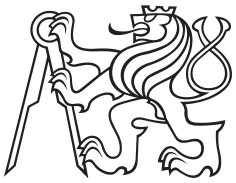
Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra kybernetiky

Algoritmy pro detekci klíčových bodů a jejich vyhodnocení

Matěj Mísař

Vedoucí: Ing. Mgr. Petr Švarný
Květen 2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Mísař** Jméno: **Matěj** Osobní číslo: **492345**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra kybernetiky**

Studijní program: **Kybernetika a robotika**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Algoritmy pro detekci klíčových bodů a jejich vyhodnocení

Název bakalářské práce anglicky:

Keypoint Detection Algorithms and Their Evaluation

Pokyny pro vypracování:

1. Vypracujte rámec umožňující spustit na stejných datech (obrázek i video) vícero algoritmů na detekci klíčových bodů (z pohledu výkonu postačí jejich spuštění “offline”, sekvenčně).
2. Nalezněte, resp. stvořte, vhodný dataset pro porovnání algoritmů.
3. Porovnejte zkoumané algoritmy vícero způsoby, alespoň:
a. Binárně per snímek (detekoval či nedetekoval přítomného člověka dle anotace) a z toho plynoucí průměrná přesnost a podobné míry.
b. Detekce v mezích segmentovaného člověka (např. v rámci bounding boxu).
c. Přesnost detekce jednotlivých keypointů dle metriky (vhodnou metriku nalezněte nebo vytvořte).
4. Využijte porovnání k automatickému anotování vstupních dat (např. pokud se několik modelů shodne, považujeme keypoint za správně nalezený).
5. Směřujte práci, aby mohla být využita vámi či někým dalším k syntéze v podobě tzv. mixture of experts modelu, kdy jsou výstupy jednotlivých algoritmů kombinovány.

Seznam doporučené literatury:

[1] Chen, Yucheng, Yingli Tian, and Mingyi He. "Monocular human pose estimation: A survey of deep learning-based methods." Computer Vision and Image Understanding 192 (2020).
[2] Bishop, Christopher M. Pattern Recognition and Machine Learning. New York :Springer, 2006.
[3] Cao, Zhe, et al. "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields." IEEE transactions on pattern analysis and machine intelligence 43.1 (2019): 172-186.
[4] Güler, Rıza Alp, Natalia Neverova, and Iasonas Kokkinos. "Densepose: Dense human pose estimation in the wild." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
[5] Fang, Hao-Shu, et al. "Rmpe: Regional multi-person pose estimation." Proceedings of the IEEE international conference on computer vision. 2017.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Mgr. Petr Švarný vidění pro roboty a autonomní systémy FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **21.12.2021** Termín odevzdání bakalářské práce: **20.05.2022**

Platnost zadání bakalářské práce: **30.09.2023**

Ing. Mgr. Petr Švarný	prof. Ing. Tomáš Svoboda, Ph.D.	prof. Mgr. Petr Páta, Ph.D.
podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

. _____
Datum převzetí zadání

_____
Podpis studenta

Poděkování

Tímto bych rád projevil své díky Ing. Mgr. Petru Švarnému za odborné vedení a čas, který mi věnoval při tvorbě této bakalářské práce. Dále bych rád poděkoval své rodině za podporu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 20. května 2022

.....
Matěj Mísař

Abstrakt

Detekce klíčových bodů člověka je komplexní úloha, která nalézá využití od animací pro zábavu po interakci člověka a robota.

Cílem této práce je porovnat AlphaPose, Detectron2, MMPose a OpenPose algoritmy pro detekci klíčových bodů lidí z hlediska náročnosti provozu a výkonnosti a zároveň vytvořit základy pro možnost rozšíření práce k syntéze v podobě tzv. mixture of experts modelu.

Naučené algoritmy založené na neuronových sítích využívají nezanedbatelné množství výpočetního výkonu, aby dosáhli nejlepších možných výsledků. Jejich porovnáním cílíme na nalezení celkově nejlepšího algoritmu s minimálními kompromisy.

Ze získaných dat jsme byli schopni určit AlphaPose jako nejvhodnější algoritmus. S touto znalostí předpokládáme přechod ze současně využívaného OpenPose na AlphaPose v naší laboratoři.

Klíčová slova: Detekce klíčových bodů, odhad lidských póz, AlphaPose, DensePose, Detectron2, MMPose, OpenPose

Vedoucí: Ing. Mgr. Petr Švarný

Abstract

Human keypoint detection is a complex task vastly used, ranging from animations in entertainment to human-robot interaction.

The goal of this thesis is to compare AlphaPose, Detectron2, MMPose and OpenPose human keypoint detection algorithms in complexity of operation and performance metrics while establishing a baseline for the possibility of extending the work to synthesis in the form of the so-called mixture of experts model.

Trained algorithms based on neural networks use non-negligible amounts of computing power in order to achieve the best possible performance. A comparison of those algorithms aims to find the best overall one with as few compromises in performance as possible.

