VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

DETEKCE KVÁDRŮ-KRABIC V OBRAZE

DETECTION OF BOXES IN IMAGE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE **ADAM ŽITŇANSKÝ**
AUTHOR

VEDOUCÍ PRÁCE **prof. Ing. ADAM HEROUT, Ph.D.**
SUPERVISOR

BRNO 2021

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2020/2021

Zadání bakalářské práce

Student: **Žitňanský Adam**
Program: Informační technologie
Název: **Detekce kvádrů-krabic v obraze**
Detection of Boxes in Image
Kategorie: Zpracování obrazu

Zadání:

1. Seznamte se s problematikou počítačového vidění na bázi strojového učení a konvolučních neuronových sítí.
2. Pořiďte (a průběžně zvětšujte a zkvalitňujte) datovou sadu pro vývoj systému pro detekci krabic (v širokém smyslu) v obraze.
3. Experimentujte s vhodnými algoritmy strojového učení pro detekci krabic v obraze. Pro učení a vyhodnocování použijte shromážděnou datovou sadu.
4. Iterativně vyhodnocujte a vylepšujte vytvořené řešení.
5. Zhodnoťte dosažené výsledky a navrhněte možnosti pokračování projektu; vytvořte plakátek a krátké video pro prezentování projektu.

Literatura:

- Bharath Ramsundar, Reza Bosagh Zadeh: TensorFlow for Deep Learning: From Linear Regression to Reinforcement Learning, O'Reily Media, 2018
- Gary Bradski, Adrian Kaehler: Learning OpenCV; Computer Vision with the OpenCV Library, O'Reilly Media, 2008
- Richard Szeliski: Computer Vision: Algorithms and Applications, Springer, 2011

Pro udělení zápočtu za první semestr je požadováno:

- body 1.-2., značné rozpracování bodů 3. a 4.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Herout Adam, prof. Ing., Ph.D.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2020
Datum odevzdání: 12. května 2021
Datum schválení: 3. května 2021

Abstrakt

Táto práca sa zaoberá problémom detekcie kvádrov, konkrétne krabíc v obraze. Hlavným výsledkom práce je návrh a implementácia systému na detekciu krabíc na základe rohov a hrán. Systém pozostáva z detektora hranových bodov a rohov založeného na konvolučnej neurónovej sieti a dekodéri na zostavenie výsledného 2d modelu krabice. V rámci riešenia vznikol taktiež dataset s 550 anotovanými snímkami krabíc s anotáciami rohov a hrán.

Abstract

This thesis addresses the problem of cuboid detection, more specifically boxes detection in images. The main result is the implementation of a system for boxes detection based on corners and edges. The system consists of a CNN regression-based corner and edge points detector and decoder, which takes CNN output and turns it into a 2d model of the cuboid. As a part of this work also a a dataset of boxes with 550 images with corners and edges annotations was created.

Kľúčové slová

konvolučné neurónové siete, počítačové videnie, detekcia objektov, detekcia kľúčových bodov, detekcia kvádrov, Stacked Hourglass Network

Keywords

convolutional neural networks, computer vision, object detection, keypoint detection, cuboid detection, Stacked Hourglass Network

Citácia

ŽITŇANSKÝ, Adam. *Detekce kvádrů-krabic v obraze*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce prof. Ing. Adam Herout, Ph.D.

Detekce kvádrů-krabic v obraze

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením prof. Ing. Adama Herouta Ph.D. a uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....
Adam Žitňanský
10. mája 2021

Poďakovanie

Týmto by som chcel poďakovať môjmu vedúcemu prof. Ing. Adamovi Heroutovi Ph.D za odborné vedenie, podporu a cenné rady, ktoré mi veľmi pomohli pri tvorbe tejto práce.

Obsah

Kapitola 1

Úvod

Každý deň je vo svete zabalené a transportované obrovské množstvo tovaru, z čoho je veľké množstvo zabalené v kartónových krabiciach. Z tohto dôvodu je detekcia krabíc aktuálnou témou v kontexte automatizácie v odvetviach ako preprava tovarov, balenie tovarov a mnohých iných, kde sú krabice vo veľkom využívané.

Cieľom tejto práce je navrhnúť a následne aj implementovať a vyhodnotiť systém na detekciu krabíc. Navrhovaný systém bude založený na detekcii kľúčových bodov, konkrétne rohov a hranových bodov krabice. Tento prístup je inšpirovaný postupmi využívanými v oblasti odhadu pózy ľudskej postavy. Pre účely detekcie kľúčových bodov systém využíva regresiu tepelných máp (heatmap regression) s využitím konvolučnej neurónovej siete. Výstup neurónovej siete je následne ďalej spracúvaný za účelom získať konečný výstup – modely tvaru pre jednotlivé krabice vo vstupnom obrázku.

Na rozdiel od detektorov založených na regiónoch, ktorých výstupom je osovo zarovnaný obdĺžnik (bounding box) vyznačujúci pozíciu detegovaného objektu, sú detektory založené na kľúčových bodoch schopné presnejšie lokalizovať krabicu na základe pozícií jej hrán a rohov. Tento spôsob popisu poskytuje presnejšie informácie o orientácii a rozmeroch, a teda je vhodný pre použitie v aplikáciach, kde je na tieto informácie kladený dôraz. Príkladom môže byť napríklad automatická manipulácia s krabicami na triediacej linke.

Kapitola 2

Detekcia objektov a existujúce systémy na detekciu kvádrov

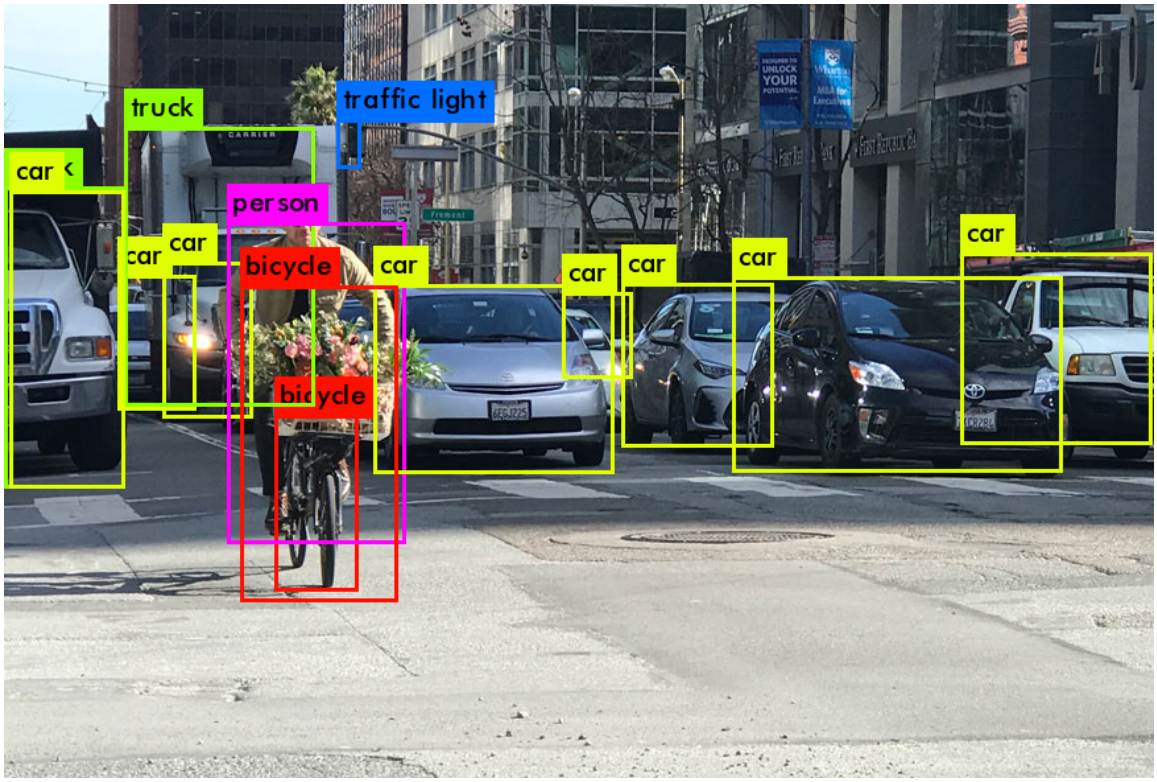
Detekcia objektov [6] je odvetvie počítačového videnia ktoré sa zaoberá zisťovaním výskytu a pozície sémantických objektov danej triedy v obraze a vo videu. Jednotlivé prístupy k detekcií objektov sa dajú rozdeliť podľa použitej technológie na metódy založené na strojovom učení a metódy využívajúce neurónové siete. V súčastnosti je pre toto odvetvie počítačového videnia, podobne ako pre mnoho iných, charakteristický nárast využitia neurónových sietí. Konkrétne sa pre tento účel využívajú najmä konvolučné neurónové siete (podrobne v kapitole 3).

Detegované objekty sú najčastejšie lokalizované obdĺžnikovou oblasťou (bounding box) pokrývajúcou detegovaný objekt, prípadne aj s označením triedy objektu. Tento spôsob popisu teda poskytuje informáciu o tom, aké objekty sa v obrázku nachádzajú, ale aj informáciu o oblasti, kde sa nachádzajú. Lokalizácia formou osovo zarovnaného bounding boxu je pre mnoho požití plne dostačujúca, avšak ak sú z nejakého dôvodu potrebné presnejšie informácie o pozícii, orientácii a rozmeroch detegovaného predmetu, môže byť tento spôsob popisu nedostačujúci. V týchto prípadoch je nutné siahnuť po alternatívnych spôsoboch, ako napríklad detekcia kľúčových bodov. Pomocou množiny kľúčových bodov je totiž možné presnejšie popísať spomínané priestorové aspekty ako rozmery a orientácia objektu.

2.1 Detektory objektov

Ako bolo spomenuté, detekcia objektov spája dve úlohy, ktorými sú určenie triedy objektu – klasifikácia a zároveň jeho pozície v obrázku – lokalizácia [31]. Pozícia objektu je zvyčajne vyjadrená v podobe obdĺžnikovej oblasti (bounding boxu), ktorá zvykne byť osovo zarovnaná.V prípade tradičných detektorov sa algoritmus detekcie dá rozdeliť do 3 krokov. Prvým krokom je tzv. informatívny výber regiónov. Keďže objekty sa môžu v obraze vyskytovať v rozličných veľkostiach a s rôznym pomerom strán, prirodzenou voľbou je skenovanie obrázku kĺzavými oknami rôznych rozmerov. Hlavným problémom tohto prístupu je jeho časová náročnosť. Taktiež pri výbere nevhodných veľkostí kĺzavých okien môže byť toto rozdelenie nedostatočné z pohľadu pokrytia jednotlivých objektov. Z týchto dôvodov existuje viacero prístupov, z ktorých každý rieši spomínané problémy iným spôsobom s cieľom nájsť kompromis vzhľadom na pomer rýchlosti a presnosti.

Ďalším krokom je klasifikácia jednotlivých regiónov. Účelom tohto kroku je zistiť, či jednotlivé regióny z predošlého kroku obsahujú hľadaný objekt, a priradiť mu triedu. V prí-



Obr. 2.1: Ukážka vyznačenia delegovaných objektov vo forme bounding boxu, triedy prislúchajúcej danému objektu a skóre. Je možné pozorovať aj to, že v niektorých prípadoch tvar bounding boxu nemusí dostatočne vystihovať tvar detekovaného objektu a teda nenesie presnejšie informácie o tvare, orientácii a rozmeroch objektov. Obrázok bol prevzatý zo zdroja[2].

pade detektorov, ktoré využívajú za účelom klasifikácie metódy strojového učenia, napríklad SVM [4], je nutný ešte ďalší medzikrok. Tým je extrakcia vizuálnych príznakov, napríklad histogram orientovaných gradientov (HOG) [5] alebo SIFT [14]. Na základe týchto príznakov potom prebieha samotná klasifikácia. Pri klasifikácií pomocou neurónových sietí tento krok nutný nieje. Práve odstránenie nutnosti extrakcie príznakov je hlavným dôvodom rozmachu detektorov využívajúcich neurónové siete. Do tejto kategórie sa radí mnoho architektúr ako RCNN [9], YOLO [21], SSD [13], ktoré sa navzájom líšia najmä v spôsobe, akým kombinujú informatívny výber regiónov a klasifikáciu. Niektoré architektúry (YOLO) tieto kroky spájajú a vykonávajú priamo regresiu pevného počtu bounding boxov spolu so skóre jednotlivých tried objektov. Tým sa stáva architektúra plne konvolučnou, čo umožňuje aby detekcia prebiehala v jednom priechode. To vedie k výraznému zrýchleniu v porovnaní s mnohonásobnou klasifikáciou jednotlivých regiónov. Ako bolo spomenuté, nedostatkom je fakt, že osovo zarovnaný bounding box je pre niektoré aplikácie nedostatočný z pohľadu presnosti lokalizácie objektu, pretože nenesie informáciu o orientácii objektu a informácia o rozmeroch je taktiež obmedzená. Praktický príklad tejto reprezentácie je možné vidieť na obrázku 2.1. Na tomto obrázku si možno všimnúť aj spomenutý problém, kedy osovo zarovnaný bounding box nevystihuje tvar objektu.

[2] https://medium.com/analytics-vidhya/yolo-object-detection-made-easy-7b17cc3e782f

2.2 Detekcia kľúčových bodov

Jedna z možností, ako presne popísať polohu objektu, je pomocou súradníc niekoľkých pre daný objekt charakteristických bodov zvaných kľúčové body (key points). Tento prístup sa často používa v oblastiach ako je odhad pózy ľudskej postavy [3], kde kľúčové body prislúchajú jednotlivým kĺbom a hlave a spoločne popisujú torzo ľudskej postavy. Podobne je možné použitie pre lokalizáciu objektov tvaru kvádra [8], kde kľúčové body predstavujúce vrcholy telesa, ale aj zložitejších objektov, napríklad lietadiel [30]. Taktiež existujú obecné detektory objektov, ktoré ako alternatívu k bounding boxu využívajú pre popis pozície objektu kľúčové body. Príkladom je architektúra CenterNet [7], ktorá popisuje objekty pomocou trojice bodov – stredu ľavého horného a pravého dolného rohu. Príklady toho, ako môžu byť využívané kľúčové body pre popis objektov vo vybraných aplikáciach, je možné vidieť v obrázku 2.2.

Z pohľadu technológii používaných na detekciu kľúčových bodov v súčastnosti aj v tejto oblasti dominujú konvolučné neurónové siete(kap. 3.2). V rámci detekcie kľúčových bodov neurónovými sieťami sa využívajú dva hlavné prístupy, a to priama regresia súradníc kľúčových bodov a nepriama tzv. regresia tepelných máp (heatmap regression). Pri priamej regresii sú výstupom siete priamo súradnice jednotlivých kľúčových bodov. Na rozdiel od toho regresia tepelných máp spočíva v predikcií skóre výskytu kľúčového bodu pre jednotlivé pixely tzv. confidence mapa. Toto skóre možno interpretovať tak, že čím je jeho hodnota vyššia, tým si je model istejší, že sa na danej pozícii vyskytuje kľúčový bod. Pri tomto prístupe sú potom samotné súradnice kľúčových bodov získané až v ďalšom kroku, najčastejšie ako súradnice pixelu s najvyšším skóre v confidence mape pre daný typ kľúčového bodu.

2.2.1 Priama regresia súradníc

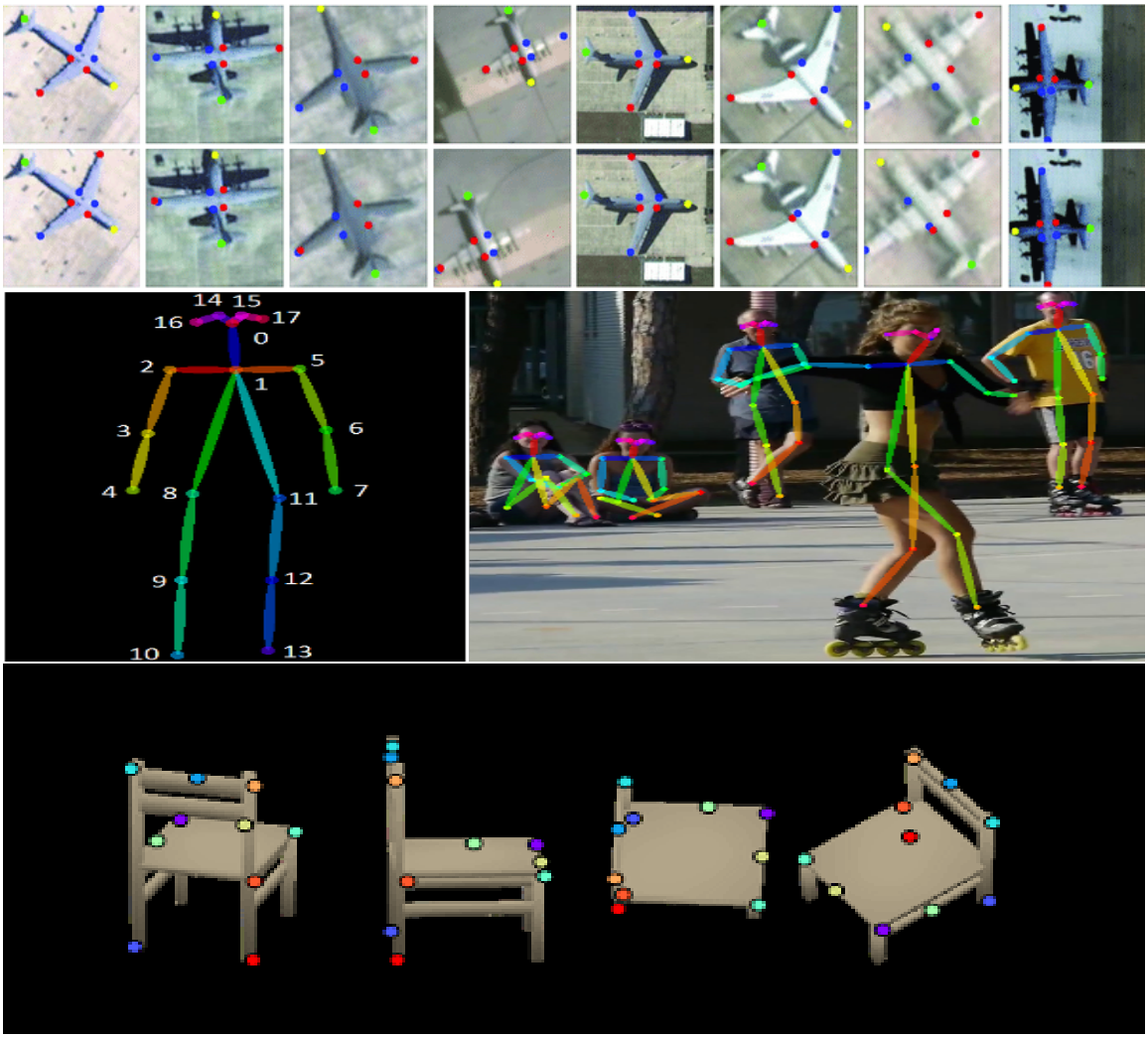
Jedným z používaných prístupov pri detekcii kľúčových bodov s použitím konvolučných neurónových sietí je tzv. priama regresia kľúčových bodov. Pri tomto prístupe má neurónová sieť za úlohu priamo odhadnúť súradnice pevne daného počtu kľúčových bodov. Keďže sieť produkuje najčastejšie dvojice súradníc x, y (v prípade 2D) pre každý kľúčový bod, jedná sa o regresiu 2n skalárnych hodnôt, kde n je počet detegovaných kľúčových bodov. Z toho plynie fakt že počet detegovaných kľúčových bodov musí byť pevne daný, nakoľko je na ňom závislá architektúra použitej neurónovej siete . To prináša aj obmedzenie vo voľbe prístupu k samotnej detekcií na prístup zhora dolu, kedy je potrebné najprv identifikovať oblasti s jednotlivými inštanciami objektov vo vstupnom snímku, a následne pre každú oblasť detegovať pevne daný počet kľúčových bodov.

