dát s použitím MuJoCo Physics Engine

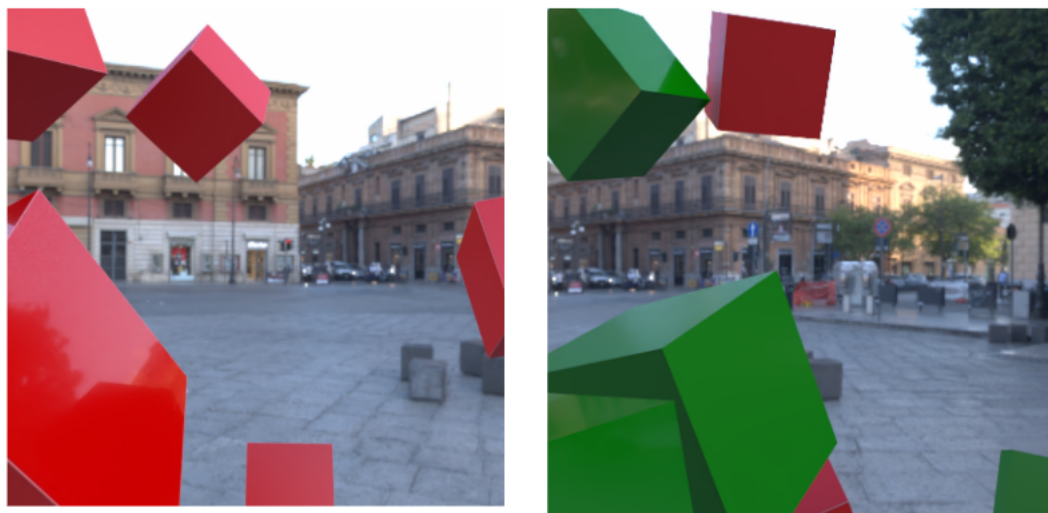
pozadie a stôl. Textúry su vyberané náhodne a sú využité prednastavené jednoduché textúry renderovacieho modulu. Medzi možnosťami textúry sú náhodne vybraná RGB farba, gradient medzi dvomi náhodne vybranými farbami a šachovnicový vzor dvoch náhodne vybraných farieb. Vďaka hotovým textúram stačí vybrať farby a mieru odlesku materiálu. Následne sú náhodným objektom priradené potrebné rozmery. Keď máme jednotlivé objekty otextúrované a naškálované, je potrebné ich umiestniť na stôl. Objekty sú ukladané tak, aby sa neprekrývali, keďže stavebnice sa v reálnom svete taktiež nedokážu prekryvať. To ale neznamená, že sa nebudú prekryvať vo finálnom vyrenderovanom obraze. Nakoniec sú z danej scény vygenerované dva obrazy simulujúce dve kamery robota NICO. Príklady vygenerovaných obrazov je možné vidieť na obrázku 3.1.

V doterajšom generovaní sme nevyužili rotáciu objektov, ani zmenu pozície kamery. Prvý spomínaný nedostatok bude doplnený. Keďže NICO potrebuje byť pevne upevnený na jednom mieste, problém pohyblivej kamery môže byť potenciálne zandbateľný. V prípade, že by tento jav spôsoboval nízke výsledky našej siete, bude pohyblivá kamera do programu doplnená. Pozícia kamery sa do xml súboru scény dá pomerne jednoducho doplniť, je ale potrebné premyslieť natočenia a možné pozície jednotlivých kamier, aby bol dataset v tomto smere zmysluplný.

3.3 Využitie NViSII generácie dát

Pri skúmaní možností využitia DOPE CNN na detekciu objektov sme zistili, že autori poskytujú pomerne jednoducho využiteľný program na generáciu anotovaných obrazov s nami definovanými objektami. Tieto obrazy sa dajú následne priamo použiť na tréning siete.

Tento python program využíva NVISII - NVIDIA Scene Imaging Interface [3]. Je to pre python vytvorený interface založený na technológii sledovania lúča (en. ray tracing) so základom NVIDIA OptiX (rozhranie API na technológiu sledovania lúčov



Obr. 3.2: Príklady vygenerovaných dát s použitím programu založenom na NVISII

využívajúce NVIDIA GPU pre optimalizáciu výpočtov).

V tomto programe je možné definovať veľkosť výstupného obrazu, počet a druh objektov v obraze, či ich veľkosť a textúru. Objekty sú primárne načítané z formátu obj. Využitím tohto programu sme získali nasledovné výsledky, zobrazené na obrázku 3.2.