We determined that AlphaPose is the best algorithm based on compared data. With this knowledge, we assume the transition from the currently used OpenPose to AlphaPose in our laboratory.

Keywords: Keypoints detection, human pose estimation, AlphaPose, DensePose, Detectron2, MMPose, OpenPose

Title translation: Human keypoint detection algorithms and their evaluation

Obsah

Obrázky

Tabulky

Kapitola 1
Úvod

Detekce klíčových bodů člověka je úloha počítačového vidění, při které dochází k interpretaci lidského těla jako určitého počtu bodů a jejich propojení. Klíčové body označují části těla jako jsou například zápěstí, koleno či rameno.

V současné době se díky svým dobrým výsledkům využívají konvoluční neuronové sítě. Výsledné detekce mají využití ve zdravotnictví, zábavním průmyslu či robotice.

Zároveň začaly vznikat soběstačné algoritmy jako celky obsahující již naučené konvoluční sítě a další funkce (například ukládání vykreslování výsledných detekcí).

Takovéto algoritmy, s již naučenými neuronovými sítěmi, se liší v implementacích a svých schopnostech. Vzhledem k využití této práce v interakci člověka a robota, zohledňujeme důležité faktory, jakými jsou kvalita výsledné detekce a potřebný výpočetní výkon s ohledem na možnost detekce v reálném čase.

Cílem bakalářské práce bylo zprovoznit vybrané algoritmy a otestovat je na nově vytvořeném či vybraném datasetu. Algoritmy bylo následně potřeba vzájemně porovnat a určit nejvhodnější pro interakci člověka a robota.

Jako zdroj vstupních dat předpokládáme záznam z kamery, umístěné na robotovi. Z tohoto důvodu nás zajímají detekce klíčových bodů člověka z obrazu a nikoliv nástroje jako motion capture a jiné. Rovněž neznáme podmínky, ve kterých bude robot fungovat. Proto budeme vyhledávat datasety bez zaměření s vysokou variabilitou.

V práci jsou pojmem algoritmy označovány již vyvinuté a funkční systémy pro detekci klíčových bodů člověka. Předpokládáme, že tyto algoritmy již disponují naučenými neuronovými sítěmi.

Struktura práce je následující: Kapitola 2 seznámí s datasety a představí zvolený dataset, tzv. COCO dataset, pro testování jednotlivých algoritmů. Kapitola 3 je věnována pojmům neuronových sítí, které se v práci objevují. Dále následuje teorie fungování AlphaPose, Detectron2, MMPose a OpenPose algoritmů s jejich vlastnostmi. V Kapitole 4 jsou popsány konkrétní testované konfigurace a modely, metriky pro porovnání a způsoby porovnávání výsledků. Zároveň jsou zde uvedeny vlastnosti validačního setu obrázků, který je využit k porovnání výsledných detekcí jednotlivých algoritmů. Kapitola 5 obsahuje výsledky a jejich vyhodnocení v rámci jednotlivých metrik. V předposlední

Kapitole 6 jsou řešeny nedostatky, objevené během zkoumané problematiky. Výsledky práce jsou shrnuty v poslední části práce, Kapitole 7.

Kapitola 2
Dataset

Současné algoritmy pro odhad lidských póz založené na neuronových sítích závisí zásadně na úspěšném natrénování. Trénovací data proto tvoří jeden z hlavních prvků při cestě ke kvalitnímu detektoru.

Často jsou datasety zaměřené na specifickou doménu (například dataset sportovních obrázků). Vzhledem k naší motivaci, nalezení nejvhodnějšího algoritmu na detekování klíčových bodů osob, se zaměříme na datasety určené právě k takové detekci bez konkrétního zaměření.

Trénovací nebo validační datasety jsou tvořené sbírkou dvojic obrázku a polem souřadnic klíčových bodů každé anotované osoby. Vhodný obecný dataset na natrénování algoritmů by měl poskytnout velký počet různorodých obrázků s dostatečným počtem anotací.

Při učení obecného detektoru klíčových bodů člověka je potřeba pokrýt velké spektrum obrázků, neobvyklých poloh osob, různorodých druhů oblečení či zakrývání různých částí těla. Při trénování neuronových sítí se navíc za účelem navýšení kvantity trénovacích dat využívá jejich umělá úprava v podobě pootočení, oříznutí a dalších metod.

Rozšířené datasety využívané k trénování detektorů lidských klíčových bodů jsou například MPII [1], COCO [2], COCO-WholeBody [3] a Halpe [4] [5]. Tyto datasety se liší primárně kvantitou svých dat a počtem klíčových bodů, ze kterých je sestavena kostra člověka.

Kostra člověka může být v jednotlivých datasetech tvořena různým množstvím klíčových bodů a jejich propojení. Zmíněné datasety používají kostry složené z následujících počtů klíčových bodů: MPII (15 klíčových bodů), COCO (17 klíčových bodů), COCO-WholeBody (133 klíčových bodů), Halpe (26 klíčových bodů nebo 136 klíčových bodů).