Podľa niektorých zdrojov je tento prístup taktiež menej robustný a dosahuje horšiu výkonnosť ako nepriamy prístup založený na regresii tepelných máp. Tompson [26] toto prisudzuje najmä faktu, že priama reprezentácia vo forme súradníc je príliš nelineárna, a teda problematická z pohľadu učenia siete.

2.2.2 Regresia tepelných máp (Heatmap regression)

Takzvaná regresia tepelných máp (heatmap regression) je alternatívnou metódou k priamej regresii súradníc kľúčových bodov. Táto metóda spočíva v inej reprezentácii kľúčových bodov. Tie v tomto prípade nie sú reprezentované n súradnicami, ako tomu bolo v prípade

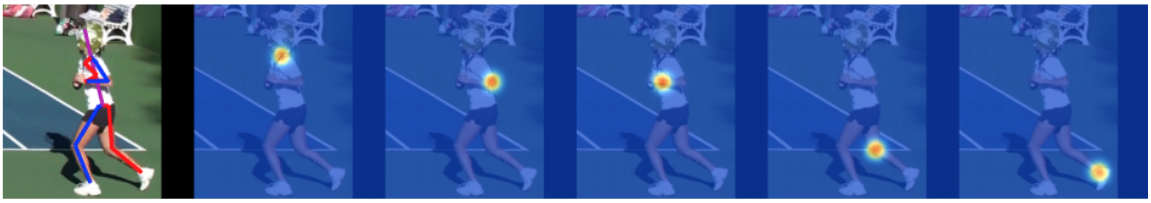
[4] https://medium.com/beyondminds/an-overview-of-human-pose-estimation-with-deep-learning-d49eb656739b



Obr. 2.2: Ukážka využitia detekcie kľúčových bodov pre rôzne aplikácie a na rôznych druhoch objektov. Konkrétne zhora využitie pre lokalizáciu a určenie typu lietadla, odhad ľudskej pózy a na dolnom obrázku popis stoličky pomocou kľúčových bodov. Obrázky boli prevzaté zo zdroja[4], [8] a [24].

priamej regresie, ale ako miesta s vysokým skóre v confidence pre jednotlivé pixely vstupného obrázka. Toto skóre sa najčastejšie interpretuje ako pravdepodobnosť výskytu daného kľúčového bodu na danej pozícii. Keďže vo väčšine prípadov použitia je potrebné rozlíšiť niekoľko tried kľúčových bodov, výstup zvyčajne pozostáva z viacerých matíc predikcii – confidence máp, kde každá popisuje výskyt jedného typu kľúčového bodu. Jednotlivé kľúčové body sú v ground-truth dátach označené zvyčajne ako 2D Gaussove krivky, kde vrchol Gaussovej krivky predstavuje presnú pozíciu príslušného kľúčového bodu.

V porovnaní s priamou regresiou súradníc má tento prístup niekoľko výhod. Prvou z nich je, že táto forma reprezentácie je jednoduchšie z pohľadu učenia neurónovej siete, a to vďaka väčšej linearite. Ďalšou výhodou je možnosť detegovať viacero inštancií kľúčových bodov, vďaka čomu je táto metóda vhodná aj pre detekciu tzv. zdola na hor. V tomto prípade sa vo vstupnom obrázku môže vyskytovať ľubovoľný počet objektov, a teda aj kľúčových bodov rovnakého typu. V neposlednom rade umožňuje táto reprezentácia lepšiu vizualizáciu



Obr. 2.3: Ukážka confidence máp pre jednotlivé kĺby, ktoré sú vizualizované v podobe tepelných máp. Jednotlivé kĺby sú reprezentované v podobe 2D Gaussových kriviek. Pre každý typ kĺbu je vyhradený samostatný kanál obsahujúci skóre výskytu daného kľúčového bodu pre jednotlivé pozície vo vstupnom obrázku. Obrázok bol prevzatý z [18].

rozhodovania siete. Vďaka týmto faktom a najmä väčšej presnosti a robustnosti sa stal tento prístup preferovaný v odvetví odhadu ľudskej pózy, kde túto metódu využíva väčšina state of the art riešení [1]. Príklad ground truth dát, kde sú kľúčové body reprezentované týmto spôsobom, je zobrazený v obrázku 2.3.

2.3 Detekcia predmetov tvaru kvádra a existujúce riešenia

Keďže krabice majú vo veľkej väčšine tvar kvádra, detekcia objektov tvaru kvádra (cuboid detection) je príbuzná problému detekcie krabíc, ktorému sa venuje táto práca. Detekcia kuboidov je špecifická najmä popisom pozície detegovaných objektov. Z geometrického hľadiska je kváder kolmý rovnobežnosten s podstavou pravouhlého štvoruholníka. Z toho plynie, že kváder tvorí 6 stien, 8 vrcholov a 12 hrán. Na základe nich je možné popísať jeho pozíciu niekoľkými spôsobmi. Pre účely detekcie sa javí ako vhodný popis na základe pozícii vrcholov, alebo alternatívne popis na základe pozície ťažiska a 3 rozmerov – dĺžky výšky a šírky.

Z hľadiska existujúcich riešení existuje niekoľko publikácii venujúcich sa detekcii kuboidov. V tejto kapitole budú dve z nich popísané podrobnejšie, a to jedna založená na metódach strojového učenia a druhá využívajúcu konvolučnú neurónovú sieť s metódou priamej regresie kľúčových bodov(podrobne v kapitole 2.2.1).

2.3.1 Detektor kuboidov založený na HOG a 3D modeli

Publikácia s názvom *Localizing 3D cuboids in single-view images* [28] predstavuje prístup k detekcii kuboidov založený na 3D modeli tvaru kvádra a HOG [5] príznakoch. Systém využíva klasifikáciu na základe HOG s využitím SVM [4] na detekciu rohov kuboidu. Jednotlivé konfigurácie kuboidu sú potom hodnotené na základe hodnotiacej funkcie, ktorá sa skladá z niekoľkých častí, kde každá hodnotí iný aspekt. Medzi tieto časti patrí skóre kvalifikátora, 2D displacement rohov, model vnútorných hrán a 3D model kuboidu. Algoritmus spočíva v získaní niekoľkých počiatočných aproximácii, ktoré sú následne optimalizované na základe hodnotiacej funkcie s využitím metódy randomised hill-climbing.

Hlavnou nevýhodou tohto prístupu je podľa zdroja [8] veľká výpočetná náročnosť spôsobená procesom optimalizácie. Konkrétne sa v tejto práci uvádza, že tento prístup vyžaduje viac ako minútu pre spracovanie jediného vstupného obrázka. Vyhodnotenie presnosti tak, ako bolo prezentované autormi v pôvodnej práci je zobrazené v obrázku 2.5. Na obrázkoch 2.4 je možné vidieť ukážku výstupu detektora.

Obr. 2.4: Ukážka výstupu – detegovaných kuboidov. Žltou sú reprezentované ground-truth kuboidy, zelenou korektné detekcie a červenou chybné detekcie. Prevzaté z pôvodnej práce [28].

2.3.2 Detektor kuboidov využívajúci konvolučnú neurónovú sieť

Novšia práca s názvom *Deep Cuboid Detection: Beyond 2D Bounding Boxes* [8] predstavuje modernejší prístup k detekcii kuboidov založený na konvolučnej neurónovej sieti. Detektor využíva vlastnú CNN architektúru, ktorej časť na extrakciu príznakov (back-bone) tvorí sieť architektúry VGG16 [23]. Mapy príznakov sú potom spracúvané vo viacerých krokoch.

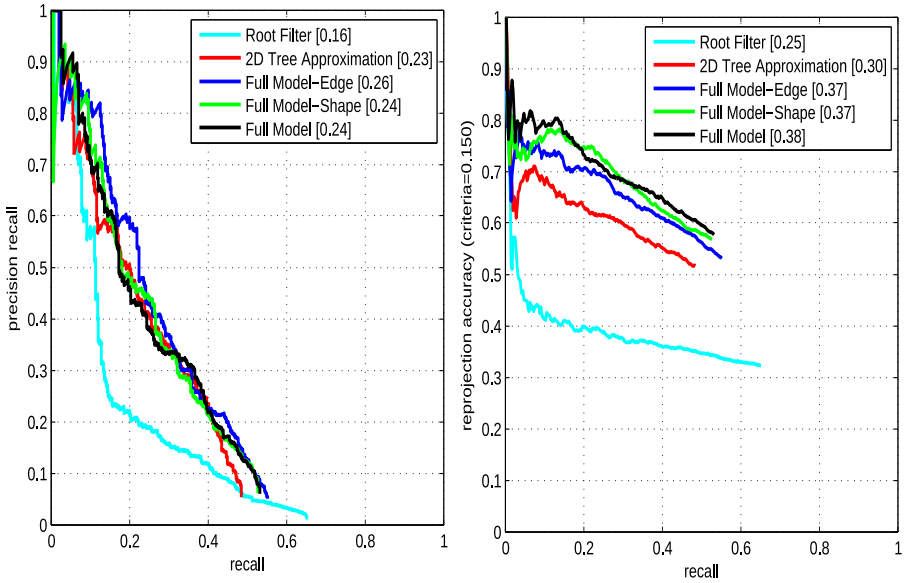
Najprv ich spracúva sieť na určenie regiónov záujmu (region proposal network), ktorá určí oblastí s pravdepodobným výskytom kuboidov. Následne sú príznaky pre každú oblasť vybrané z konvolučnej mapy príznakov a predané dvom plne prepojeným vrstvám, podobne ako pri klasifikácii regiónov v architektúre Faster R-CNN [22]. Rozdielom je však to, že okrem bounding boxu sieť predikuje taktiež normalizovaný offset rohov kuboidu voči stredu regiónu.

S cieľom spresniť predikcie je využívaný takzvaný iteratívny pooling príznakov. Ten spočíva vo viacnásobnej regresii bounding boxov pre presnejšie určenie oblasti, v ktorej sa nachádza objekt. V každom kroku sú z konvolučnej mapy príznakov vybrané príznaky na základe novo zvolenej oblasti, ktoré sú následne opäť predané na vstup plne prepojených vrstiev pre získanie nových predikcii.

V obrázku 2.6 sú zobrazené hodnoty metriky PCK (kap. 2.4.1) spolu s precision-recall krivkou vyhodnotené autormi systému. Z hľadiska rýchlosti systém dosahuje hodnoty 14 fps na grafickej karte Nvidia Titan Z. Príklady výstupov systému je možné vidieť na obrázku 2.7. Na obrázku je taktiež možné pozorovať efekt iteratívneho poolingu príznakov, keďže zobrazuje výstupy po prvej a druhej iterácii.

2.4 Vyhodnocovanie výkonnosti

Pre súčasnosť je typický rýchly vývoj v oblasti počítačového videnia, ktorý je možné pozorovať na stále sa lepšej a lepšej výkonnosti state of the art metód naprieč všetkými oblasťami. Veľmi dôležitým procesom pre určovanie smeru ďalšieho vývoja je vyhodnocovanie výkonnosti. Nakoľko nieje možné vopred odhadnúť, či novo navrhovaný prístup skutočne prinesie zlepšenie, je potrebné tento predpoklad overiť experimentálne. Aby bolo možné jednotlivé metódy objektívne porovnať, je nutné zaviesť dobre špecifikované metriky vyjadrujúce výkonnosť systémov v sledovaných aspektoch. Tie spoločne s verejnými dátovými sadami, ktoré sú v jednotlivých odvetviach štandardom pre vyhodnocovanie umožňujú dosiahnuť



Obr. 2.5: Na ľavej strane vyhodnotenie lokalizácie kuboidov ako celku na základe bounding boxov. Na pravej strane vyhodnotenie lokalizácie rohov pri správne detegovaných kuboidoch na základe vzdialenosti od ground-truth (15 % z druhej odmocniny plochy ground-truth bounding boxu). Prevzaté z pôvodnej práce [28].

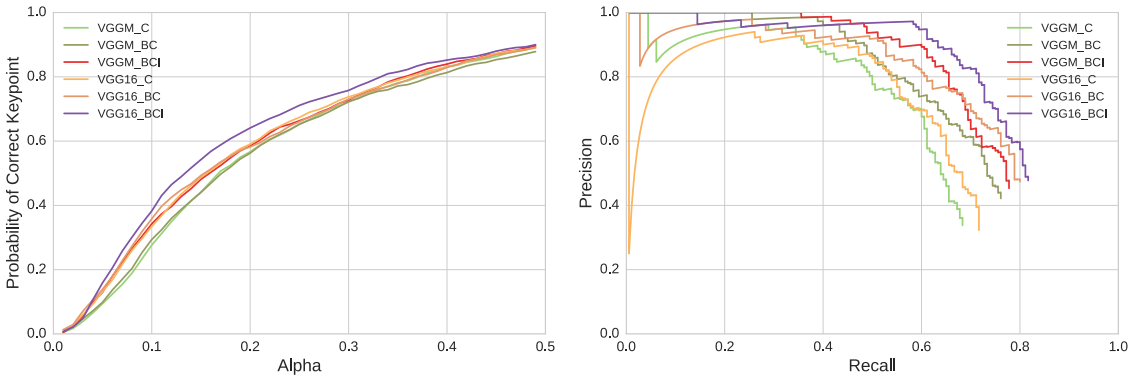
čo možno najobjektívnejšie porovnanie, a teda aj zhodnotenie prínosu novo navrhovaných metód. V nasledujúcich sekciách sa budem venovať metrikám, ktoré sú štandardne využívané na vyhodnocovanie detekcie objektov a lokalizácie kľúčových bodov, ktoré boli taktiež použité v rámci vyhodnocovania detektoru krabíc navrhnutého v rámci tejto práce.

2.4.1 PCK (Probability of correct keypoint)

PCK [16] je metrika určená na vyhodnocovanie systémov na detekciu kľúčových bodov, ktorá je využívaná najmä v oblasti odhadu pózy ľudskej postavy. Ako plynie z názvu, táto metrika vyjadruje podiel správne detegovaných kľúčových bodov (niekedy vyjadrený percentuálne). To, či je daný kľúčový bod detegovaný korektne, sa určí na základe jeho vzdialenosti k ground-truth kľúčovému bodu. Tá sa následne porovná s určeným prahom. V prípade, že je vzdialenosť menšia ako prah, je detekcia považovaná za korektnú, inak za nekorektnú. Hodnota tohto prahu môže byť určená absolútne, alebo relatívne vzhľadom k veľkosti detekovaného kľúčového bodu. V oblasti odhadu pózy ľudskej postavy sa najčastejšie používa verzia označovaná ako PCKh, ktorá využíva relatívne určenie prahu vzhľadom k veľkosti hlavy príslušnej postavy. Samotná hodnota sa po určení správnosti jednotlivých detekcii spočíta podľa vzorca

$$PCK = \frac{|P_c|}{|P_a|}, \tag{2.1}$$

kde P_c označuje množinu správne detegovaných kľúčových bodov a P_a označuje množinu všetkých detegovaných kľúčových bodov.



Obr. 2.6: Hodnoty metriky PCK v závislosti na relatívnej vzdialenosti α. Grafy boli prevzaté z pôvodnej práce [8].

2.4.2 Recall

Recall [20] alebo aj senzitivita je metrika, ktorá vyjadruje pomer počtu korektne predpovedaným pozitívnych prípadov voči počtu reálnych pozitívnych prípadov. V prípade detekcie objektov recall predstavuje akú časť hľadaných objektov detektor korektne detegoval. V ideálnom prípade, kedy detektor identifikoval všetky hľadané objekty, je hodnota recallu rovná 1. To však samé o sebe veľa nevypovedá o výkonnosti detektora, nakoľko táto metrika nezohľadňuje podiel falošných detekcii, a teda sa využíva v kombinácii s ďalšími metrikami, ktoré tento aspekt zohľadňujú. To je najčastejšie presnosť (precission), alebo pri lokalizácii kľúčových bodov aj spomínaná metrika PCK. Grafické znázornenie výpočtu recall-u je zobrazené na obrázku 2.8.

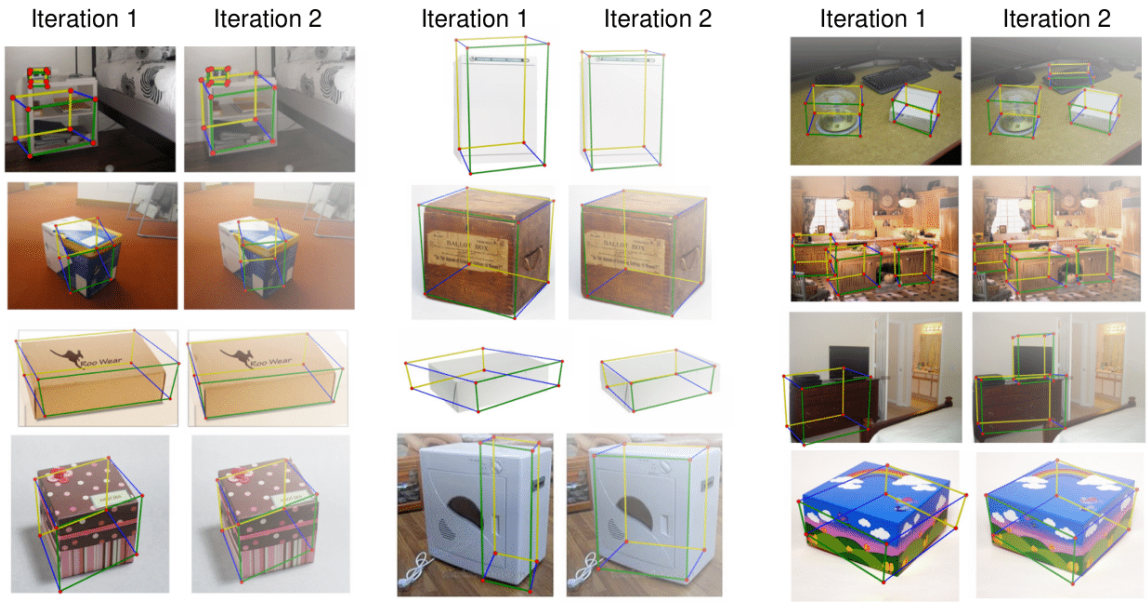
V prípade vyhodnocovania klasifikátorov je jednoduché priamo určiť, kedy bol daný výskyt úspešne predikovaný. Pre účely vyhodnocovania lokalizáciu kľúčových bodov je však potrebné definovať kritérium, ktoré určuje kedy je predikovaná pozícia považovaná za správnu vzhľadom k reálnej pozícii kľúčového bodu. To je možné určiť na základe relatívnej vzdialenosti predikovanej a reálnej pozície tak, ako je popísané v kapitole 2.4.1.