3.4 Výber neurónovej siete

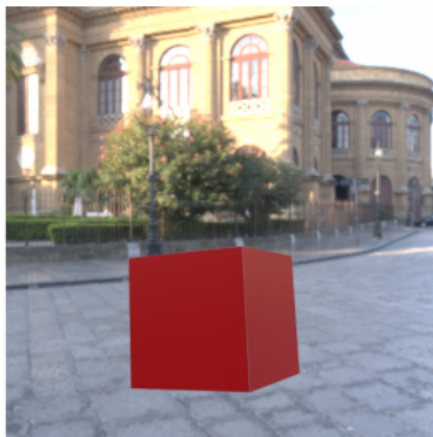
Kapitola 4

Výsledky experimentov

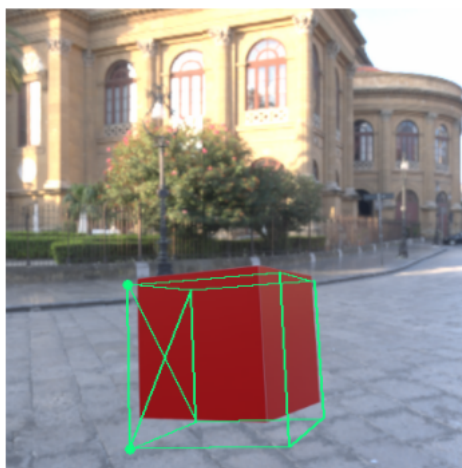
4.1 Dataset jedného obrazu s jednou kockou

Vybraná architektúra siete nemusí byť pre potrebnú aplikáciu vždy ideálna a tréning na veľkom datasete počas množstva epoch, zvykne trvať dlhú dobu. Preto sme testovali, či je sieť DOPE schopná detegovať naše objekty záujmu tak, že sme vytvorili najjednoduchší možný dataset, pozostávajúci z jedného obrazu obsahujúceho jednu kocku. V tomto prípade by sa mala sieť pomerne jednoducho naučiť správne detegovať kocku v danom obraze. V prípade, že by sa sieť kocku ani po dlhom trénovaní nenaučila rozpoznať, znamená to, že je niekde v procese tréningu siete alebo v samotnom princípe využitia danej architektúry pre konkrétny problém nevhodná. Využitý obraz vidíme na obrázku 4.1.

Po trénovaní siete 650 epoch môžeme vidieť výstup siete na obrázku 4.2. Už na prvý pohľad môžeme vidieť, že sieti sa podarilo pomerne správne určiť pozíciu a rotáciu kocky.



Obr. 4.1: Obraz datasetu veľkosti jeden, na ktorom bola sieť trénovaná



Obr. 4.2: Výstup siete pre obraz z datasetu veľkosti jeden, na ktorom bola sieť tréno-
vaná

Záver

Cieľom tejto práce bolo vyskúšať využitie konvolučných neurónových sietí na 3D objektovú detekciu stavebnicových kociek v RGB obraze robota NICO. Základom nášho tréningu boli synteticky vygenerované anotované dáta a konvolučná neurónová sieť DOPE.

Najprv sme zhodnotili možnosti generovania dát a vytvorili sme menej rozsiahle datasety, obsahujúce rôzne počty základných červených kociek. Na tomto datasete sme otestovali schopnosť siete učiť sa určiť správnu pozíciu a orientáciu kocky. Po vyhovujúcich výsledkoch sme vytvorili rôzne datasety obsahujúce aj kocky iných farieb a sieť sme na týchto datasetoch natrénovali.

Literatúra

- [1] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6. Ieee, 2017.
- [2] Mahmoud Hassaballah, Aly Amin Abdelmgeid, and Hammam A Alshazly. Image features detection, description and matching. In *Image Feature Detectors and Descriptors*, pages 11–45. Springer, 2016.
- [3] Nathan Morrical, Jonathan Tremblay, Yunzhi Lin, Stephen Tyree, Stan Birchfield, Valerio Pascucci, and Ingo Wald. Nvisii: A scriptable tool for photorealistic image generation, 2021.
- [4] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.
- [5] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.
- [6] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.
- [7] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. *arXiv preprint arXiv:1809.10790*, 2018.
- [8] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018.
- [9] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*, 2019.

Prílohy

Prílohy prace