COCO dataset se 17 klíčovými body byl zvolen pro otestování a porovnání vybraných algoritmů. Tento formát byl již v laboratoři využíván a 17 klíčových bodů považujeme za dostatečné množství k pokrytí hlavních částí lidského těla, viz Obrázek 2.1. Tyto klíčové body se nachází v oblasti nosu, očí, uší, ramen, loktů, zápěstí, pasu, kolen a kotníků.

Za účelem vytvoření datasetu s konkrétní tématikou je možné využít programy jako jsou například COCO Annotator [6] nebo CVAT [7]. Ty usnadní formátování anotací, ovšem označení jednotlivých klíčových bodů či ohraničujících boxů probíhá manuálně.

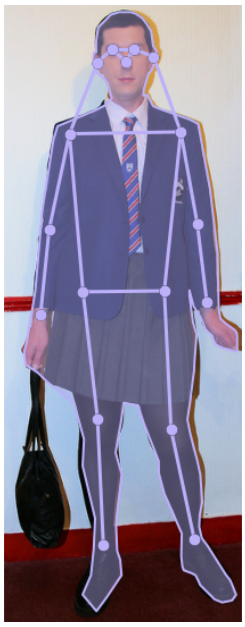
2.1 COCO

MS COCO [2] (*Microsoft Common Objects in Context*) je rozsáhlá datová sada obsahující detekce a segmentace objektů. Celý dataset obsahuje přes 330 tisíc obrázků, na kterých se nachází dohromady více než 250 tisíc lidí.

Součástí COCO datasetu je projekt [8] zastřešující soutěže v různých detekčních úlohách. Jednou ze soutěží je úkol detekce kostry člověka tvořené ze 17 klíčových bodů.

COCO dataset na rozpoznání 17 klíčových bodů lidí obsahuje různorodé obrázky osob v různých polohách a aktivitách. Součástí datasetu jsou i obrázky každodenních předmětů nebo zvířat, které neobsahují osoby. Tato část datasetu obsahuje 118 tisíc obrázků určených k natrénování algoritmů a 5 tisíc validačních obrázků.

V porovnání používáme formát kostry z COCO datasetu složené ze 17 klíčových bodů (viz Obrázek 2.1), nadále budeme tento formát označovat jednoduše jako COCO formát.



Obrázek 2.1: Kostra člověka složená ze 17 klíčových bodů podle formátu COCO. Obrázek 000000042102.jpg převzat z [8].

Anotace COCO datasetu jsou uváděny ve formátu $[x_1, y_1, v_1, ..., x_k, y_k, v_k]$, kde x a y jsou souřadnice klíčových bodů. Argument v představuje stav viditelnosti daného klíčového bodu a je definován jako $v = 0$: bod není označený, $v = 1$: bod je označený, ale není viditelný, $v = 2$: bod je označený a viditelný.

K vyhodnocení soutěže detekce klíčových bodů je využívána metrika OKS (*object keypoint similarity*), která zachycuje obdobu k průměrné přesnosti (*average precision*) a průměrné senzitivitě (*average recall*). Představení vztahu pro výpočet OKS metriky se nachází v Oddíle 4.2.3.

Kapitola 3

Algoritmy pro detekci klíčových bodů

Cílem práce je nalezení vhodného algoritmu pro využití ve výzkumu interakce člověka a robota. Z tohoto důvodu je vyžadována možnost algoritmus modifikovat a případně i přeučit. Proto je práce zaměřena pouze na veřejně dostupné open-source projekty, které jsou aktivně vyvíjeny.

Vzhledem k neznámému prostředí, ve kterém bude robot umístěn, a z důvodu robustnosti, budeme předpokládat, že se na kameře může objevit větší počet lidí. Detekce klíčových bodů většího počtu lidí s sebou nese mnoho nových výzev: neznámý počet lidí, různorodé polohy končetin či překryv sousedících osob. Všechny tyto proměnné přidávají na složitosti. [9]