2.4.3 IoU (Intersect over union)

IoU je metrika využívaná v detekcii objektov na určenie podobnosti bounding boxov. Jej hodnota popisuje ako dobre predikovaný bounding box pokrýva ground-truth bounding box. Táto metrika má využitie či už pri vyhodnocovaní, kedy slúži na určenie správnosti predikcie, alebo taktiež v rámci tréningu detektora, kedy môže byť použitá taktiež ako chybová funkcia [32].

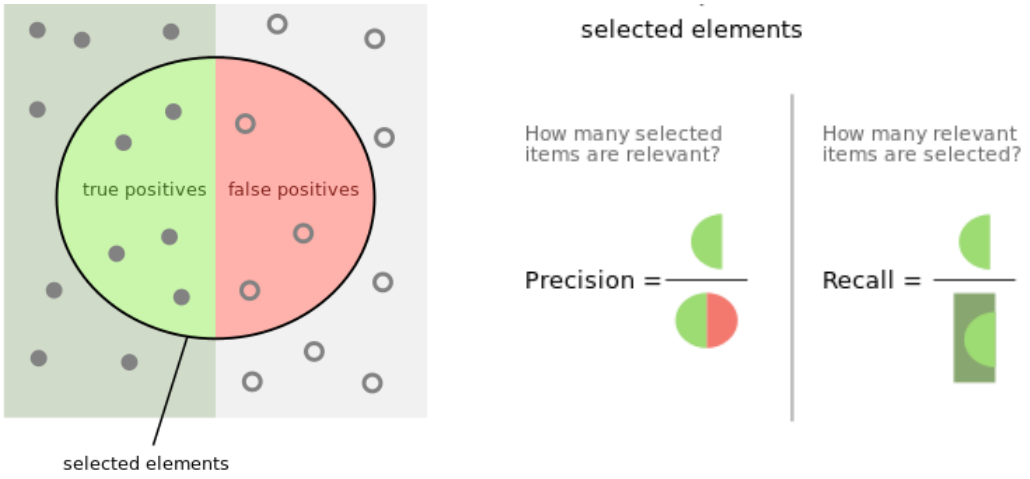
V prípade použitia v rámci vyhodnotenia výkonnosti detektoru sa najprv spočíta hodnota IoU medzi ground-truth a predikovaným bounding boxom. Následne sa táto hodnota porovná s pevne stanovenou prahovou hodnotou (často používaná je hodnota 0.5) za účelom vyhodnotiť správnosť detekcie. Pokiaľ je hodnota väčšia ako stanovený prah, je predikcia považovaná za korektnú, v opačnom prípade za chybnú. Grafické znázornenie výpočtu a konkrétne príklady s uvedenými hodnotami IoU pre lepšiu predstavu možno vidieť v obrázku 2.9.

[5] https://en.wikipedia.org/wiki/Precision_and_recall

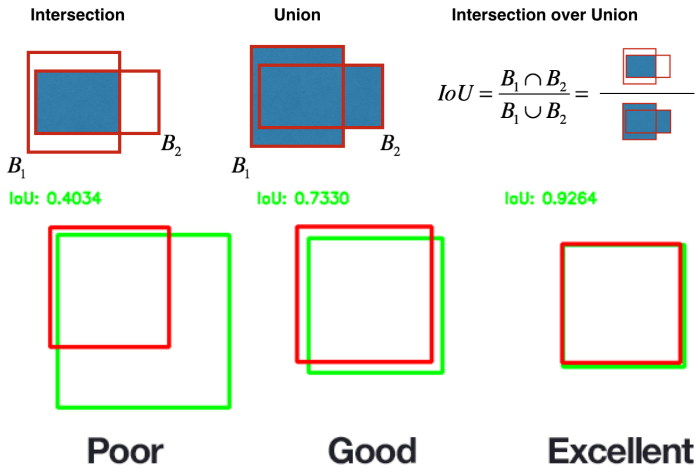


Obr. 2.7: Príklady výstupov systému. Na obrázku sú zobrazené predikcie jednotlivých kuboidov po prvej a druhej iterácii poolingu príznakov. Obrázok bol prevzatý z pôvodnej práce [8].

[7] https://www.oreilly.com/library/view/hands-on-convolutional-neural/9781789130331/a0267a8a-bd4a-452a-9e5a-8b276d7787a0.xhtml



Obr. 2.8: Grafické znázornenie výpočtu recallu. Prevzaté zo zdroja[5].



Obr. 2.9: Grafické znázornenie výpočtu IoU a ukážky ako vyzerá správne a nesprávne predikovaný bounding box. Pre jednotlivé prípady sú pre predstavu uvedené hodnoty IoU. Prevzaté zo zdroja[7].

Kapitola 3

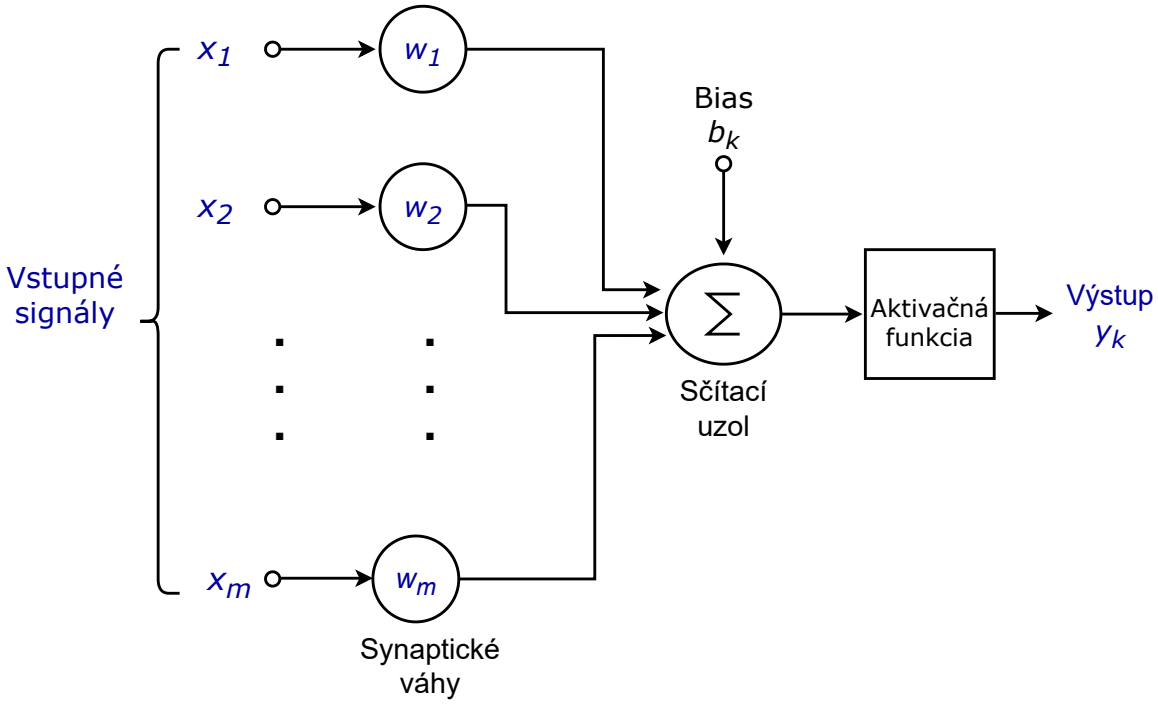
Umelé neurónové siete a ich použitie pri detekcií a lokalizácií objektov

Neurónová sieť [27] je systém na spracovanie dát, ktorý sa skladá z veľkého množstva jednoduchých spracujúcich elementov – neurónov. Tieto spoločne tvoria systém, ktorého architektúra je inšpirovaná cerebrálnym kortexom mozgu. Vďaka tomu sú neurónové siete schopné riešiť problémy, ktoré sú pre ľudí jednoduché, ale na druhej strane komplikované na algoritmizáciu, a teda komplikované pre číslicový počítač. Príkladom takýchto činností sú spracovanie ručne písaného textu, prevod reči na text, klasifikácia a detekcia objektov v obraze a mnohé iné. Z tohto dôvodu sú umelé neurónové siete v posledných rokoch stále viac a viac používané v aplikáciach na veľké množstvo problémov reálneho sveta, a preto sú predmetom rozsiahleho výskumu a vývoja.

3.1 Princíp neurónovej siete

Pred tým ako budeme môcť popísať princíp fungovania umelých neurónových sietí [29], musíme sa najprv stručne pozrieť na ľudským mozog. Ten sa skladá zo stoviek miliard malých výpočetných jednotiek zvaných neuróny, ktoré pracujú paralelne a informácie si vzájomne predávajú pomocou prepojení zvaných synapsie. Neuróny sčítavajú impulzy na ich vstupoch–synapsiach a ak je výsledná hodnota vyššia ako akčný potenciál, vyšlú impulz do ďalšej úrovne. Umelé neurónové siete modelujú toto správanie pomocou umelých neurónov, ktoré sú navzájom prepojené pomocou tzv. váhových spojení. Umelý neurón spočíta váženú sumu všetkých vstupov a určí výstupnú hodnotu pomocou aktivačnej funkcie.

Ako je vidieť v blokovom diagrame na obrázku 3.1, model neurónu sa skladá z troch hlavných komponentov. Prvým sú synapsie zvané aj prepojenia, kde každému prepojeniu prislúcha určitá váha. Táto váha určuje ako sa bude podieľať daný vstup na výstupe nerónu a to tak, že vstup je prenásobený váhou, a následne privedený do ďalšej komponenty, ktorou je sčítačka. Váhy sú upravované v procese tréningu–učenia siete. Sčítačka sčíta všetky vstupy váhované príslušnými váhami a jej výstup je následne pripojený na vstup aktivačnej funkcie. Pred tým je k výstup sčítačky ešte pripočítaná hodnota zvaná bias, ktorej úloha je zvýšiť (alebo v prípade zápornej hodnoty znížiť) hodnotu vstupu aktivačnej funkcie. Úlohou aktivačnej funkcie je obmedziť hodnotu na výstupe na konečný rozsah a jej funkčná hodnota predstavuje finálny výstup daného neurónu.



Obr. 3.1: Bloková schéma umelého neurónu so všetkými jeho časťami. V schéme je znázornená postupnosť transformácie vstupu na výstup. Hodnoty vstupov sú najprv vynásobené váhami a následne privedené do sčítačky, potom sa k sume pričíta hodnota bias. Výsledná hodnota je vstupom aktivačnej funkcie, ktorá má za úlohu hodnotu transformovať na finálny výstup tak, že ju namapuje do určitého intervalu.

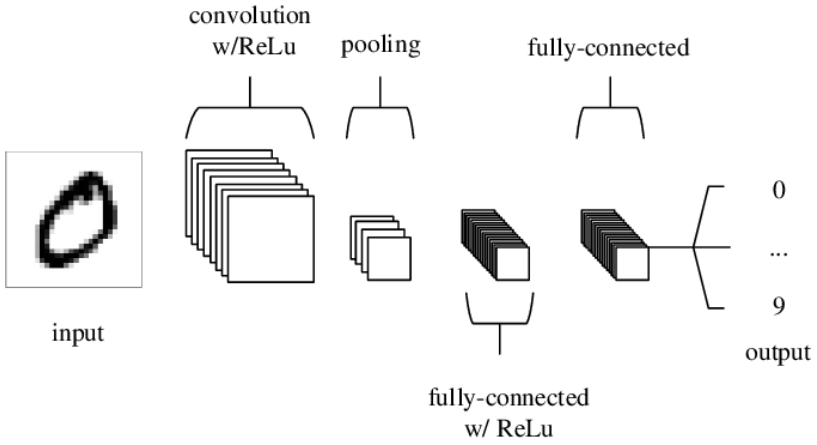
3.2 Konvolučné neurónové siete

Konvolučné neurónové siete (CNN) [17] sú jedným z typov architektúr umelých neurónových sietí. Ich oblasť určenia sú najmä úlohy z oblastí spracovanie obrazu a rozpoznávanie vzorov.

Hlavnou nevýhodou tradičných umelých neurónových sietí je že pri spracovaní vysoko dimenzionálnych dát ,akými sú napríklad obrázky, narážajú na problém vysokej výpočetnej a pamäťovej zložitosti. To je spôsobené obrovským množstvom váh kvôli veľkému počtu jednotlivých vstupných hodnôt a plnému prepojeniu jednotlivých vrstiev. Pre predstavu jediný neurón v prvej vrstve klasickej neurónovej sieti má pre vstup v podobe obrázka rozmerov 64×64 s tromi kanálmi (RGB) 12288 váh. Každé ďalšie zdvojnásobenie rozmeru vstupu vedia na štvornásobný počet váh. Tento problém ešte zvýrazňuje fakt, že spracovanie obrazu je veľmi komplexný problém, a teda je nutné využívať pomerne veľký počet neurónov v niekoľkých vrstvách.

Architektúra konvolučných neurónových sietí je špecializovaná práve pre vstup ako sú obrázky, vďaka čomu bola dosiahnutá redukcia počtu potrebných parametrov modelu. Táto redukcia počtu parametrov vedie nielen k zníženiu výpočetnej zložitosti, ale taktiež k zlepšeniu generalizácii modelu, a teda aj k lepšej presnosti na nevidených dátach. Sieť s menším počtom parametrov je totiž menej náchylná k pretrénovaniu (overfitting), ktoré spravidla vedie k zhoršeným generalizačným schopnostiam.

Ako bolo spomenuté, architektúra CNN je sústredená na optimalizáciu pre vstup v podobe obrázku. Jeden z hlavných rozdielom oproti štandardným umelým neurónovým sieťam



Obr. 3.2: Schéma jednoduchej neurónovej siete určenej na klasifikáciu ručne písaných číslic z dátovej sady MINST. Sieť sa skladá z piatich vrstiev. Konkrétne v tomto poradí vstupná vrstva, konvolučná vrstva pre extrakciu príznakov, pooling vrstva na redukciu dimenzionality a plne prepojené vrstvy s ReLu aktivačnou funkciou pre finálnu predikciu skóre jednotlivých číslic na základe príznakov extrahovaných predošlými vrstvami. Obrázok prevzatý zo zdroja [17].

je fakt, že vrstvy CNN sú tvorené neurónmi organizovanými do troch dimenzii. To odpovedá dimenzionalite vstupu – obrázka (výška × šírka × hĺbka). Hĺbka v tomto pojatí neodpovedá počtu vrstiev siete, ale tretiemu rozmeru aktivačného priestoru. Na rozdiel od štandardných neurónových sietí sú neuróny ľubovolnej vrstvy prepojené iba s malou oblasťou predchádzajúcej vrstvy, čo vedie k želanej redukcii parametrov.

3.2.1 Architektúra konvolučných neurónových sietí

Konvolučná neurónové siete sú tvorené tromi hlavnými typmi vrstiev, menovite konvolučná vrstva, pooling vrstva a plne prepojená vrstva (fully-connected layer). Rôzne kombinácie týchto vrstiev spojené jedna za druhov potom tvoria konvolučnú neurónovú sieť. Na obrázku 3.2 je ukážka architektúry veľmi jednoduchej konvolučnej neurónovej siete určenej na klasifikáciu ručne písaných číslic z dátovej sady MINST [12].

3.2.2 Konvolučná vrstva

Konvolučná vrstva je, ako jej názov napovedá, kľúčová pre samotný koncept fungovania konvolučných neurónových sietí. Táto vrstva sa skladá zo súboru naučiteľných konvolučných jadier,zvaných aj filtre, ktoré sú relatívne malých rozmerov, ale siahajú cez celú hĺbku vstupu. Váhy tejto vrstvy predstavujú koeficienty týchto filtrov. Vrstva spracováva dáta na vstupe tak, že spočíta konvolúciu každého konvolučného jadra cez celú priestorovú dimenzionalitu vstupu, a vyprodukuje tak 2D aktivačnú mapu. Tú je možné vizualizovať vo forme tepelnej mapy. Konkrétny príklad je zobrazený na obrázku 3.4.

Výpočet konvolúcie s konvolučným jadrom spočíva vo výpočte sumy vstupu váhovaného príslušným koeficientom konvolučného jadra pre každú jeho pozíciu tak, ako je to ukázané v obrázku 3.3. V procese učenia sú tieto koeficienty konvolučného jadra upravované tak, aby reagovali na špecifický príznak na danej pozícii. Každé konvolučné jadro produkuje samostatnú aktivačnú mapu, a tie sú naskladané nad seba, aby vytvorili finálny výstup

konvolučnej vrstvy. Výstupom konvolučnej vrstvy je teda toľko aktivačných máp, koľko konvolučných jadier (filtrov) obsahuje.

Ako už bolo spomenuté, pri klasických neurónových sieťach vniká pri obrazovom vstupe problém s veľkým počtom parametrov. Tento problém je zapríčinený plným prepojením neurónov. Konvolučné neurónové siete riešia tento problém tak, že obmedzujú toto prepojenie neurónu len na malú oblasť vstupu, ktorá sa nazýva receptive field. Tá môže nadobúdať rozličné rozmery.

Pre spomínaný príklad obrázka s rozmermi 64×64 to vyzerá nasledovne. Pri použití receptive field rozmeru 6×6 s plným prepojení v smere hĺbky sa dostávame na 108 parametrov ($6 \times 6 \times 3$), čo je významný úbytok voči plnému prepojeniu pri štandardnej architektúre neurónovej siete, ktoré viedlo k 12188 váham pre každý neurón.

Ďalšiu optimalizáciu dosahujú CNN pomocou optimalizácie dimenzionality výstupu. Dimenzionalita výstupu konvolučnej vrstvy je závislá na hyper-parametroch, ktorými sú hĺbka, stride (krok) a spôsob dopĺňania hodnôt vstupu (padding) pre výpočet konvolúcie na okrajoch vstupu.

Hĺbka

Parameter hĺbka alebo aj počet filtrov slúži na špecifikovanie hĺbky výstupu konvolučnej vrstvy. Z pohľadu architektúry predstavuje táto hodnota počet konvolučných jadier (filtrov) danej vrstvy. Ako bolo spomenuté, pre každý filter je na výstupe produkovaná jedna aktivačná mapa, ktorá zachytáva určitý príznak. Redukcia hĺbky vedie k významnému zníženiu počtu parametrov, avšak aj k zníženiu schopnosti modelu rozlišovať komplikovanejšie vzory.