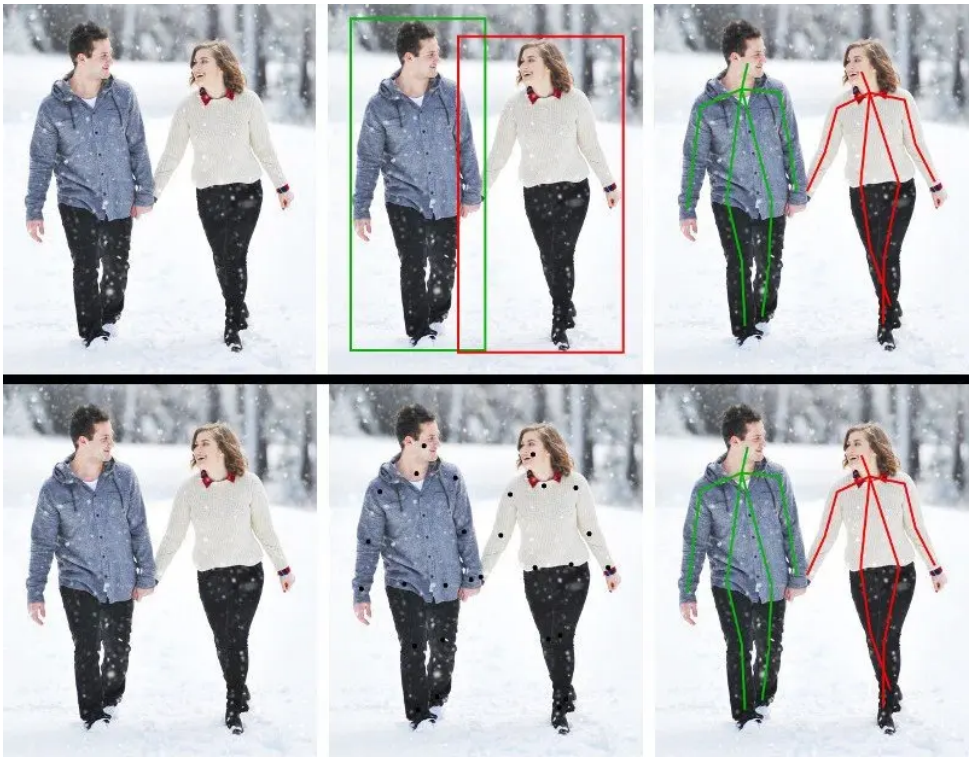
Při detekci se využívá jedna ze dvou metod. Metoda shora-dolů [9] (*top-down*) využívá objektové detektory následované hledáním klíčových bodů nalezených osob. Detektor nejprve nalezne jednotlivé osoby a ohraničí je pomocí ohraničujících boxů (*bounding box*). Poté v rámci jednotlivých boxů určuje klíčové body daného člověka. Pomocí této metody dochází k rozložení jednoho komplexního problému na řadu méně obtížných úloh.

Metoda zdola-nahoru [9] [10] (*bottom-up*) využívá komplexnější přístup k detekci klíčových bodů, kdy detektor nejprve nalezne jednotlivé klíčové body všech osob v obrázku a následně tyto body spojí v kostry osob.

Během rešerše jsme se setkali s algoritmy jako jsou například Xnect nebo DeeperCut, které ovšem nesplnily náš požadavek aktivního vývoje. Další skupinou algortimů jsou například Pose Estimator 2 [11] nebo Microsoft Kinect [12], které nejsou open-source.

Nakonec jsme vybrali čtyři algoritmy (AlphaPose, Detectron2, MMPose, OpenPose), které jsou aktivně vyvíjeny, open-source, obsahují naučené modely s COCO formátem (viz Podkapitola 2.1) a dokáží zpracovat barevné obrázky a videa. V následujících podkapitolách budou jednotlivé algoritmy více popsány.

Všechny vybrané algoritmy podporují fungování jak na samotných procesorech, tak na grafických kartách.



Obrázek 3.1: Nahoře: postup metody shora-dolů, dole: postup metody zdola-nahoru. Převzato z [13].

3.0.1 Trackování

Určování póz může být posíleno pomocí trackování [14]. Tato úloha spojuje detekci klíčových bodů osob se schopností přiřazovat unikátní identifikátory jednotlivým detekovaným osobám v sérii obrázků, resp. ve video záznamech.

Trackování k fungování využívá informace získané v průběhu zpracování videa. Oproti tomu běžné detektory bez schopnosti trackování zpracovávají video v rámci jednotlivých snímků bez uchování informace o osobách.

Trajektorie, vzniklé sledováním pohybů, lze využít například ve zdravotnictví, animacích či k rozpoznání gest a lidských činností.

Protože trackování využívá sérii obrazů, resp. video, není v tomto výzkumu využito. Přesto bude u algoritmů uvedeno, pokud trackování umožňují s ohledem na to, že se v aplikaci předpokládá užití videa jako zdroje obrazu.

3.1 Pojmy neuronových sítí

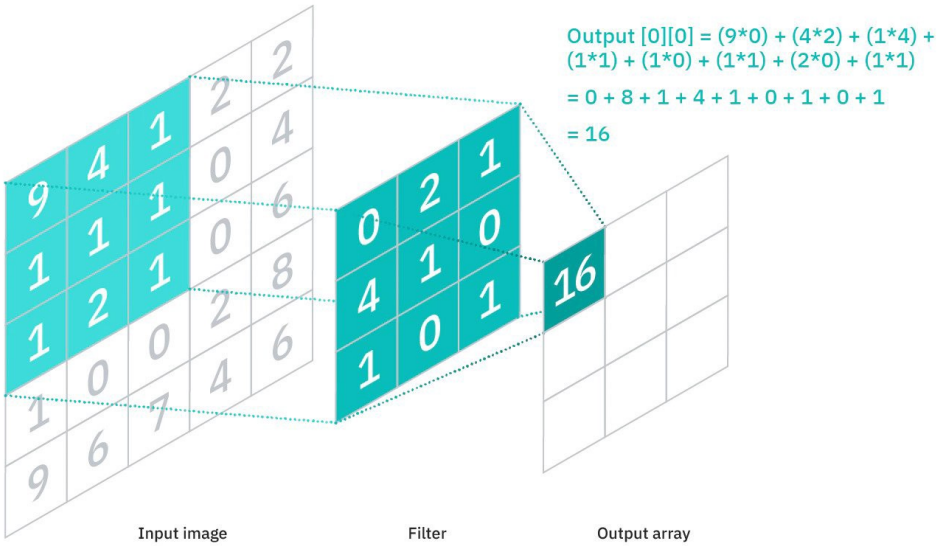
Následující podkapitola obsahuje pojmy spojené s neuronovými sítěmi, které se vyskytují v průběhu práce, a seznámení s vybranými algoritmy.

3.1.1 Konvoluční síť

K detekci v obrazových, resp. video, záznamech se využívají konvoluční neuronové sítě [15] nebo zkráceně konvoluční sítě, které mohou sloužit k nalezení objektů či pouze určení, zda se objekt na záznamu nachází.

K fungování konvoluce dochází pomocí filtru neboli jádra (*kernel*), které má zpravidla menší rozměry než vstupní snímek. Vstupní obrázek je postupně zpracováván v jednotlivých částech rozměru jádra.

Výstupu dat z konvoluční vrstvy se říká příznaková mapa (*features map*). Počáteční vrstvy neuronové sítě dokáží rozpoznat jednodušší elementy obrázku a jak postupně data z původního obrázku prostupují dalšími vrstvami jsou sítě schopné rozpoznávat složitější prvky. [16]



Obrázek 3.2: Grafické znázornění zpracování vstupního obrázku pomocí jádra v konvoluci. Převzato z [16].

V Obrázku 3.2 dochází k vynásobení jednotlivých hodnot vstupního obrázku hodnotami jádra (zde označen jako Filter). Ty jsou následně sečteny v jednu hodnotu, která je zapsána do příznakové mapy. Konkrétní výpočet lze vidět v právem horním rohu.

Plně konvoluční síť (*fully convolutional network*) se od konvoluční sítě liší v operacích, které obsahuje. Plně konvoluční síť obsahuje pouze konvoluce, které buď snižují (*down sampling*) nebo zvyšují rozlišení (*up sampling*).

Deformační konvoluční sítě [17] (*DCN*) představují alternativu konvolučních sítí, které mají standardně osově a středově symetrické jádro. DCN mají deformované jádro, které není osově nebo středově, případně osově ani středově, symetrické.

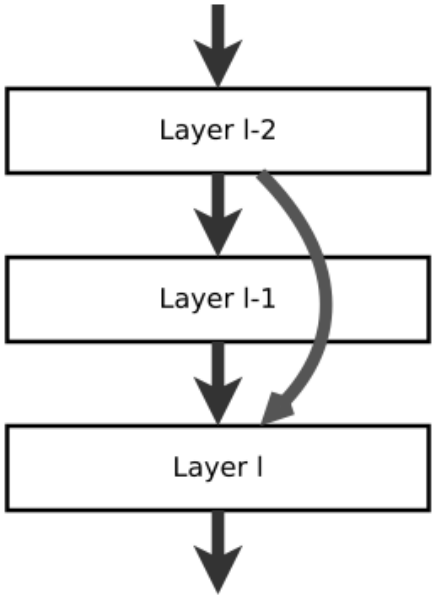
3.1.2 ResNet

ResNet [18] neboli "Residual Network" je hluboká neuronová síť, která byla v roce 2015 představena vědci z Microsoft Research. Nese označení hluboká, neboť obsahuje velké množství vrstev. ResNet přinesl nové možnosti v odvětví strojového vidění.

Hlavním cílem ResNet bylo zvýšení počtu vrstev v síti, které přinášejí větší preciznost detekcí, společně se schopností jej snadno naučit.

Pro zjednodušení učení sítě využívá ResNet takzvaná "skoková propojení" (*skip connections*). Tato "skoková propojení" přeskakují jednu nebo více vrstev v síti, viz Obrázek 3.3.

Konkrétní ResNet bývá uváděn jako ResNetX, kde X představuje počet vrstev sítě, případně jako ResNet-wX, kde X značí počet kanálů podsítí s vysokým rozlišením v posledních třech fázích. [19] Počet kanálů představuje třetí rozměr konvolučního jádra.