Stride

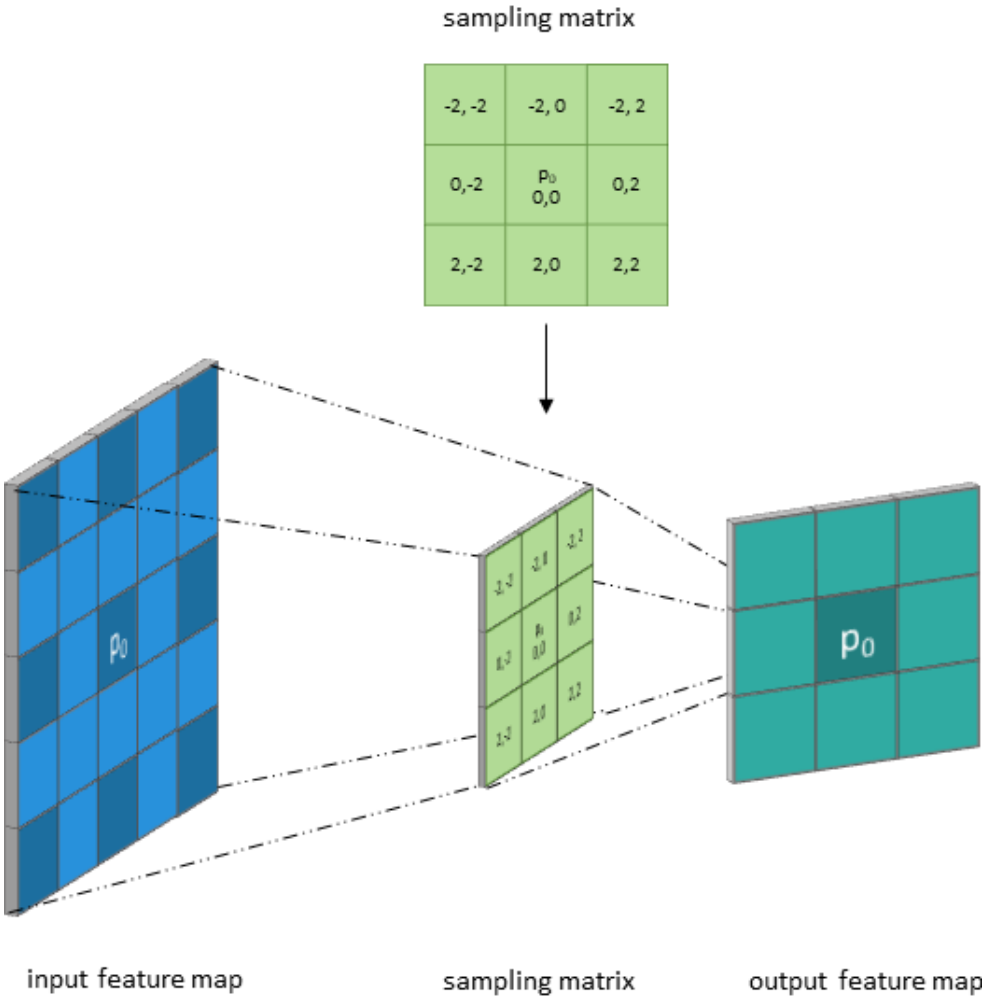
Tento hyper-parameter vyjadruje krok posunu konvolučného jadra po priestore vstupu. Nízke hodnoty kroku vedú k veľkému prekryvu receptive field, a teda produkujú veľké aktivačné mapy. Veľké hodnoty naopak redukujú prekryv a teda aj rozmery aktivačných máp.

Padding

Padding rieši prípad výpočtu konvolúcie pre okrajové pixely. Keďže konvolučné jadro sa posúva po vstupe, na okraji nastáva situácia, kedy časť konvolučného jadra siaha mimo vstup. Aby bolo možné konkrétne spočítať konvolúciu pre tento prípad, je potrebné hodnoty vstupu na pozíciach, kde konvolučné jadro vyčnieva za hranicu vstupu, doplniť nulami. Toto zamedzí strate informácii na okraji vstupu, a taktiež redukuje stratu rozlíšenia na výstupe.

3.2.3 Pooling vrstva

Účelom pooling vrstvy je redukcia dimenzionality vstupu, a teda aj ďalšia redukcia počtu parametrov. Pracuje nad všetkými aktivačnými mapami na vstupe, teda v celej hĺbke vstupu. Samotná redukcia dimenzionality prebieha najčastejšie s využitím funkcie maximum. Tento druh poolingu sa preto nazýva ako max-pooling. Okrem max-poolingu sa často používa aj tzv. average-pooling, ktorý ako názov napovedá, využíva výpočet aritmetického priemeru. Parametrami pooling vrstvy sú veľkosť filtru a krok posunu po vstupe (stride). Pri max-poolingu je pre jednotlivé pozíciu filtra na výstup privedené maximum hodnôt v oblasti pod filtrom. Najčastejšie sú používané pooling vrstvy s rozmermi filtra

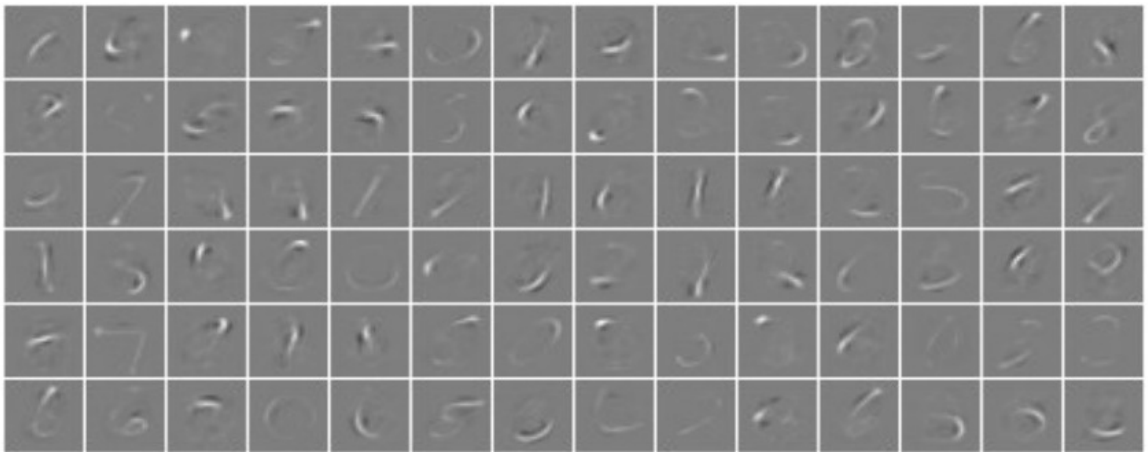


Obr. 3.3: Grafické znázornenie výpočtu konvolúcie vstupu a filtra, kde p_0 označuje pozíciu stredu konvolučného jadra na vstupe. Hodnota výstupu pre danú pozíciu sa spočíta ako vážený priemer hodnôt vstupu pod filtrom, kde váhy predstavujú koeficienty filtra na danej pozícii. Obrázok bol prevzatý zo zdroja[2].

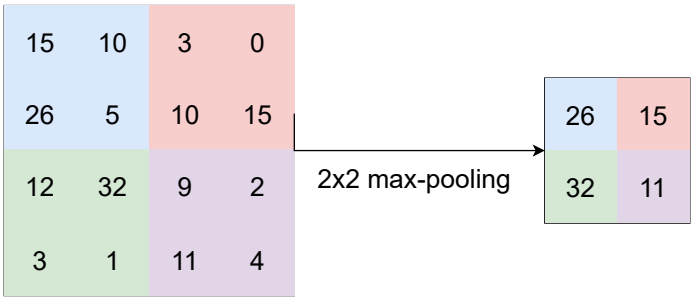
2×2 a krokom 2. To vedie na redukciu šírky a výšky vstupu na polovičnú, pričom hĺbka zostane zachovaná. Alternatívne sa používa tzv. prekrývajúci sa pooling s veľkosťou filtra 3×3 a krokom 2. Použitie veľkosti filtra väčšej ako 3 vedie z dôvodu stratovému charakteru tejto vrstvy zvyčajne k zhoršenej výkonnosti modelu. Ukážku transformácie vstupu pooling vrstvou ukazuje obrázok 3.5.

3.2.4 Batch Normalizácia

Batch normalizácia [10] je technika normalizácie aktivácii medzi jednotlivými vrstvami hlbokých neurónových sietí. S cieľom redukcie výpočetnej náročnosti sa však táto normalizácia vykonávaná vrámci menších podmnožín trénovacích dát (mini-batches). Táto technika umožňuje využívanie vyšších hodnôt rýchlosti učenia (learning rate), čím zrýchľuje proces tréningu siete. Okrem toho taktiež zlepšuje schopnosť generalizácie modelu. V súčastnosti



Obr. 3.4: Vizualizované aktivácie prvej konvolučnej vrstvy zo spomínanej jednoduchej neurónovej siete po natrénovaní na dátovej sade ručne písaných číslic MINST. Obrázok bol prevzatý z [17].



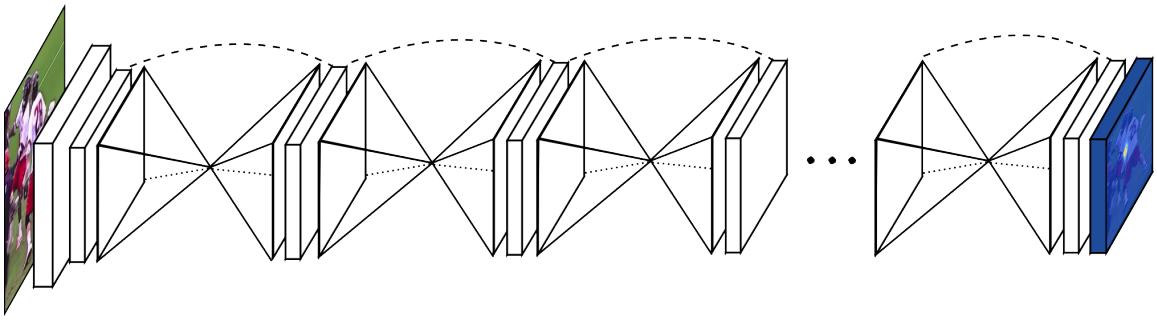
Obr. 3.5: Schematická ukážka vstupu a výstupu max pooling vrstvy. V tomto prípade bola použitá max-pooling vrstva s veľkosťou filtra 2×2 a krokom (stride) rovným hodnote 2. Na obrázku je možné pozorovať, ako aplikácia tejto vrstvy na vstup s rozmermi 4×4 viedla na redukciu rozmeru na 2×2.

je táto technika hojne využívaná v rámci rôznych typov neurónových sietí vrátane konvolučných.

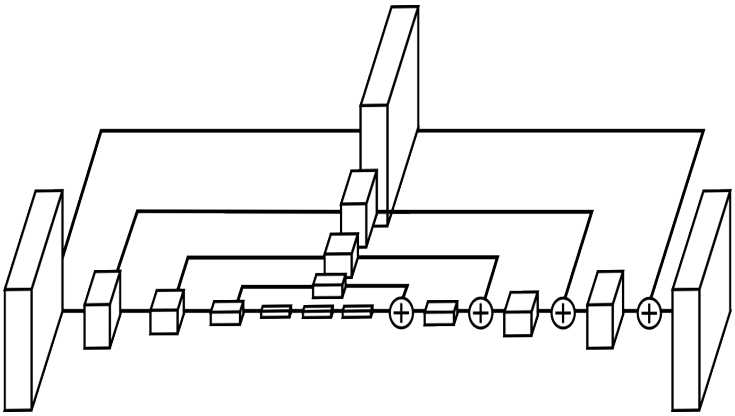
3.3 Stacked Hourglass Network

Stacked Hourglass Network [18] je architektúra konvolučnej neurónovej siete určená na regresiu kľúčových bodov vo forme skóre pre každý pixel (tzv. heatmap regression). Táto architektúra bola pôvodne navrhnutá pre použitie v oblasti odhadu pózy ľudskej postavy, kde je využívaná na detekciu a lokalizáciu jednotlivých kĺbov.

Ako je vidieť na schéme v obrázku 3.6, sieť pozostáva z niekoľkých takzvaných Hourglass modulov, ktoré sú naskladané jeden za druhým. Hourglass modul má tvar presýpacích hodín, a funguje na princípe encoder-decoder bloku s využitím tzv. skip connections (prepojenia s vynechaním niektorých vrstiev). Účelom vynechania vrstiev je zachovať priestorové informácie vo viacerých rozlíšeniach. Spojenie viacerých Hourglass modulov za seba tak, že výstup predošlého modulu je vstupom nasledujúceho, spolu s výpočtom chyby na kaž-



Obr. 3.6: Schéma architektúry Stacked Hourglass Network. Sieť je tvorená niekoľkými za sebou zreťazenými Hourglass modulmi. Obrázok prevzatý zo zdroja [18].

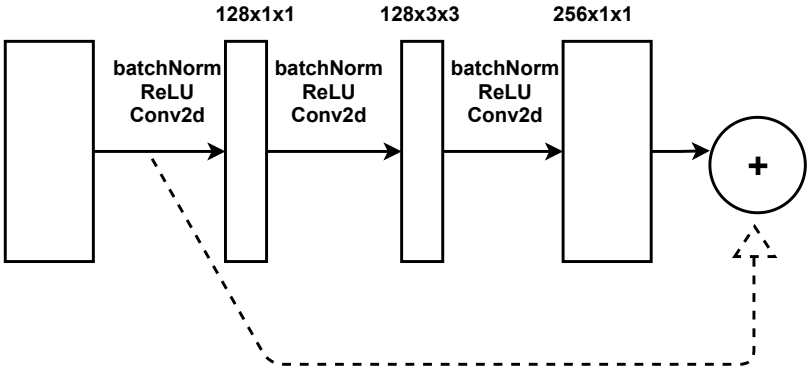


Obr. 3.7: Zjednodušená schéma jedného Hourglass modulu. Každý blok predstavuje jeden reziduálny blok popísaný v sekcii 3.3.1. Na obrázku je možné vidieť taktiež aj tzv. skip connections, ktoré zabezpečujú zachovanie informácii v rôznych rozlíšeniach tak, že zachovávajú mapy príznakov z jednotlivých krokov enkódovania, a tie sa následne sčítajú s mapami príslušných rozmerov pri procese dekódovania.Obrázok prevzatý zo zdroja [18].

dom výstupe umožňujú sieti iteratívne prehodnocovať priestorové vzťahy na vyššej úrovni, a zlepšovať tým presnosť finálnych predikcii.

3.3.1 Architektúra Hourglass modulu

Ako bolo spomenuté, Hourglass modul má tvar presýpacích hodín. Pri zachovaní tohto tvaru však existuje viacero alternatív vo voľbe použitých vrstiev. Autori sa na základe experimentov rozhodli pre použitie tzv. reziduálnych učiacich modulov. Na obrázku 3.7 je zobrazená schéma Hourglass modulu, kde každý blok reprezentuje jeden reziduálny modul. Reziduálny modul sa skladá z 2 konvolučných vrstiev s filtrami veľkosti 1×1 a jednej s filtrami 3×3. Okrem toho využíva Batch normalizáciu medzi jednotlivými konvolučnými vrstvami. Ako aktivačná funkcia je využitá ReLu. Schéma reziduálneho modulu je zobrazená na obrázku 3.8.



Obr. 3.8: Schéma reziduálneho modulu. Čísla nad blokmi popisujú parametre konvolučnej vrstvy (počet filtrov × rozmer filtra × krok)

3.3.2 Koncept *Intermediate supervision*

Koncept priameho dozoru (Intermediate supervision) spočíva vo viacnásobnom vyhodnocovaní chybovej funkcie pre takzvané pomocné predikcie. Tieto pomocné predikcie sú produkované na výstupe každého Hourglass modulu, pričom ako finálna predikcia sa potom použije výstup z posledného bloku. Keďže pozície jednotlivých kĺbov pri odhade ľudskej pózy sú vzájomne závislé, musí sieť vnímať jednotlivé príznaky nielen v lokálnom, ale aj v globálnom kontexte. Produkcia pomocných predikcii núti sieť udržiavať si prehľad o týchto globálnych vzťahoch už v skorších fázach. Opakované prehodnocovanie jednotlivými Hourglass modulmi umožňuje tieto globálne príznaky lepšie vyhodnotiť a korigovať na ich základe prvotné predikcie. Z tohto dôvodu je princíp iteratívneho vyhodnocovania v rôznych podobách využívaný v modeloch na odhad pózy ľudskej postavy, ale hodí sa všade tam, kde existujú významné priestorové vzťahy detegovaných kľúčových bodov.

Kapitola 4

Návrh

4.1 Voľba prístupu k detekcii a lokalizácii krabice

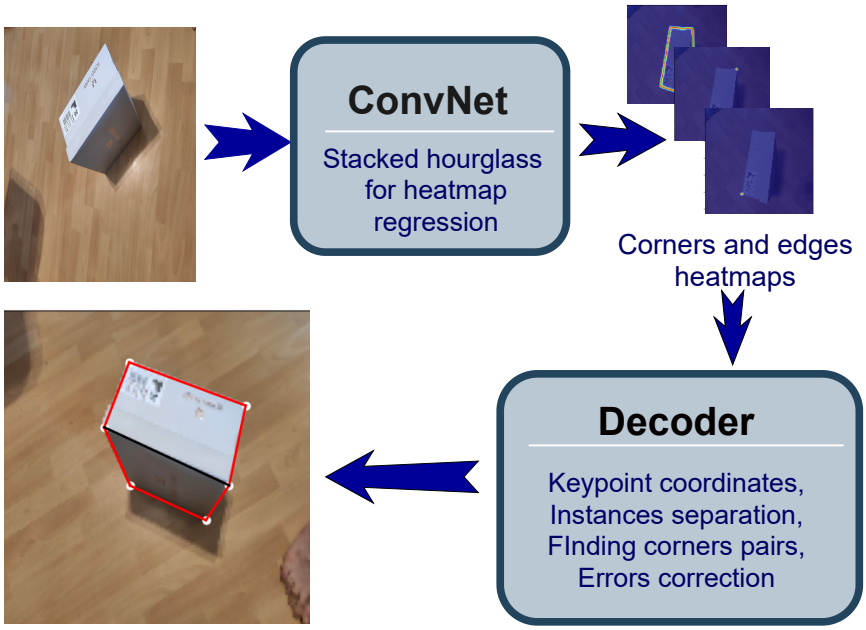
Ako bolo načrtnuté v kapitole 2, existuje viacero prístupov k detekcii a lokalizácii objektov. Táto práca sa zaoberá detekciu krabíc z pohľadu detekcie objektov tvaru kvádra, pri čom využíva detekciu a lokalizáciu kľúčových bodov. Jednotlivé kľúčové body v tomto prípade predstavujú časti krabice ako sú rohy a hrany (hranové body), ktoré sú typické pre geometrický tvar krabice – kváder. Výhodou tohto prístupu je fakt, že umožňuje presnejší popis pozície detekovaného objektu ako je tomu v prípade detekcie objektu ako celku s výstupom v podobe osovo zarovnaného bounding boxu. Táto reprezentácia totiž nieje dostatočná na to, aby popísala orientáciu objektu v priestore a informácia o rozmeroch objektu je taktiež iba čiastočná. Z pohľadu použitej technológie detekcie kľúčových bodov práca využíva nepriamu regresiu kľúčových bodov s využitím konvolučných neurónových sietí.

Ako inšpirácia pre tento prístup dobre poslúžila oblasť odhadu pózy ľudskej postavy, keďže je tento problém v niektorých aspektoch podobný. Podobne, ako sú pri odhade pózy ľudskej postavy lokalizované jednotlivé kĺby, ktoré spoločne popisujú pozíciu a priestorovú orientáciu torza ako celku, tento systém využíva detekciu rohov a hranových bodov s cieľom popísať pozíciu krabice. Analogicky k odlišnosti jednotlivých kĺbov na tele je možné rozlíšiť niekoľko typov rohov a hrán na základe definovaných kritérií. Napríklad to môže byť rozličný počet viditeľných stien, ktoré sa podieľajú na danom rohu, rozlíšenie obrysovej a vnútornej hrany a podobne. Konkrétne rozdelenie hrán a rohov na jednotlivé triedy je popísané v kapitole 4.3.1.