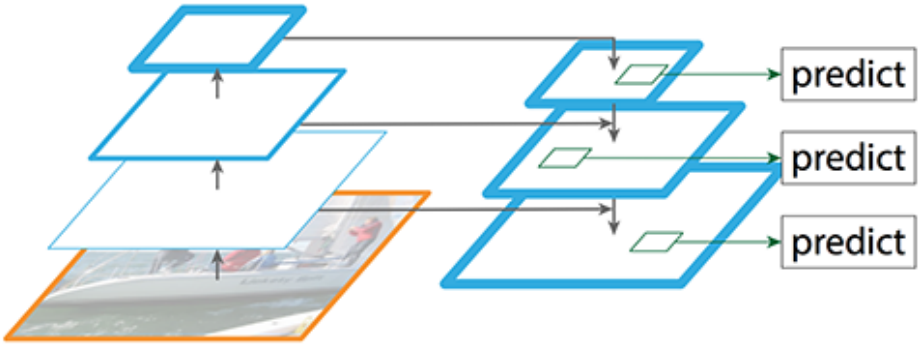
Obrázek 3.3: Grafické znázornění přeskočení jedné vrstvy v síti pomocí "skip connection". Převzato z [20].

3.1.3 Feature Pyramid Network

Feature Pyramid Network [21] (*FPN*) je komponenta extrahující příznaky obrázků na několika úrovních. Svým postupem fungování připomíná při grafickém znázornění pyramidu.

Jak je vidět na Obrázku 3.4, zpracování postupuje ve dvou směrech. V prvním směru nahoru dochází ke snižování rozlišení. Postupně je síť schopna rozpoznat složitější prvky jak bylo zmíněno výše (viz Oddíl 3.1.1). Druhým směrem dolů dochází ke zvyšování rozlišení, a kvůli lepšímu umístění detekovaných objektů do původního obrázku jsou přidána "skoková propojení"

podobná těm z ResNetu.



Obrázek 3.4: Grafické znázornění postupu zpracování obrázku pomocí Feature Pyramid Network. Vlevo směrem nahoru dochází ke snižování rozlišení, Vpravo směrem dolů dochází ke zvyšování rozlišení. Převzato z [18].

V naší práci je FPN využívána v algoritmu Detectron2 pro detekci klíčových bodů člověka, viz Podkapitola 3.3.

3.1.4 R-CNN

R-CNN (*Region Based Convolutional Neural Network*) je konvoluční síť zaměřená na objektové detekce. Princip jejího fungování stojí na rozdělení vstupního obrázku na řadu menších oblastí, ve kterých hledá objekty. Tento přístup slibuje rychlejší fungování oproti klasické konvoluční síti.

Jedna ze sítí založených na variantě R-CNN je Mask R-CNN [22]. Tato síť detekuje objekty v podobě ohraničujících boxů jako R-CNN, ovšem navíc generuje jejich segmentační masky.

Poslední sítí s původem u R-CNN, která se vyskytuje v naší práci, je Keypoint R-CNN [23]. Tato síť má stejnou architekturu jako Mask R-CNN, ale místo segmentačních masek generuje klíčové body.

3.1.5 Spacial Transformer Network

Konvoluční sítě samy o sobě mají omezené možnosti být prostorově nezávislé vůči vstupním datům. Řešení přináší Spacial Transformer Network [24] (*STN*), která dovoluje transformaci vstupních dat nebo příznakových map v podobě rotace, oříznutí nebo škálování. Transformace má za cíl upravit data do očekávané, kanonické, podoby před vstupem do sítě.

3.2 AlphaPose

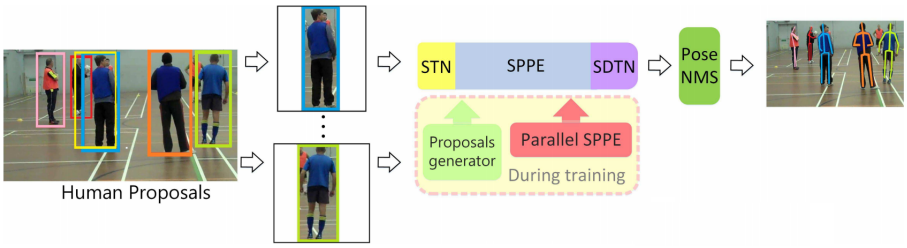
AlphaPose [25] je systém k odhadu lidských póz představený v roce 2017 a je založený na lokálních odhadech lidských póz (*Regional Multi-person Pose estimation*). Jako první na světě přesáhl 70 % v průměrné přesnosti (*mean average precision*).

3.2.1 Popis

K fungování využívá metodu shora-dolů, která rozděluje určení kostry člověka na dva kroky. V prvním kroku dojde k vymezení osob pomocí ohraničujících boxů za využití detektoru osob. V druhém kroku dochází k lokalizaci jednotlivých bodů člověka a jejich spojení v pózu.

Druhý krok metody shora-dolů je v systému AlphaPose implementován pomocí SPPE (*single-person pose estimator*) a Symetrického STN (*spacial transformer network*), který je složený z prostého STN a SDTN (*spacial de-transformer network*). Ve fázi trénování je k Symetrickému STN paralelně přidána SPPE vrstva jako dodatečná regularizace. Symetrické STN a paralelní SPPE mají za cíl zlepšit fungování SPPE v hlavní větvi, do kterého vstupují osoby v detekovaných ohraničujících boxech zatížené potenciálními chybami.