4.2 Návrh systému

Navrhovaný systém má za úlohu detegovať inštancie krabíc v štandardnom RGB obrázku. Keďže sa jedná o komplexnú úlohu, spracovanie prebieha po častiach v niekoľkých krokoch. Na základe toho bol systém rozdelený na viacero častí s cieľom rozdeliť zodpovednosť za jednotlivé kroky. Toto rozdelenie je schematicky znázornené na obrázku 4.1.

Prvým krokom je detekcia hranových bodov a rohov krabíc. Túto úlohu plní konvolučná neurónová sieť, konkrétne sieť s architektúrou Stacked Hourglass (kap. 3.3). Výstupom tohoto kroku sú predikcie výskytu jednotlivých typov rohov a hranových bodov. Tie sú reprezentované formou matice pre každý typ rohu a hrany, ktorá predstavuje tzv. confidence mapu – hodnota na danej pozícii predstavuje skóre výskytu na danej pozícii.



Obr. 4.1: Zjednodušený diagram architektúry systému. Modré bloky symbolizujú jednotlivé hlavné časti detekčnej pipeline. Na obrázkoch sú ukázané príklady vstupov a výstupov jednotlivých častí systému.

Výstup neurónovej siete je potom spracúvaný časťou systému zvanou dekodér s cieľom získať konečný výstup v podobe jednotlivých krabíc popísaných množinami hrán a rohov. Na to je potrebné najprv určiť na základe výstupu CNN – confidence máp súradnice jednotlivých rohov.

Po tom, čo sú známe súradnice jednotlivých hrán a rohov, nasleduje proces zoskupenia na jednotlivé objekty – krabice. Keďže systém využíva prístup k detekcii tzv. zdola hore, nieje dostačujúce jednotlivé rohy jednoducho prepojiť do tvaru kvádra. Najprv je totiž nutné priradiť detekované kľúčové body jednotlivým inštanciám krabíc, ktorých sa mohlo na vstupe vyskytovať ľubovoľné množstvo. Pri tom však treba brať do úvahy taktiež fakt, že informácie o pozíciach hrán a rohov môžu obsahovať chyby. Presnejšie môžu nastať dva prípady, a to chýbajúce body, ale aj falošné detekcie v miestach, kde sa roh/hrana v skutočnosti nenachádza. Ďalšou úlohou dekodéra je teda snaha identifikovať tieto situácie, a ak je to možné ich aj korigovať, aby výstup čo najlepšie odpovedal realite.

4.3 Dátová sada

Keďže súčasťou systému je konvolučná neurónová sieť, je potrebné zaobstarať dostatočne veľkú dátovú sadu kartónových krabíc pre jej tréning. Väčšionou je v podobných prípadoch najvýhodnejšie siahnuť po už existujúcej verejne dostupnej dátovej sade, a to ako z dôvodu šetrenia času potrebného pre jej získanie, tak pre možnosť priameho objektívneho porovnania s existujúcimi riešeniami. Kvôli špecializácii tohto systému však táto možnosť odpadá, nakoľko verejne dostupná dátová sada krabíc doposiaľ neexistuje. Preto je nutné ako súčasť tejto práce túto dátovú sadu vytvoriť.

To spočíva v získaní vhodných fotografii krabíc a následnej anotácii. Pre účely tejto práce je potrebné anotovať pozície rohov spolu z ich triedou a taktiež páry bodov popisujú-

Obr. 4.2: 6 rozličných typov rohov na základe počtu viditeľných stien ktoré ho tvoria a toho či sa nachádza na prednej alebo zadnej strane. Dva rozdielne typy hrán sú odlíšené farebne a to červenou vnútorné hrany a čiernou okrajové hrany.

cich začiatok a koniec hrany spolu s jej typom. Pre proces vyhodnocovania budú anotované taktiež osovo zarovnané obdĺžnikové oblasti (bounding-boxy) pokrývajúce jednotlivé krabice.

Nakoľko je proces tvorby dátovej sady časovo náročný, bude jeho tvorba prebiehať po častiach. Na začiatku bude vytvorená prvotná dátová sada pozostávajúca z menšieho množstva (150-200) anotovaných obrázkov krabíc pre počiatočný vývoj. Tá bude následne priebežne rozširovaná o ďalšie snímky. Podrobnejšie informácie o procese tvorby dátovej sade sú uvedené v kapitole 5.1.

4.3.1 Triedy hrán a rohov

S cieľom zachytiť dodatočné informácie, ktoré môžu byť využité pre presnejšiu identifikáciu orientácie krabice, a prípadne taktiež pre použitie pri korekcii chýb v detekciách, boli hrany a rohy rozdelené do niekoľkých tried. Konkrétne sa jedná o 2 triedy hrán a 6 tried rohov. Hrany sú rozdelené na obrysové a vnútorné. Obrysové hrany sú tie, ktoré tvoria siluetu krabice. Vnútorné sú naopak tie, ktoré sa nachádzajú na obrázku v rámci obrysu, a teda obrys neovplyvňujú. Rohy sa delia podľa počtu viditeľných strán, kde môžu nastať prípady s jednou, dvoma a tromi stranami. Ďalej sa rohy rozdeľujú podla toho, či sa nachádzajú na prednej strane alebo zadnej strane krabice. Tieto 2 kritéria tvoria v kombináciách 6 tried rohov. Reálne príklady jednotlivých typov hrán a rohov sú zobrazené na obrázku 4.2.

4.4 Detekcia hrán a rohov

Detekcia hrán a rohov je v ponímaní tohto systému špecializovaný prípad detekcie kľúčových bodov. Keďže roh tvorí priesečník hrán, teda jeden konkrétny bod, môžeme ho priamo reprezentovať jedným kľúčovým bodom. Menší problém nastáva pri hrane, nakoľko tá je reálne tvorená nekonečným množstvom bodov. V realite je však vstupný obrázok diskrétny a hrana je na ňom vyobrazená na konečnom množstve pixelov. Pre účely tohto systému teda môže byť hrana interpretovaná ako konečnú množinu hranových bodov. Získanie týchto bodov z počiatočného a koncového bodu danej hrany spočíva v rasterizáciou tejto úsečky.

Po tom, čo sme si ujasnili spôsob vyjadrenia hrán a rohov v podobe kľúčových bodov je potrebné zvoliť konkrétny spôsob ako tieto kľúčové body detektovať. Ako je spomenuté v kapitole 2.2, existujú 2 základné prístupy k detekcii kľúčových bodov pomocou konvolučných neurónových sietí, a to je priama regresia(kap. 2.2.1) a tzv. regresia tepelných máp(kap. 2.2.2). Tento systém využíva metódu regresie tepelných máp, a to najmä z dôvodu že sa radí do kategórie detektorov s prístupom zdola hore (bottom-up). Preto je nutné, aby metóda detekcie kľúčových bodov umožňovala detekciu viacerých inštancii daného typu kľúčového bodu, čo táto metóda na rozdiel od priamej regresie spĺňa. Ďalším dôvodom je aj už spomínaná lepšia presnosť tohoto prístupu pozorovaná v oblasti odhadu pózy ľudskej postavy.

4.4.1 Voľba architektúry CNN

Momentálne existuje pomerne veľké množstvo architektúr neurónových sietí určených na detekciu kľúčových bodov pomocou regresie tepelných máp, ktoré pripadajú do úvahy pre využitie v tomto systéme. Preto je nutné zvážiť ktorú z dostupných architektúr použiť. Pri tomto rozhodnutí je potrebné prihliadať na niekoľko aspektov. Prvým z nich je presnosť detekcie. Keďže väčšina z týchto architektúr má svoj pôvod v oblasti odhadu pózy ľudskej postavy môžeme získať prehľad o ich výkonnosti na základe dosiahnutých výsledkov v tejto oblasti. Keďže sa však jedná iba o čiastočne podobný problém, sú tieto informácie o výkonnosti skôr prehľadného charakteru, a je možné, že pre tento problém by sa výsledky jednotlivých architektúr miery líšili. Ďalšími dôležitými rozhodovacími kritériami sú rýchlosť inferencie, a v neposlednom rade aj dostupnosť implementácie modelu.

Na základe hore uvedených kritérií som sa rozhodol použiť architektúru Stacked Hourglass Network, ktorej bližší popis možno nájsť v kapitole 3.3

4.5 Spracovanie výstupu CNN

Výstupom konvolučnej neurónovej siete je niekoľko tzv. confidence máp, jedna pre každý typ kľúčového bodu. Tie popisujú skóre výskytu daného kľúčového bodu na odpovedajúcej pozícii vstupu. Tento výstup je však potrebné ďalej spracovať až do podoby konečného výstupu. Tou je množina detegovaných krabíc, kde každá je popísaná množinou hrán a rohov. Toto spracovanie je úlohou ďalšej časti systému zvanej dekodér. Spracovanie prebieha v niekoľkých krokoch odpovedajúcich jednotlivým problémom, ktoré je potrebné vyriešiť. Tými sú separácia inštancii, prevod z confidence máp na súradnice, identifikácia párov rohov tvoriacich hrany a korekcia prípadných chýb, ktoré vznikli z dôvodu nesprávne detegovaných kľúčových bodov.

4.5.1 Separácia krabíc

Keďže systém je založený na bottom-up prístupe, na vstupe sa môže vyskytnúť viacero inštancii detegovaných objektov bez apriórnej informácie o ich počte a pozícii. Úlohou systému je teda identifikovať jednotlivé inštancie krabíc a každej z nich priradiť príslušnú podmnožinu detegovaných rohov a hrán, ktoré ju tvoria. Pre tento účel systém využíva separáciu na základe obrysov krabíc. Tento krok je možné vykonať ešte pred procesom párovania detegovaných rohov a umožňuje identifikovať prvotných kandidátov na samostatne stojace inštancie. Tie sú potom spracovávané separátne. Problémom je však prípad prekrytia krabíc, ktorý je potrebné odhaliť v rámci procesu párovania rohov.

4.5.2 Získanie súradníc rohov

Miesta s vysokými hodnotami v confidence mapách jednotlivých tried predstavujú pravdepodobné pozície kľúčových bodov. Pre ďalšie spracovanie je však potrebné získať množinu detegovaných rohov popísaných ich súradnicami. Pre top-down prístup je typické považovať za pozíciu kľúčového bodu pozíciu maxima confidence mapy pre danú triedu kľúčových bodov. Pri bottom-up prístupe, ktorý využíva tento systém, je však tento spôsob nevhodný, nakoľko sa môže v detekcii vyskytovať ľubovoľné množstvo kľúčových bodov daného typu. Použitie maxima by v tomto prípade viedlo k extrakcii iba jedného z nich, zatiaľ čo všetky ostatné by boli ignorované. Navrhovaný spôsob spočíva v potlačení miest s nízkym skóre (non-maximum-suppression), potom nasleduje identifikácia izolovaných oblastí s nenulovým skóre a následne nájdenie súradníc odpovedajúcich jednotlivým oblastiam, a teda aj kľúčovým bodom.

4.5.3 Identifikácia párov rohov

V momente, kedy sú známe pozície rohov a hrán krabice, má systém všetky potrebné informácie na zostavenie modelov krabíc. Je však nutné tieto informácie správne interpretovať vo vzájomnej súvislosti. Konkrétne je potrebné identifikovať jednotlivé páry rohov, ktoré tvoria hrany krabice. Pre tento účel je vhodné využiť confidence mapu hranových bodov. Ak daný pár rohov skutočne tvorí hranu, ležia na ich spojnici hranové body, a teda sú na ich spojnici vysoké hodnoty v confidence mape hrán. Na základe počtu nenulových miest v confidence mape okolo tejto spojnice teda môže byť učinené rozhodnutie, či práve skúmaný pár rohov predstavuje hranu alebo nie.

4.5.4 Korekcia chýb

Keďže pri detekcii rohov a hrán mohlo dôjsť k chybám, môžu nastať prípady, kedy výsledné modely detegovaných krabíc nemajú korektný tvar. Príkladom môže byť napríklad absencia jedného z rohov a z toho plynúca chýbajúca hrana či viaceré hrany. Nepresnosť môže nastať aj v procese identifikácie párov rohov, napríklad z dôvodu blízkosti dvoch rohov. V týchto prípadoch je možné využiť fakt, že krabice majú z veľkej väčšiny tvar kvádra. S využitím geometrie je teda možné niektoré z týchto chýb identifikovať a do určitej miery aj korigovať. Konkrétne sa dá pre tieto účely využiť najmä rovnobežnosť a rovnosť rozmerov protiľahlých stien, na základe čoho je možné v niektorých prípadoch odhadnúť pozíciu chýbajúceho rohu. Taktiež je možné využiť fakt, že obrysové hrany krabice musia tvoriť uzavretú komponentu, a teda pokiaľ tomu tak nieje, niektorá z obrysových hrán chýba. Konkrétne prípady a použité algoritmy sú bližšie popísané v kapitole 5.5.4.

Kapitola 5

Realizácia

V tejto kapitole bude detailne popísaný postup realizácie navrhnutého systému na detekciu krabíc. Tá prebiehala vo viacerých krokoch, menovite tvorba dátovej sady, tréningu konvolučnej neurónovej siete pre detekciu rohov a hrán a implementácia algoritmov pre spracovanie výstupu konvolučnej neurónovej siete do podoby konečného výstupu – modelov krabíc. Každý zo spomínaných krokov bude v tejto kapitole podrobne popísaný v príslušnej sekcii.

5.1 Tvorba dátovej sady

Keďže získanie dátovej sady bolo nutnou podmienkou pre vývoj detektora, bolo získanie prvotnej dátovej sady prvým krokom pred začiatkom samotnej implementácie systému. Aby bol systém dostatočne komplexný, je potrebné zaistiť dostatočnú rozmanitosť a to z pohľadu viacerých aspektov. Medzi tieto aspekty patrí veľkosť krabice, farba krabice, typ pozadia, orientácia na snímku, počet krabíc a ďalšie. Prvotná dátová sada bola zložená najmä z jednoduchších prípadov v podobe kontrastnejších pozadí a väčšieho rozmeru krabíc, ktoré sa zväčša vyskytovali samostatne. Úmyslom jednoduchšej dátovej sady bolo najprv overiť funkčnosť samotného konceptu detekcie rohov a hranových bodov na jednoduchých prípadoch. Problematické prípady boli do dátovej sady doplňované postupne v procese jej priebežného rozširovania. Tieto problematické prípady spočívali najmä v menej kontrastnom pozadí, výskyte viacerých inštancií krabíc, ktoré sa prípadne prekrývajú, menšej veľkosti krabice na snímku a tak ďalej.

Prvým krokom pri samotnej tvorbe dátovej sady bolo získanie fotografii krabíc. Snímky boli odfotografované mobilným telefónom v prostredí domácnosti. Fotografie majú rôzne typy pozadí a to ako jednofarebné, tak aj vzorkované ako napríklad koberec, plávajúca podlaha a iné. Na niektorých snímkach sa taktiež v pozadí vyskytujú iné objekty, najmä časti nábytku. Na snímkoch sa nachádzajú rôzne počty krabíc, zväčša 1-2, v niektorých prípadoch aj 3 a 4. Z pohľadu svetelných podmienok boli fotografie zhotovované za umelého aj denného svetla. Po zozbieraní fotografii nasledovala ich anotácia–vyznačenie jednotlivých kľúčových bodov.

5.1.1 Anotácie

Anotovanie fotografii spočívalo primárne vo vyznačení kľúčových bodov, konkrétne rohov a hranových bodov. V prípade rohu anotácia obsahuje jeho súradnice a triedu. Keďže hranu v jej diskrétnej podobe na obrázku tvorí niekoľko hranových bodov, ktoré tvoria úsečku – hranu, boli hrany anotované v podobe súradníc začiatku a konca hrany spoločne s informá-

Obr. 5.1: Ukážka niekoľkých fotografií z dátovej sady.

ciou o type hrany. Na anotáciu kľúčových bodov bol použitý jednoduchý vlastný anotačný nástroj vytvorený pre tento účel. Anotácie rohov a hrán sú uložené vo formáte csv.

Okrem kľúčových bodov boli taktiež anotované regióny s jednotlivými krabicami v podobe bounding-boxov pre testovaciu dátovú sadu. Bounding box anotácie sú v textovom formáte YOLO[1] a boli vytvorené s využitím programu LabelImg[2]. Bounding box anotácie boli v rámci tejto práce využité za účelom vyhodnotenia úspešnosti jedného z algoritmov na identifikáciu a separáciu jednotlivých inštancii.

5.1.2 Rozdelenie pre tréning a vyhodnocovanie

Aby bolo možné natrénovaný model objektívne vyhodnotiť, bola dátová sada rozdelená na tri časti. Tými sú konkrétne tréningovú sada, validačná sada a testovacia sada. Toto rozdelenie má za cieľ zabezpečiť možnosť vyhodnotenia nad nevidenými dátami, a teda overiť schopnosť generalizácie modelu. Schopnosť generalizovať je totiž kľúčová pre reálne použitie, kde sú jednotlivé vstupy spravidla vopred nevidené.

Niekedy je zvykom použiť rozdelenie len na dve časti, a to tréningovú a validačnú. V tomto prípade však môže nastať problém takzvaného pretrénovania na validačných dátach (validation data overfitting) [19]. Tento jav je spôsobený opakovaným ladením hyper parametrov modelu na základe výkonnosti modelu na validačnej dátovej sade. Z tohto dôvodu je vhodnejšie oddeliť ešte jednu disjunktnú množinu testovacích dát, ktoré budú slúžiť na finálne vyhodnocovanie modelu. Samotné rozdelenie dátovej sady na uvedené tri časti bolo prevedené náhodne v pomere 80:10:10 (v poradí trénovacia, testovacia a validačná sada).

[1] https://towardsdatascience.com/image-data-labelling-and-annotation-everything-you-need-to-know-86ede6c684b1

[2] https://github.com/tzutalin/labelImg

5.2 Použité implementačné technológie

Obr. 5.2: Logá použitých technológii.

Systém bol implementovaný v programovacom jazyku Python verzie 3.8. Tento jazyk som zvolil najmä z dôvodu, že umožňuje rýchle prototypovanie a taktiež kvôli dostupnosti množstva knižníc z oblastí počítačového videnia, neurónových sietí a vizualizácie dát.

Na implementáciu Konvolučnej neurónovej siete bol využitý framework Tensorflow [15] vo verzii 2. Pri implementácii dekodéra výstupu konvolučnej neurónovej siete sú využité niektoré funkcie knižnice pre spracovanie obrazu OpenCV[3] a taktiež je v implementácii niektorých algoritmov využitá knižnica pre akceleráciu výpočtov NumPy[4]. Pre účely vizualizácie bola využitá knižnica Matplotlib[5].