Výstupem z SPPE jsou návrhy lidských póz. Během procesu mohou vzniknout redundantní návrhy, které jsou pomocí Pose NMS (*non-maximum suppression*) eliminovány. Pose NMS zvolí návrh pózy s největší důvěryhodností jako referenci a porovnává vůči němu návrhy póz v jeho blízkosti. Návrhy, které spadají do eliminačních kritérií jsou vyřazeny a postupně zůstanou pouze unikátní pózy.



Obrázek 3.5: Ilustrace rámce fungování AlphaPose. Vlevo vstupují ohraničené osoby do SPPE a Symetrického STN (případně ve fázi testování vstupují zároveň do paralelní SPPE vrstvy). Výstupem jsou návrhy póz, které jsou v případě redundance eliminovány pomocí Pose NMS. Převzato z [25].

Při trénovaní sítě je dataset rozšířen pomocí PGPG (*pose-guided proposals generator*). PGPG využívá znalost ohraničujícího boxu k vygenerování nových póz a tedy nových syntetických trénovacích dat. Rozšíření dat usnadňuje algoritmu naučit se předpovídat různorodé pózy a korektně určit části lidského těla.

3.2.2 Vlastnosti

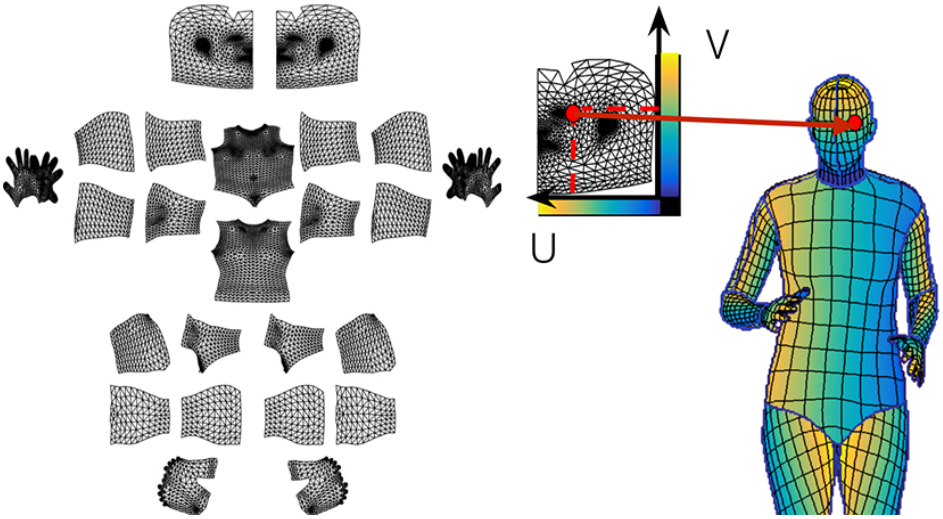
AlphaPose [4] nativně obsahuje metodu pro ukládání výsledných detekcí v souřadné soustavě pro datasety COCO, COCO-WholeBody, Halpe. Součástí repositáře je metoda pro trackování lidí zvaná PoseFlow.

3.3 Detectron2

Detectron2 [26] je knihovna zastřešující řadu projektů. V rámci těchto projektů obsahuje řadu nejmodernějších algoritmů detekcí a nástrojů na segmentace. Od roku 2020 nahrazuje předchozí verzi Detectronu.

Jedním z podporovaných projektů je DensePose, fungující pomocí metody shora-dolu. DensePose se snaží o nalezení souvislostí mezi 2D obrázkem a 3D povrchem lidského těla. Výsledná detekce povrchu těla je rozdělena na 24 částí (hlava, tělo, nohy, ruce, atd.) a každá část je parametrizována dvěma souřadnicemi (u, v).

Ve zbytku práce je názvem Detectron2 označován projekt DensePose.



Obrázek 3.6: Rozdělení povrchu těla a souřadnice reprezentující jednotlivé části. Převzato z [27].

3.3.1 Popis

Společně s DensePose [27] byl za účelem určování povrchu lidského těla vytvořen COCO-DensePose dataset. Tento dataset obsahuje 50 000 obrázků z COCO datasetu, u kterých byly manuálně anotovány i povrchy lidských těl.

DensePose funguje pomocí systémů založených na konvolučních neuronových sítích (*CNN*), naučených pomocí zmíněného COCO-DensePose datasetu. Ty mají za úkol rozpoznat lidskou osobu od pozadí a určit povrch lidského těla. DensePose využívá již dříve vyvinutý Dense Regression (*DenseReg*) systém, který stojí na plně konvoluční neuronové síti. Tento systém je naučen na mapování 3D mřížky na povrch lidského těla. Kombinací s Mask-RCNN architekturou založenou na lokálních detekcích vzniká výsledný systém s názvem DensePose-RCNN.

I přesto, že cílem projektu DensePose v rámci Detectron2 je nalezení spojitostí mezi 2D obrázky a 3D povrchy lidských těl, dokáže detekovat klíčové body lidí. Z toho důvodu je Detectron2, resp. DensePose, zahrnut v práci.