5.3 Implementácia siete s architektúrou Stacked Hourglass

Keďže pôvodná autorská implementácia tejto architektúry nieje voľne dostupná, využil som verejnú implementáciu v jazyku Python využívajúcu framework Tensorflow 2[6]. Tá bola následne modifikovaná pre potreby detekcie rohov a hrán krabice. Z pohľadu architektúry neurónovej siete táto modifikácia spočívala najmä v zmene počtu výstupných kanálov na počet odpovedajúci počtu tried hrán a rohov.

Následne boli na základe experimentov a priebežného vyhodnocovania úspešnosti vykonané aj ďalšie zmeny, medzi ktorými bolo zväčšenie rozlíšenia vstupného obrázka, a teda aj veľkosti výstupných confidence máp a ďalšie. V nasledujúcich sekciách sa nachádza presnejší popis prevedených experimentov a jednotlivých modifikácii, ktoré boli na základe týchto experimentov aplikované.

5.3.1 Počet Hourglass modulov

Ako bolo popísané v kapitole 3.3, sieť architektúry Stacked Hourglass Network sa skladá z niekoľkých Hourglass modulov. Počet týchto modulov je možné meniť s cieľom nájsť ideálny počet z hľadiska rýchlosti a presnosti predikcii. Zvyšovanie počtu týchto modulov vedie k viacnásobnému iteratívnemu prehodnocovaniu, a teda teoreticky k väčšej presnosti. Na druhej strane však taktiež vedie k nárastu počtu parametrov siete. To vedie k zvýšeným nárokom na výpočetný výkon a pamäť, a teda aj k spomaleniu inferencie. Príliš veľký počet parametrov modelu môže zapríčiniť taktiež väčšiu náchylnosť k pretrénovaniu, respektíve väčšie nároky na veľkosť dátovej sady, aby k pretrénovaniu nedochádzalo. V rámci experimentov bola testovaná výkonnosť jednotlivých konfigurácii, a to z pohľadu presnosti aj rýchlosti modelu. Výsledky sú prezentované v tabuľke 5.1.

[3] https://opencv.org/
[4] https://numpy.org/
[5] https://matplotlib.org/
[6] https://github.com/ethanyanjiali/deep-vision/tree/master/Hourglass/tensorflow

Tabuľka 5.1: Tabuľka zobrazuje výsledky experimentálneho vyhodnotenia architektúr s rôznym počtom Hourglass modulov. Pre jednotlivé konfigurácie bola vyhodnotená presnosť detekcie s využitím metriky PCK(kap. 2.4.1) a recall pre rohy krabice. Pri vyhodnocovaní metrík PCK a recall bola použitá hodnota prahu relatívnej vzdialenosti $\alpha = 0.5$. Taktiež bol vyhodnotený čas inferencie v milisekundách ako priemerná hodnota z 1000 inferencií na grafickej karte Nvidia GTX1060.

# Hourglass modulov	Avg. PCK	Avg. recall	Čas inferencie [ms]
2	0.867	0.787	63
3	0.886	0.778	93
4	0.870	0.795	123
5	0.872	0.787	150
6	0.0.873	0.751	181

Tabuľka 5.2: Tabuľka prezentuje výsledky experimentálneho vyhodnotenia siete architektúry Stacked Hourglass s pôvodnou hodnotou hĺbky mapy príznakov v porovnaní s hodnotou dvojnásobnou a polovičnou. Pre danú hĺbku príznakovej mapy bola vyhodnotená presnosť detekcie s využitím metriky PCK a recall s prahom relatívnej vzdialenosti $\alpha = 0.5$. Taktiež bol vyhodnotený čas inferencie v milisekundách ako priemer z 1000 inferencií na grafickej karte Nvidia GTX1060. Pre toto porovnanie bola použitá sieť s 2 Hourglass modulmi.

Hĺbka mapy príznakov	Avg. PCK @ 0.5	Avg. recall	Čas inferencie [ms]
512	0.9142	0.729	135
256 (pôvodná hodnota)	0.867	0.787	67
128	0.862	0.713	65

5.3.2 Hĺbka mapy príznakov

Ďalší experiment spočíval v zmene hĺbky mapy príznakov s ktorým sieť pracuje. Ako bolo popísané v kapitole 3.2.2, jedným z parametrov konvolučnej vrstvy je počet filtrov. Každý filter produkuje na výstupe vrstvy jeden kanál, ktorý popisuje určitý príznak. Voľba počtu filtrov, a teda hĺbky mapy príznakov ovplyvňuje koľko rôznych príznakov bude sieť spracovávať. Voľba ideálnej hodnoty závisí na tom, koľko užitočných príznakov pre daný problém dáta obsahujú a je nutné ju voliť experimentálne. Hodnota tohto parametru okrem presnosti predikcie ovplyvňuje taktiež aj pamäťové a výpočetné nároky modelu, a preto je nutné pri jej voľbe hľadať určitý kompromis medzi týmito aspektmi.

Keďže použitá architektúra bola vytvorená a optimalizovaná za iným účelom – odhadom ľudskej pózy, rozhodol som sa experimentálne vyhodnotiť aj iné ako pôvodné hodnoty tohto parametru. Pôvodne architektúra pracuje maximálne s 256 filtrami. V rámci experimentov bola testovaná dvojnásobná hodnota a polovičná hodnota. Výsledky sú zobrazené v tabuľke 5.2.

5.4 Tréning konvolučnej neurónovej siete

Proces tréningu neurónovej siete prebiehal na stolnom počítači s grafickou kartou Nvidia GTX 1060. 65 epoch tréningu trvalo na tomto hardware v priemere približne 12 hodín. Sieť

bola niekoľkokrát opakovane trénovaná a následne vyhodnocovaná s cieľom hľadania najvhodnejších hyper-parametrov. Konkrétne boli testované rôzne hodnoty learning-rate spolu s rôznymi typmi optimizérov, rôzne hodnoty batch size a ďalšie. V počiatočnej fáze vznikali pri tréningu problémy spôsobené nerovnomernou distribúciou nulových a nenulových hodnôt v dátach. Toto bolo vyriešené modifikáciou chybovej funkcie, kde bolo experimentálne testovaných niekoľko jej variant. Okrem toho bol vyhodnocovaný taktiež efekt využitia augmentácia trénovacích dát. Jednotlivé experimenty budú spolu s výsledkami popísané samostatne ďalej v tejto kapitole.

5.4.1 Syntéza ground-truth dát

Anotácie obsahujú kľúčové body popísané súradnicami a triedou, avšak výstup neurónovej siete má podobu skóre pre jednotlivé pixely v 8 kanáloch (jeden kanál pre každú triedu). Z tohto dôvodu musia byť anotácie pre účely tréningu konvertované do podoby zhodnej s výstupom siete. Tento krok je potrebné aplikovať na každý obrázok v dátovej sade.

Proces spočíva vo vytvorení tenzoru núl o rozmeroch $128 \times 128 \times 8$, kde 8 reprezentuje počet tried rohov a hrán a 128 predstavuje výšku a šírku výstupnej confidence mapy. Následne je potrebné vložiť reprezentácie jednotlivých kľúčových bodov do príslušných kanálov na základe triedy kľúčového bodu. Kľúčové body sú reprezentované dvojrozmernou Gaussovou krivkou, ktorá má maximum na pozícii bodu ktorý predstavuje.

Prvý z parametrov Gaussovej krivky – stredná hodnota je teda určená súradnicami kľúčového bodu. Druhý parameter – smerodajná odchýlka je vopred pevne stanovený, a to zvlášť pre rohy a hranové body, pričom jej hodnota bola zvolená experimentálne. Pri rohoch je použitá hodnota 1.5 a pre hranové body hodnota 1.0. Experimenty ukázali, že zväčšenie smerodajnej odchýlky z hodnoty 1.0 na 1.5 pri rohoch vedia k malému zlepšeniu presnosti modelu v porovnaní s hodnotou 1.0. Ďalšie zväčšenie však spôsobovalo problémy kvôli častejšiemu prekrývaniu jednotlivých kľúčových bodov a neviedlo k ďalšiemu zlepšeniu presnosti. Podobne tomu bolo aj pri hranách, kde zväčšenie hodnoty nemalo požadovaný efekt. Pravdepodobne tomu tak je z dôvodu, že hranu tvorí veľké množstvo hranových bodov blízko pri sebe, vďaka čomu hranové body zaberajú výrazne väčšiu plochu ako rohy.

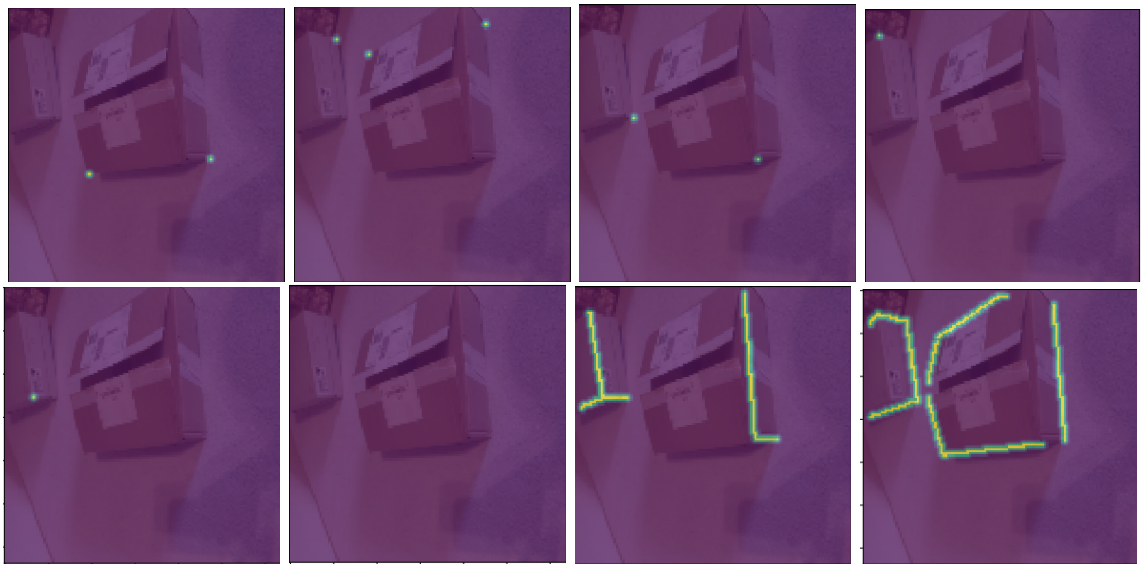
Príklad ground-truth confidence máp pre jednotlivé triedy rohov a hranových bodov vizualizovaných v podobe tepelnej mapy je možné vidieť na obrázku 5.3.

5.4.2 Chybová funkcia

Typicky používaná chybová funkcia (loss function) v úlohách regresie je priemerná štvorcová chyba (mean squered error, skrátene MSE). Z tohto dôvodu bola MSE prvou voľbou aj pre tréning CNN tohto systému. Prvotné experimenty však ukázali problémy modelu naučiť sa predikovať Gaussové krivky predstavujúce kľúčové body, najmä v prípade rohov. Ako príčina týchto problémov sa neskôr ukázala práve chybová funkcia.

Problém spočíva v nerovnomernej distribúcii nulových a nenulových hodnôt v trénovacích dátach. Keďže v obrázkoch sa vyskytuje iba niekoľko málo rohov, je počet nenulových hodnôt v ground-truth confidence mape pomerne malý v porovnaní s nulovými hodnotami, ktoré predstavujú všetky ostatné miesta v obrázku. Toto viedlo k tendencii neurónovej siete naučiť sa triviálne riešenia v podobe predpovede nulových hodnôt pre všetky pozície. Z dôvodu prevahy nulových hodnôt totiž toto riešenie predstavuje lokálne minimum chybovej funkcie, v ktorom sieť v priebehu tréningu uviazla.

Pre riešenie tohto problému bolo teda nutné chybovú funkciu modifikovať tak, aby tento prípad dostatočne penalizovala. Pre tento účel bola použitá kombinácia MSE s váhovaním



Obr. 5.3: Ukážka ground-truth dát použitých pre tréning. Na obrázku je možné vidieť niekoľko confidence máp vizualizovaných vo forme tepelnej mapy. Každá z nich popisuje pozície iného typu kľúčových bodov.

na základe podielu počtu nenulových pixlov (označené ako n_b) voči celkovému počtu pixlov ($w \times h$) v ground-truth dátach. Váhy sú definované pre každý pixel confidence mapy ako:

$$w_{ij} = \begin{cases} 1 - \frac{n_b}{w \cdot h} & \text{ak } l_{i,j} > 0 \\ \frac{n_b}{w \cdot h} & \text{inak} \end{cases}, \tag{5.1}$$

,

kde i a j predstavujú indexy odpovedajúceho pixlu. Následne sa spočíta štvorcový rozdiel medzi predikciami a ground-truth dátami pre jednotlivé pixely, ktorý sa vynásobí príslušnými váhami. Potom je táto matica hodnôt redukovaná na skalár – spočíta sa aritmetický priemer chýb na všetkých pozíciach :

$$E = \frac{1}{w \cdot h} \sum_{i=1}^{w} \sum_{j=1}^{h} (y_{ij} - l_{ij})^2 \cdot w_{ij}. \tag{5.2}$$

.

5.4.3 Voľba optimizéra

Pôvodná práca navrhuje použitie optimizéra RMSprop [25] s hodnotou learning rate na hodnote 0.00025 spolu s jedným znížením na polovicu po tom, čo sa validačná chyba nezlepší v 5 po sebe idúcich epochách. V rámci prevedených experimentov boli testované taktiež iné optimizéry ako aj rôzne hodnoty learning rate. Pre finálny tréning bol využitý optimizér Adam [11] s pôvodnými hodnotami parametrov pre Tensorflow 2. Porovnanie je zobrazené v tabuľke 5.3.

5.4.4 Augmentácia dát

Keďže použitá vlastná dátová sada je pomerne malá, je vhodné využiť augmentáciu s cieľom zlepšiť generalizačné schopnosti modelu. Z hľadiska transformácii boli experimentálne

Tabuľka 5.3: Tabuľka prezentuje experimentálne zistené hodnoty metrík PCK a recall v závislosti od optimizéra použitého pre tréning neurónovej siete. Metriky PCK a recall boli vyhodnocované vzhľadom k prahu relatívnej vzdialenosti $\alpha = 0.5$. Pre toto porovnanie bola použitá sieť s 2 Hourglass modulmi.

Optimizér	**Počiatočná miera učenia**	**Avg. PCK @ 0.5**	**Avg.recall**
Adam	0.001	0.867	0.788
RMSprop	0.00025	0.871	0.728

Tabuľka 5.4: Tabuľka prezentuje experimentálne zistené hodnoty metrík PCK a recall pre model natrénovaný bez augmentácie trénovacích dát v porovnaní s využitím náhodnej zmeny jasu a kontrastu vrámci určeného intervalu. V tomto prípade bol použitý model s 2 Hourglass modulmi. Pri vyhodnocovaní bol použitým prah relatívnej vzdialenosti $\alpha = 0.5$.

Augmentácia	**Avg. PCK @ 0.5**	**Avg.recall**
žiadna	0.858	0.762
jas a kontrast	0.867	0.787

vyskúšané rozličné kombinácie. Z otestovaných transformácii sa uplatnili zmena kontrastu a zmenu jasu. Maximálne hodnoty jasu a kontrastu boli stanovené na základe vizuálneho vyhodnotenia s cieľom zamedziť nerealistickým transformáciám. Zmena jasu jednotlivých trénovacích obrázkov prebiehala dynamicky počas tréningu. Konkrétne bola v jednotlivých tréningových krokoch hodnota jasu a kontrastu zmenená o náhodne vygenerovanú hodnotu. Za týmto účelom boli použité funkcie z image modulu Tensorflow 2, a to konkrétne funkcia *random_contrast()* s hodnotami faktoru zmeny 0.6 až 1.7 a funkcia *random_brightness()* s hodnotou parametru *max_delta* rovnou 20. Ukážka augmentovanej fotografie v porovnaní s pôvodnou je zobrazená na obrázku 5.4.

Okrem toho bolo testované taktiež zmenšenie obrázka s cieľom zlepšiť výkonnosť pre vzdialené a malé objekty. Toto však neviedlo k želaným výsledkom a spôsobovalo značný pokles presnosti na netransformovaných validačných dátach. Vplyv použitia augmentácie na výkonnosť modelu je možné pozorovať v tabuľke 5.4.

5.5 Implementácia dekodéra

Časť systému zvaná dekodér má za úlohu spracovanie výstupu neurónovej siete do podoby konečného výstupu systému. V tejto kapitole budú detailnejšie popísané jednotlivé kroky spomenuté v kapitole 4.5, a to najmä z pohľadu implementácie algoritmov ktoré využívajú.

5.5.1 Rozlíšenie jednotlivých krabíc

Pri odhade pózy s prístupom zhora dolu sú najprv identifikované oblasti kde sa nachádzajú jednotlivé inštancie objektov a následne sú detekované jednotlivé kľúčové body, čím sa zabezpečí, že všetky kľúčové body patria jednému objektu. Ako však bolo spomenuté tento systém využíva prístup zdola nahor, a teda dekodér spracováva celú množinu kľúčových bodov, ktoré patria ľubovoľnému počtu krabíc, ktoré sa nachádzali na vstupnej fotografii.

Obr. 5.4: Vľavo vstupný tréningový vstupný obrázok pred augugmentáciou, v pravo obrázok po augmentácii. Augmentácia spočíva v náhodnej zmene kontrastu a jasu obrázka.

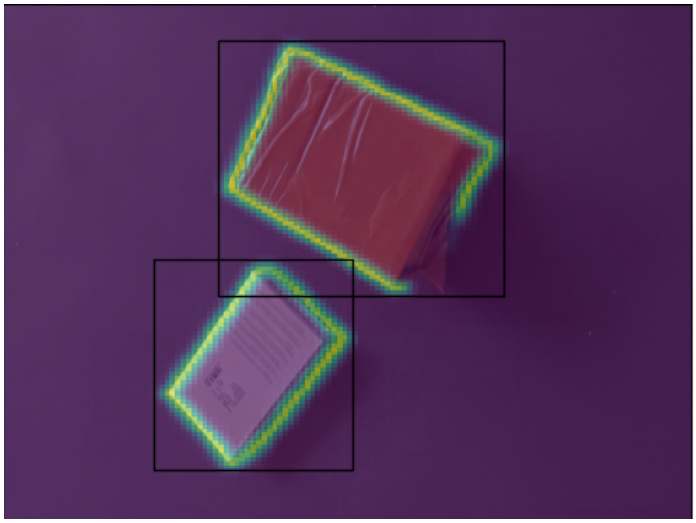
S cieľom získať prehľad o oblastiach s jednotlivými objektmi v podobe bounding boxov, podobne ako pri top-down detekcii, môžu byť využité obrysové hrany krabice. Každú krabicu totiž pre daný pohľad tvorí niekoľko obrysových hrán, ktoré spoločne definujú oblasť v tvare polygónu pokrývajúcu celú krabicu. Jednotlivé polygóny je možné nájsť na základe identifikácie spojitých komponentov v predikciách obrysových hrán. Na základe týchto komponentov je následne možné získať bounding boxy jednotlivých inštancii krabíc. Konkrétne sú súradnice ľavého dolného rohu získané ako minimálne hodnoty x a y súradníc danej komponenty a pravý horný roh analogicky ako hodnoty maximálne. Konkrétny príklad s vizualizovanými predikciami obrysových hrán a na základe nich určených bounding boxov je zobrazený v obrázku 5.5.

Problematické pre tento prístup sú prípady, kedy sa jednotlivé inštancie objektov prekrývajú, a teda ich nemožno odlíšiť na základe izolovaných komponentov. V tomto prípade sú tieto prekrývajúce sa objekty označené spoločným bounding boxom a je potrebné túto situáciu riešiť neskôr iným spôsobom.

5.5.2 Prevod confidence mapy na súradnice rohov

Pre ďalšie spracovanie je potrené získať z confidence máp rohov súradnice pozícií, na ktorých sa jednotlivé rohy vyskytujú. Každý z rohov predstavuje v confidence mape jednu Gaussovu krivku, kde Súradnice rohu odpovedajú pozícii jej maxima. Aby sme získali súradnice všetkých detegovaných bodov, ktorých môže byť v danej confidence mape niekoľko, nemohol byť využitý konvenčný prístup založený na pozícii maxima z danej confidence mapy. Tento prístup je totiž vhodný iba pre extrakciu jediného výskytu pri prístupe zhora nadol. Z tohto dôvodu prebieha extrakcia rohov v niekoľkých čiastkových krokoch.

Prvým je prahovanie s cieľom potlačiť miesta s malým skóre (non-maximum suppression). Hodnota prahu bola stanovená experimentálne na základe grafu 6.1 na hodnotu 0.2. Nasleduje identifikácia miest s vysokým skóre, ktoré predstavujú jednotlivé rohy. To spočíva v nájdení spojitých komponentov v prahovaných confidence mapách pre jednotlivé triedy



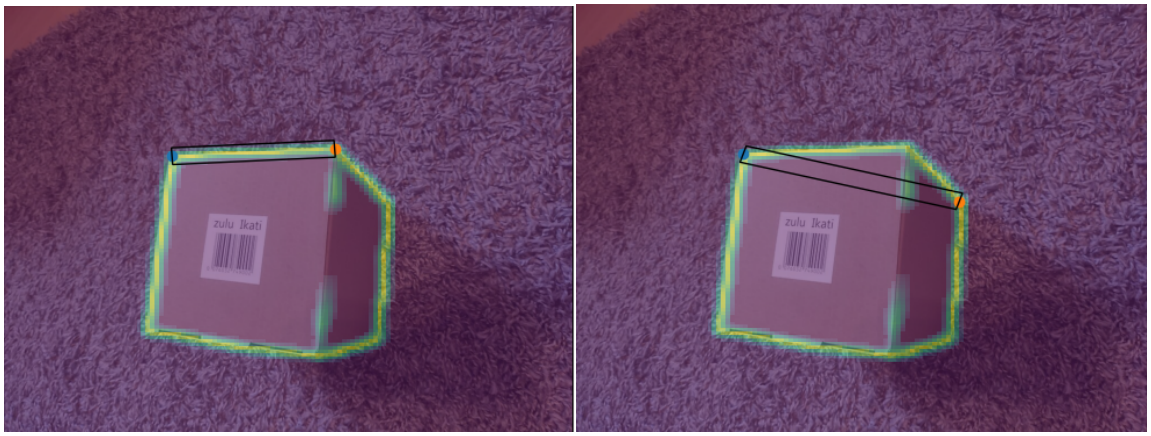
Obr. 5.5: Obrázok zobrazuje vizualizáciu confidence mapy obrysových hrán vo forme tepelnej mapy, ktorá je zobrazená v prekryve so vstupným obrázkom. Na obrázku možno vidieť jednotlivé spojité komponenty tvorené predikovanými hranovými bodmi a taktiež dva bounding boxy, ktoré boli na ich základe identifikované.

rohov. Následne je potrebný posledný krok – pre každú identifikovanú oblasť získať súradnice odpovedajúceho rohu. Tie sa získajú ako pozícia ťažiska tejto oblasti. Ťažisko sa získa ako aritmetický priemer súradníc jednotlivých bodov oblasti. Otestovaná bola taktiež alternatíva s použitím váženého priemeru na základe skóre, toto však neviedlo k očakávanému zlepšeniu.

5.5.3 Identifikácia párov rohov tvoriacich hrany

V tomto momente sú známe súradnice jednotlivých rohov krabice. Niektoré dvojice z týchto rohov tvoria hrany krabice. Je však potrebné zistiť, ktoré rohy prislúchajú k jednotlivým hranám. Na tento účel je možné použiť confidence mapy hranových bodov. Algoritmus je založený na vyhodnocovaní podmienky, či bola medzi jednotlivými dvojicami bodov detegovaná hrana. Toto vyhodnocovanie prebieha tak, že sa najprv pre skúmaný pár rohov vytvorí maska v tvare obdĺžnika nad ich spojnicou. Šírka obdĺžnika je pevne daná a bola zvolená experimentálne. Potom je vyhodnotený pomer počtu všetkých pixelov masky a nenulových pixelov pod maskou v confidence mape hrán. Následne sa tento pomer porovná s experimentálne určenou prahovou hodnotou. Ak je hodnota väčšia ako prah, je aktuálne testovaný pár rohov prehlásený za korektný a do výsledného modelu krabice je vložená nimi tvorená hrana. Ak došlo k pozitívnej identifikácii, oblasť confidence mapy pod maskou sa vynuluje, ak nie, ponechá sa nezmenená. Následne sa tento krok opakuje kým sú identifikované všetky hrany. Toto vyhodnocovanie je vykonávané zvlášť pre obrysové a vnútorné hrany aby bolo možné určiť triedu identifikovanej hrany.

V rámci vývoja bol taktiež testovaný prístup založený na parametrickom vyjadrení jednotlivých hrán na základe hranových bodov. Na to bola využívaná metóda fittingu Gaussových kriviek na confidence mapy hranových bodov. Tento prístup však bol neskôr nahradený hore popísaným, najmä z dôvodu výrazne menšej výpočetnej náročnosti.



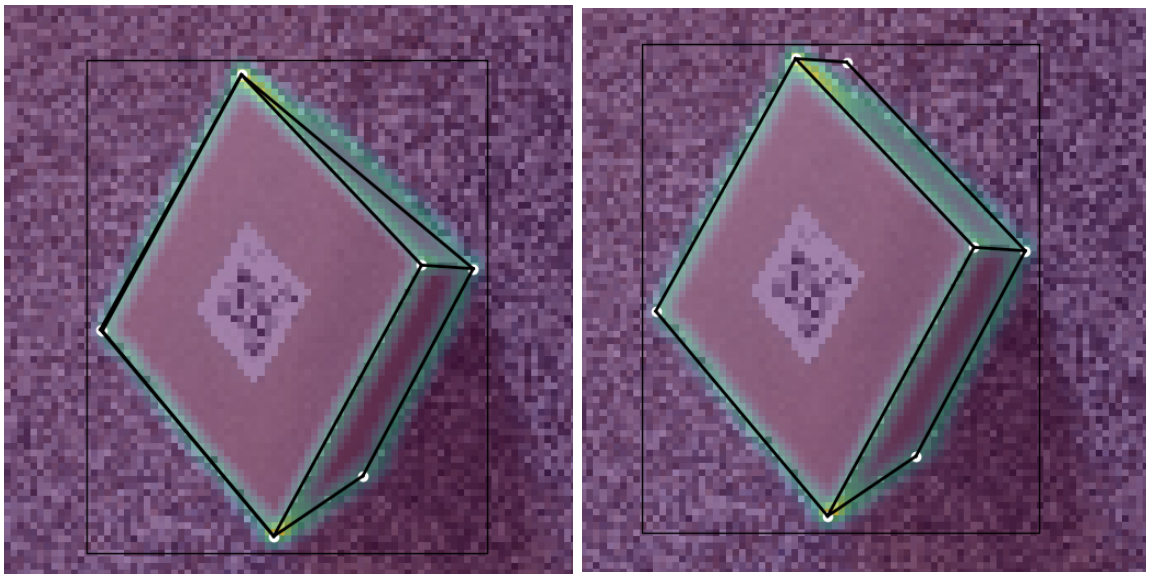
Obr. 5.6: Ukážka algoritmu pre určenie párov rohov. Na ľavej strane je zobrazený príklad korektného páru rohov – rohy skutočne tvoria hranu. V toto prípade je podiel nenulových pixelov pod maskou v confidence mape hrán rovný 91 %. Na pravej strane je naopak príklad nesprávneho páru (26 % nenulových pixelov). V tomto prípade sa jedná o confidence mapu obrysových hrán.

5.5.4 Korekcia chýb modelu krabice

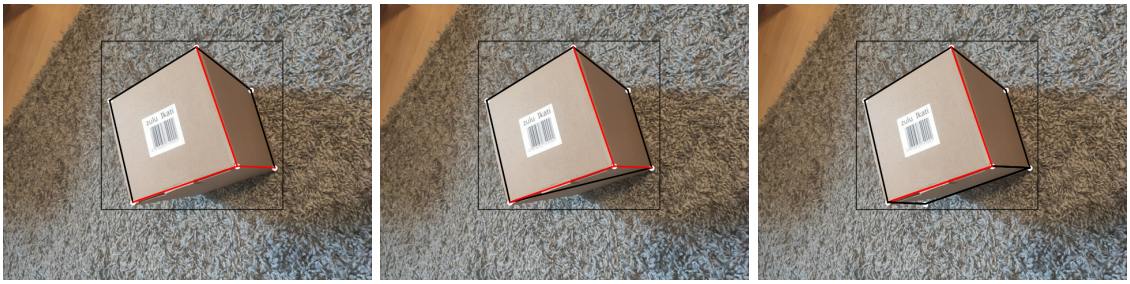
V predikciách neurónovej siete sa v určitej miere samozrejme vyskytujú aj chyby. Príkladom týchto chýb sú napríklad nedetegované kľúčové body, spojenie blízkych kľúčových bodov do jedného a falošné detekcie. K určitému druhu odchýlok môže v niektorých prípadoch taktiež dôjsť v procese spracovania výstupu neurónovej siete dekodérom. Tieto chyby sa následne prejavia na výstupe systému v nepresnostiach modelov detegovaných krabíc.

V niektorých prípadoch je možné využiť vlastnosti geometrického tvaru krabice – tvar kvádra, na identifikáciu a do určitej miery aj korekciu spomínaných chýb. Konkrétny príklad takejto chyby je nedetegovanie jedného z rohov. Táto chyba nastala pravdepodobne z dôvodu, že sa tento roh nachádzal v blízkosti ďalšieho rohu. Chyba spôsobila, že model tvaru krabice je nesprávny, a to tak, že jedna zo stien krabice má tvar trojuholníka miesto obdĺžnikového tvaru, ktorý odpovedá realite. Táto situácia môže byť detegovaná a následne korigovaná odhadnutím pozície chýbajúceho rohu. Pre samotný odhad pozície je v tomto prípade trojuholník tvoriaci chybnú stenu doplnený na rovnobežník. Vizualizácia popísanej korekcie sa nachádza na obrázku 5.7, kde na ľavej strane sa nachádza model krabice pred korekciu a na pravej po korekcii.

Ďalší spôsob korekcie je založený na fakte, že obrysové hrany musia tvoriť uzavretú komponentu. Ak niektorá z obrysových hrán chýba, nebude táto podmienka splnená, čo je možné pomerne jednoducho odhaliť. Následne je možné túto chybu korigovať doplnením hrany medzi vrcholmi, kde je porušená spojitosť. Pri tom môže nastať situácia popísaná v predošlom kroku – niektorá stena nadobudne nekorektný tvar v podobe trojuholníka. Toto sa dá znova korigovať spôsobom popísaným v predchádzajúcom odstavci. Konkrétny príklad s vizualizáciou jednotlivých krokov je zobrazený na obrázku 5.8.



Obr. 5.7: Ukážka korekcie chyby ktorá vznikla z dôvodu nedetegovania jedného z rohov. Na ľavej strane je zobrazený model tvaru krabice pred korekciou chyby. Možno si všimnúť, že jedna zo stien krabice má nesprávny tvar – tvorí trojuholník. Následne je táto chyba korigovaná doplnením tohto trojuholníka na rovnobežník, ako je možné vidieť na obrázku vpravo.



Obr. 5.8: Ukážka korekcie chýbajúcej obrysovej hrany na základe porušenia spojitosti. Na obrázku je možne vidieť jednotlivé kroky v ktorých korekcia prebiehala. Zľava doprava je na obrázkoch pôvodne predikovaný model pred korekciou, následne po doplnení chýbajúcej hrany medzi vrcholy kde došlo k porušeniu spojitosti a úplne vpravo po korigovaní trojuholníkovej strany krabice na rovnobežník.

Kapitola 6

Experimentálne vyhodnotenie

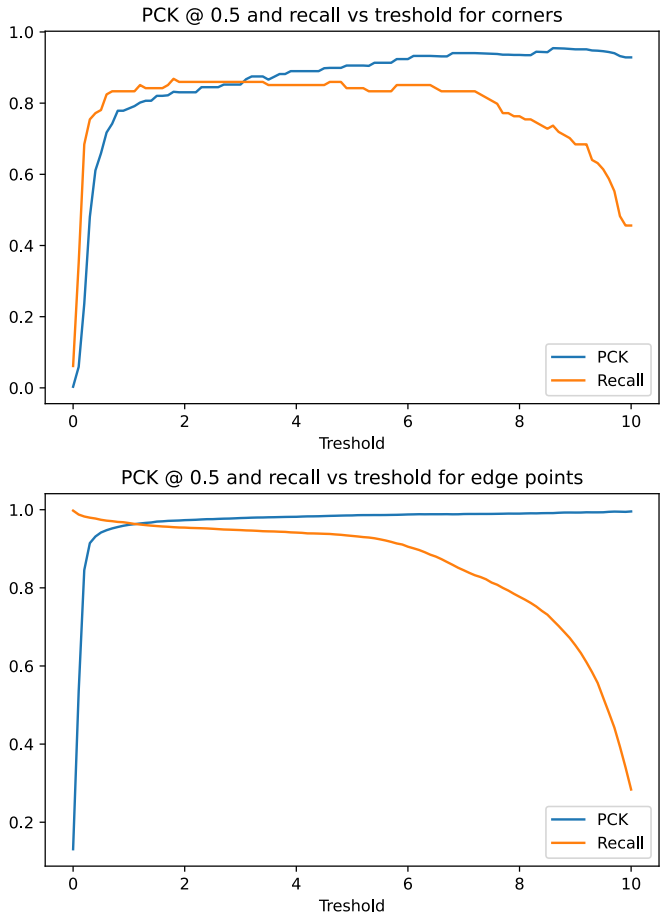
Táto kapitola sa zaoberá vyhodnotením implementovaného systému na detekciu krabíc. Systém bol vyhodnocovaný z pohľadu viacerých aspektov, ktoré sú dôležité z pohľadu jeho prípadnej využiteľnosti. Medzi tieto aspekty patria najmä presnosť a rýchlosť inferencie. Primárne vyhodnotenie spočívalo vo vyhodnotení detekcie kľúčových bodov. Pre vyhodnotenie kroku separácie inštancii krabíc bolo taktiež využité vyhodnotenie na základe bounding boxov vyznačujúcich oblasti s jednotlivými krabicami.

6.1 Vyhodnotenie detekcie rohov a hrán krabice

Keďže systém popisuje krabice na základe kľúčových bodov, je presnosť ich lokalizácie kľúčová pre kvalitu výstupu systému. Spôsob vyhodnocovania je inšpirovaný postupmi používanými v odvetví odhadu pózy ľudskej postavy.

Pre samotné vyhodnotenie boli využité dve metriky. Prvou z nich je pravdepodobnosť korektného kľúčového bodu (Probability of correct keypoint) skrátene PCK (kap. 2.4.1), ktorá popisuje presnosť produkovaných predikcii. Ďalšou je recall (kap. 2.4.2), ktorý vyjadruje podiel detegovaných kľúčových bodov z tých, ktoré sa reálne nachádzajú na vstupnom obrázku.

Ako bolo spomínané, sieť predikuje skóre výskytu kľúčových bodov pre jednotlivé pozície obrázka a nie priamo súradnice kľúčových bodov. Súradnice sú potom extrahované postupom popísaným v kapitole 5.5.2. Jedným z podstatných krokov pri získavaní koordinátov z confidence máp rohov je non-maximum suppression. Hodnota prahu použitého v tomto kroku má významný vplyv na konečnú presnosť detekcie kľúčových bodov, a to nasledujúcim spôsobom. Pri nízkej hodnote prahu sú za kľúčové body považované aj miesta z nižším skóre, a teda je väčšia pravdepodobnosť že medzi nimi budú aj falošné detekcie – (false positives). Naopak s rastúcou hodnotou prahu narastá pravdepodobnosť že budú potlačené aj detekcie, ktoré boli správne, čo povedie k nedetegovaniu niektorých kľúčových bodov (false negatives). Voľba hodnoty prahu je teda dôležitá pre celkovú výkonnosť detekčného systému. Pri jeho voľbe je nutné zvážiť ktorý typ chyby je v danom prípade menej problematický a na základe toho zvoliť najvhodnejšiu hodnotu prahu. Aby bolo možné túto voľbu vykonať, bolo prevedené vyhodnotenie metrík PCK a recall v závislosti na prahovej hodnote. Konkrétne namerané hodnoty je možné vidieť v grafoch na obrázku 6.1.



Obr. 6.1: Grafy zobrazujú závislosť hodnôt metrík PCK a recall na hodnote zvoleného prahu. Graf hore reprezentuje vyhodnotenie detekcie rohov krabíc, zatiaľ čo graf na dole predstavuje vyhodnotenie detekcie hranových bodov.

6.2 Vplyv veľkosti dátovej sady na presnosť detektora

V úlohách využívajúcich učenie s učiteľom (supervised learning) má veľkosť dátovej sady významný vplyv na presnosť modelu. Keďže tento systém bol trénovaný na vlastnej dátovej sade, ktorá bola zhromažďovaná postupne v priebehu vývoja, bol spomínaný efekt veľmi dobre pozorovateľný. V tabuľke6.1 sú uvedené experimentálne zistené hodnoty PCK (kap. 2.4.1) a recall pre modely natrénované na rôzne veľkých častiach dátovej sady. Na základe výsledkov tohto experimentu možno konštatovať, že presnosť detektora môže byť ešte vylepšená zväčšením dátovej sady použitej pre tréning.

6.3 Vyhodnotenie separácie inštancií na základe bounding boxov

Ako bolo popísané v kapitole 5.5.1, jedným z čiastkových krokov spracovania výstupu CNN je krok separácie inštancii krabíc. Výstupom tohto kroku sú obdĺžnikové oblasti s výskytom jednotlivých objektov (bounding box). Pre vyhodnotenie tohto kroku je teda vhodné použiť metriky využívané pri vyhodnocovaní detektorov objektov, ktorých výstup

Tabuľka 6.1: V tabuľke sú uvedené experimentálne zistené hodnoty metrík PCK a recall pre rohy pri 3 rôznych veľkostiach dátovej sady použitej pre tréning modelu. Pre zachovanie objektivity bola vo všetkých prípadoch využitá rovnaká množina testovacích dát, ktorá bola vždy disjunktná s množinou dát použitých pre tréning.

# trénovacích obrázkov	# rohov	PCK @ 0.5	Recall @ 0.5
150	838	82 %	79 %
300	1706	87 %	85 %
436 (všetky)	2492	89 %	86 %

Tabuľka 6.2: Vyhodnotenie kroku separácie inštancii krabíc na základe bounding boxov. Algoritmus separácie spočíva v identifikácii spojitých komponentov v confidence mape obrysových hrán. V tabuľke sú uvedené hodnoty metrík recall a PCK. Predikovaný bounding box bol považovaný za korektný pokiaľ hodnota IoU bola väčšia ako 0.5. V tomto kroku je väčší dôraz kladený na recall, nakoľko falošné detekcie môžu byť korigované neskôr v procese zostavovania modelov krabíc.

Recall @ 0.5 IoU	Acc. @ 0.5 IoU	# test imgs
96.82 %	76.25 %	54

má podobu bounding-boxov. Konkrétne boli vyhodnotené hodnoty metrík recall a precision, pričom korektnosť predikovaného bounding-boxu bola posudzovaná vzhľadom k hodnote IoU(sec. 2.4.3) 0.5. Konkrétne namerané hodnoty uvedených metrík sú vypísané v tabuľke 6.2.

6.4 Vyhodnotenie rýchlosti

Ďalším z dôležitých aspektov z pohľadu využiteľnosti systému je jeho rýchlosť. V mnohých aplikáciach je totiž nutné spracovanie v reálnom čase. Bežným príkladom sú aplikácie vyžadujúce detekciu objektov vo videu. Iné aplikácie s nižšími nárokmi na rýchlosť môžu zase vyžadovať určitú rýchlosť spracovania kvôli interaktivite (near-real-time).

Vyhodnocovanie rýchlosti systému prebiehalo na grafickej karte Nvidia GTX 1060. Test spočíval v mnohonásobnom meraní času inferencie a výpočte priemernej hodnoty času potrebného na inferenciu CNN a spracovanie dekodérom. V tabuľke 6.3 sú uvedené konkrétne nemerané hodnoty.

6.5 Porovnanie s existujúcimi riešeniami

Nakoľko sa táto práca zaoberá pomerne špecifickým problémom, je pomerne problematické nájsť ekvivalentné riešenia pre porovnanie. Na základe rozšírenosti automatizácie naprieč všetkými priemyselnými odvetviami možno predpokladať, že existujú proprietárne systémy na detekciu krabíc. Keď sa však pozrieme na open-source systémy a vedecké publikácie, ponúkajú sa iba všeobecné riešenia na detekciu objektov, alebo prípadne obecné detektory kuboidov. Prístup ktorý využíva systém predstavený v tejto práci je zvoleným prístupom najviac príbuzný detekcií kuboidov. Z hľadiska porovnania existuje však niekoľko rozdielov medzi týmto systémom a voľne dostupnými systémami na detekciu kuboidov, ktoré sú

Tabuľka 6.3: Tabuľka zobrazuje namerané hodnoty času inferencie systému ako celku. Uvedené hodnoty boli získané ako priemer zo 100 opakovaných meraní pre 10 náhodne vybraných vstupov . Merania boli vykonávané na stolnom počítači s grafickou kartou Nvidia GeForce GTX 1060 a procesorom Intel Xeon E3-1226 v3. Pri tomto meraní bola použitá sieť s 2 Hourglass modulmi a pôvodnou hĺbkou mapy príznakov.

CNN priemer [ms]	**dekodér priemer [ms]**	**spolu priemer [ms]**
67	17	84

popísané v kapitole 2.3. Prvý rozdielom je fakt, že táto práca sa špecializuje na detekciu krabíc, zatiaľ čo spomínané systémy boli vytvorené ako obecné detektory kuboidov, a teda boli trénované a vyhodnocované na rôznych typoch predmetoch tvaru kvádra. Ďalším rozdielom je rozdiel v reprezentácii objektov. Systém navrhnutý v tejto práci popisuje krabicu na základe viditeľných rohov a hrán s cieľom vystihnúť tvar presnejšie ako klasický osovo zarovnaný bounding-box, zatiaľ čo spomínané systémy majú za cieľ odhadnúť kompletný 3D model kvádra. Tieto fakty spoločne s nedostupnosťou zdrojových kódov týchto systémov činia priame porovnanie značne komplikovaným.

6.6 Vizuálne vyhodnocovanie a slabé stránky systému

Okrem vyhodnocovania systému na základe exaktných metrík tak, ako je to popísané v predošlých sekciách, bolo počas vývoja často využívané aj vizuálne vyhodnocovanie. To spočívalo najmä vo vizuálnom zhodnotení problémových prípadov s cieľom identifikovať príčinu, a pokiaľ možno ju odstrániť. Identifikované problémy sa dajú rozdeliť do dvoch kategórii podľa toho, v ktorej časti systému majú pôvod.

Prvým typom sú chyby detektora hrán a rohov. Tie spočívajú vo falošných alebo chýbajúcich detekciách kľúčových bodov. Jedným z problémov tohto typu, ktorý sa vyskytol, bol problém s detekciou rohov a hranových bodov na tieni vrhanom krabicou. Tento problém bol riešený s využitím techniky získavania ťažkých príkladov (hard sample mining) [2]. Princípom je rozšírenie dátovej sady o viacero problematických príkladov daného typu s cieľom poskytnúť modelu dostatok informácii na správne posúdenie. Týmto spôsobom sa podarilo problém do značnej miery eliminovať.

Prirodzene nie všetky problémy systému sa podarilo úspešne odstrániť. Z pohľadu spracovania výstupu neurónovej siete sú problematické najmä situácie, keď sa určitým spôsobom prekrýva viacero objektov– krabíc, kedy nastávajú problémy so separáciou jednotlivých inštancii.

Ďalší typ chýb spočíva v chybách detekcie jednotlivých kľúčových bodov. Ako je zjavné z výsledkov grafu v obrázku 6.1 detektor pochopiteľne nepracuje so 100 percentnou úspešnosťou. Chyby predikcie sa prejavujú najmä v prípade chýbajúcich detekcii rohov, čo môže spôsobiť absenciu niektorých hrán na výstupe. Konkrétne príklady vizualizovaných výstupov, vrátane niektorých problematických zobrazuje obrázok 6.2.

Obr. 6.2: Ukážka vizualizovaných výstupov systému vrátane niekoľkých problematických prípadov.

Kapitola 7

Záver

Cieľom tejto bakalárskej práce bolo navrhnúť a implementovať systém na detekciu kartónových krabíc v RGB obrázkoch. Súčasťou práce bolo taktiež vytvorenie vlastnej dátovej sady kartónových krabíc pre tréning konvolučnej neurónovej siete. Vytvorené riešenie je založené na detekcii krabice na základe jej typických častí–rohov a hrán. Systém využíva metódu takzvanej regresie tepelných máp, ktorá je často využívaná v oblasti odhadu ľudskej pózy, a aplikuje ju v kontexte detekcie krabíc. Výstupy konvolučnej neurónovej siete sú následne spracúvané časťou systému zvanou dekodér do konečnej podoby modelov tvaru krabíc, ktoré sa nachádzali na vstupnom obrázku. V rámci práce bolo vykonané veľké množstvo experimentov týkajúcich sa tréningu siete, ako aj algoritmov pre spracovanie predikcii. Výsledné riešenie dosahuje na základe vyhodnotenia nad nevidenými dátami PCK 90 % a recall 86 % pre rohy krabice respektíve 97.5 % a 96 % pre hranové body a môže byť použité napríklad ako základ pre implementáciu automatizovaných systémov na triediacej linke.

Na základe pozorovania vzťahu veľkosti dátovej sady a presnosti, ktorý je popísaný v kapitole 6.2, usudzujem že presnosť systému môže byť ďalej zvýšená použitím komplexnejších metód augmentácie, tréningom na syntetických dátach, alebo jednoducho zväčšením dátovej sady. Ďalší nápad pre vývoj spočíva v experimentovaní s prístupom zhora nadol s využitím kombinácie detektora založeného na regiónoch pre separáciu inštancii a detekciou kľúčových bodov tak, ako bola popísanej v tejto práci.

K práci som vytvoril taktiež krátke prezentačné video[1].

[1] https://www.youtube.com/watch?v=amZLA1l5XcI

Literatúra

[1] Andriluka, M., Pishchulin, L., Gehler, P. a Schiele, B. 2D Human Pose Estimation: New Benchmark and State of the Art Analysis. In: IEEE. *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, s. 3686–3693. DOI: 10.1109/CVPR.2014.471. ISBN 9781479951192.

[2] Chen, K., Chen, Y., Han, C., Sang, N. a Gao, C. Hard sample mining makes person re-identification more efficient and accurate. *Neurocomputing*. 1. vyd. 2020, zv. 382, s. 259–267. DOI: https://doi.org/10.1016/j.neucom.2019.11.094. ISSN 0925-2312. Dostupné z: https://www.sciencedirect.com/science/article/pii/S0925231219316984.

[3] Chen, Y., Tian, Y. a He, M. Monocular human pose estimation: A survey of deep learning-based methods. *Computer Vision and Image Understanding*. 1. vyd. Elsevier BV. Mar 2020, zv. 192, s. 102897. DOI: 10.1016/j.cviu.2019.102897. ISSN 1077-3142. Dostupné z: http://dx.doi.org/10.1016/j.cviu.2019.102897.

[4] Cortes, C. a Vapnik, V. Support-vector networks. *Machine learning*. 1. vyd. Springer. 1995, zv. 20, č. 3, s. 273–297.

[5] Dalal, N. a Triggs, B. Histograms of oriented gradients for human detection. In: IEEE. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. 2005, sv. 1, s. 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177. ISBN 0-7695-2372-2.

[6] Dasiopoulou, S., Mezaris, V., Kompatsiaris, I., Papastathis, V. . a Strintzis, M. G. Knowledge-assisted semantic video object detection. *IEEE Transactions on Circuits and Systems for Video Technology*. 1. vyd. 2005, zv. 15, č. 10, s. 1210–1224. DOI: 10.1109/TCSVT.2005.854238.

[7] Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q. et al. CenterNet: Keypoint Triplets for Object Detection. *CoRR*. 3. vyd. 2019, abs/1904.08189. Dostupné z: http://arxiv.org/abs/1904.08189.

[8] Dwibedi, D., Malisiewicz, T., Badrinarayanan, V. a Rabinovich, A. Deep Cuboid Detection: Beyond 2D Bounding Boxes. *CoRR*. 1. vyd. 2016, abs/1611.10010. Dostupné z: http://arxiv.org/abs/1611.10010.

[9] Girshick, R. B., Donahue, J., Darrell, T. a Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*. 5. vyd. 2013, abs/1311.2524. Dostupné z: http://arxiv.org/abs/1311.2524.

[10] Ioffe, S. a Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *CoRR*. 3. vyd. 2015, abs/1502.03167. Dostupné z: http://arxiv.org/abs/1502.03167.

[11] Kingma, D. P. a Ba, J. *Adam: A Method for Stochastic Optimization*. 2017.

[12] LeCun, Y., Cortes, C. a Burges, C. MNIST handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*. 1. vyd. 2010, zv. 2.

[13] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. E. et al. SSD: Single Shot MultiBox Detector. *CoRR*. 5. vyd. 2015, abs/1512.02325. Dostupné z: http://arxiv.org/abs/1512.02325.

[14] Lowe, D. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*. 1. vyd. November 2004, zv. 60, s. 91–110. DOI: 10.1023/B

[15] Martín Abadi, P. B. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Software available from tensorflow.org. Dostupné z: https://www.tensorflow.org/.

[16] Munea, T. L., Jembre, Y. Z., Weldegebriel, H. T., Chen, L., Huang, C. et al. The Progress of Human Pose Estimation: A Survey and Taxonomy of Models Applied in 2D Human Pose Estimation. *IEEE Access*. 1. vyd. 2020, zv. 8, s. 133330–133348.

[17] Nash, K. O. andRyan. An Introduction to Convolutional Neural Networks. *CoRR*. 2. vyd. 2015, abs/1511.08458. Dostupné z: http://arxiv.org/abs/1511.08458.

[18] Newell, A., Yang, K. a Deng, J. Stacked Hourglass Networks for Human Pose Estimation. *CoRR*. 2. vyd. 2016, abs/1603.06937. Dostupné z: http://arxiv.org/abs/1603.06937.

[19] Ng, A. Y. Preventing Överfittingöf Cross-Validation Data. In: Fisher, D. H., ed. *Proceedings of the Fourteenth International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997, s. 245–253. ICML '97. ISBN 1558604863.

[20] Powers, D. Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. *Mach. Learn. Technol.* 1. vyd. Január 2008, zv. 2.

[21] Redmon, J., Divvala, S. K., Girshick, R. B. a Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *CoRR*. 5. vyd. 2015, abs/1506.02640. Dostupné z: http://arxiv.org/abs/1506.02640.

[22] Ren, S., He, K., Girshick, R. B. a Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *CoRR*. 3. vyd. 2015, abs/1506.01497. Dostupné z: http://arxiv.org/abs/1506.01497.

[23] Simonyan, K. a Zisserman, A. Very deep convolutional networks for large-scale image recognition. *ArXiv preprint arXiv:1409.1556*. 1. vyd. 2014.

[24] Suwajanakorn, S., Snavely, N., Tompson, J. J. a Norouzi, M. Discovery of Latent 3D Keypoints via End-to-end Geometric Reasoning. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa Bianchi, N. et al., ed. *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2018, sv. 31. ISBN 9781510884472. Dostupné z: https://proceedings.neurips.cc/paper/2018/file/24146db4eb48c718b84cae0a0799dcfc-Paper.pdf.

[25] Tieleman, T. a Hinton, G. *Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude* [COURSERA: Neural Networks for Machine Learning]. 2012. Dostupné z: http://www.cs.toronto.edu/~hinton/coursera/lecture6/lec6.pdf.

[26] Tompson, J. J., Jain, A., LeCun, Y. a Bregler, C. Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. a Weinberger, K. Q., ed. *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2014, sv. 27. ISBN 9781510800410. Dostupné z: https://proceedings.neurips.cc/paper/2014/file/e744f91c29ec99f0e662c9177946c627-Paper.pdf.

[27] Uhrig, R. E. Introduction to artificial neural networks. In: IEEE. *Proceedings of IECON '95 - 21st Annual Conference on IEEE Industrial Electronics*. 1995, sv. 1, s. 33–37 vol.1. DOI: 10.1109/IECON.1995.483329. ISBN 0-7803-3026-9.

[28] Xiao, J., Russell, B. a Torralba, A. Localizing 3D cuboids in single-view images. In: Pereira, F., Burges, C. J. C., Bottou, L. a Weinberger, K. Q., ed. *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2012, sv. 25. ISBN 9781627480031. Dostupné z: https://proceedings.neurips.cc/paper/2012/file/58238e9ae2dd305d79c2ebc8c1883422-Paper.pdf.

[29] Zayegh, A. a Bassam, N. Neural Network Principles and Applications. In: IntechOpen, ed. *Neural Network Principles and Applications*. November 2018. DOI: 10.5772/intechopen.80416. ISBN 978-1-78984-540-2.

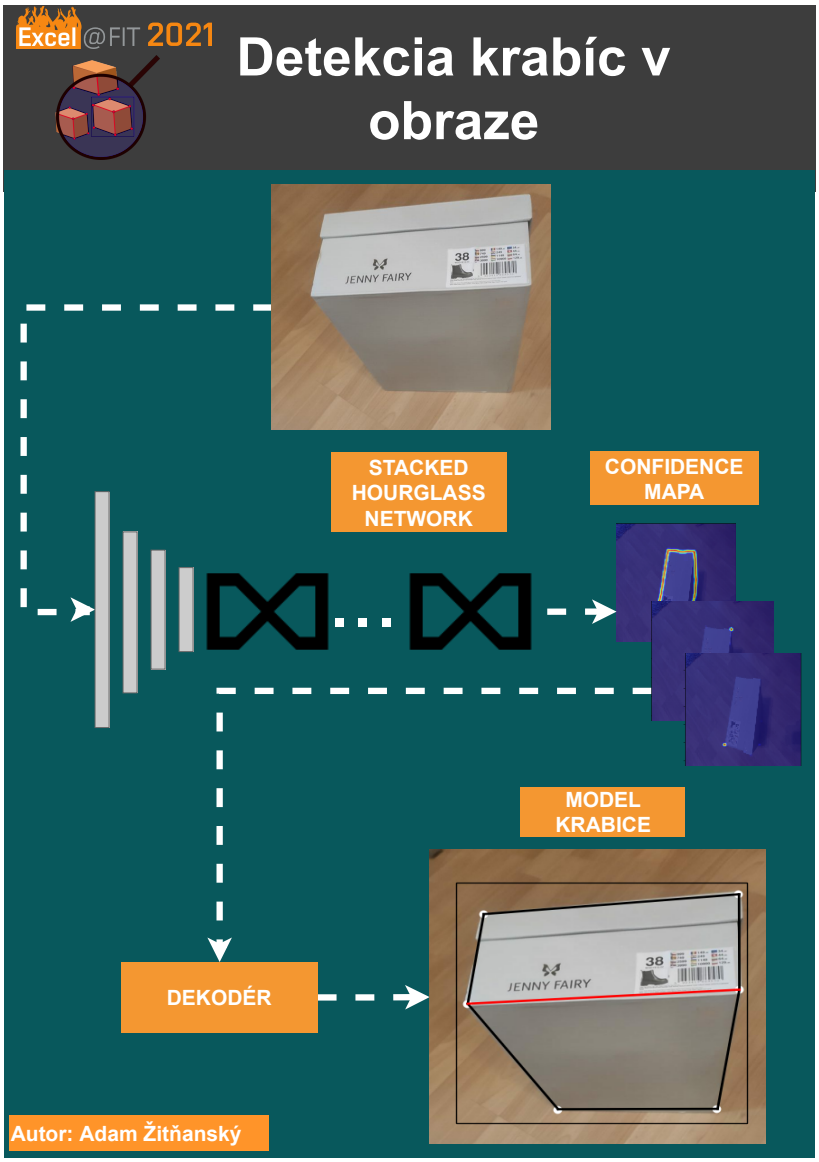
[30] Zhang, Y., Sun, H., Zuo, J., Wang, H., Xu, G. et al. Aircraft Type Recognition in Remote Sensing Images Based on Feature Learning with Conditional Generative Adversarial Networks. *Remote Sensing*. 1. vyd. 2018, zv. 10, č. 7. DOI: 10.3390/rs10071123. ISSN 2072-4292.

[31] Zhao, Z.-Q., Zheng, P., Xu, S.-T. a Wu, X. Object Detection With Deep Learning: A Review. *IEEE Transactions on Neural Networks and Learning Systems*. 2. vyd. 2019, zv. 30, č. 11, s. 3212–3232. DOI: 10.1109/TNNLS.2018.2876865.

[32] Zhou, D., Fang, J., Song, X., Guan, C., Yin, J. et al. IoU Loss for 2D/3D Object Detection. In: IEEE Computer Society. *2019 International Conference on 3D Vision (3DV)*. 2019, s. 85–94. DOI: 10.1109/3DV.2019.00019. ISBN 978-1-7281-3131-3.

Príloha A

Plagát



Obr. A.1: Plagát