3.3.2 Vlastnosti

Demonstrační skript, který je součástí oficiálního online repositáře, nativně neobsahuje metodu pro uložení výsledných klíčových bodů. Naučené modely podporují výstup ve formátu COCO (viz Podkapitola 2.1). Detectron2 nepodporuje možnost trackování lidí.

3.4 MMPose

"MMPose je sada nástrojů, vytvořená k odhadu póz osob založená na PyTorch knihovně." [28] První veřejná verze (0.5.0) vyšla v roce 2020.

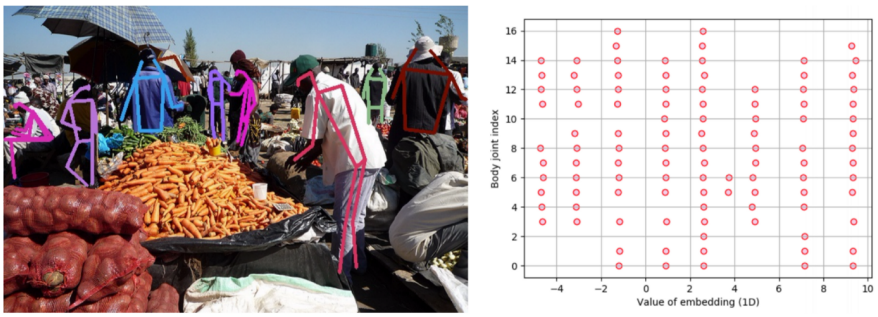
Oproti Detectron2, který detekuje mimo jiné různorodé objekty a předměty, se MMPose zaobírá pouze určováním klíčových bodů lidí či zvířat. V balíčku jsou implementovány obě metody k nalezení klíčových bodů (shora-dolu, zdola-nahoru), mezi kterými si může uživatel vybrat. MMPose zároveň obsahuje velký počet algoritmů (HRNet, Hourglass, DeepPose, Associative Embedding a další).

K testování jsme zvolili konfiguraci MMPose s HRNet (*High-Resolution Network*) a asociativním vnořením (*Associative Embedding*), které jsou později v práci představeny podrobněji.

3.4.1 Popis

Metoda Associative Embedding [29] představuje dohled nad konvolučními neuronovými sítěmi. Lze implementovat do libovolné sítě, která vytváří předpověď vzhledem k pixelům.

Svým postupem je podobná metodě zdola-nahoru, která rozděluje detekování klíčových bodů a jejich řazení do skupin na více kroků. Oproti klasické metodě zdola-nahoru Associative Embedding učí sítě současně v jednom kroku určovat klíčové body osob a řadit je do skupin.

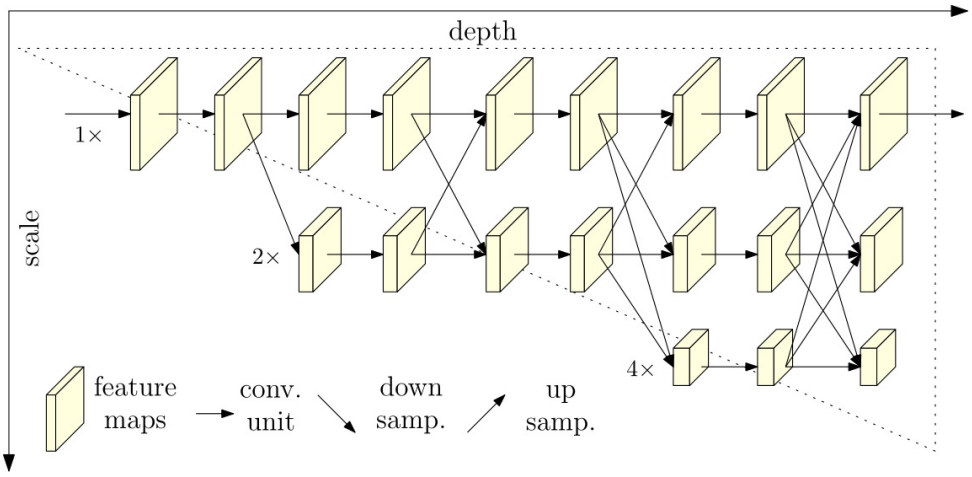


Obrázek 3.7: Graficky znázorněné rozdělení štítků klíčových bodů. Vlevo jsou klíčové body spojené v kostry osob, vpravo jsou klíčové body přiřazeny k odpovídajícím hodnotám štítků. Převzato z [29].

Během hledání klíčových bodů osob dochází zároveň k předpovědi štítků a teplotních map (*heatmap*). Teplotní mapa obsahuje vysoké hodnoty v

místech s největší pravděpodobností výskytu klíčového bodu. V případě více osob dochází k přiřazování klíčových bodů k odpovídajícím osobám pomocí štítků. Ty by měly mít stejné hodnoty v případě, že klíčové body patří ke stejné osobě.

HRNet je jedna z nejnovějších neuronových sítí. Oproti ostatním sítím, které zpravidla snižují rozlišení obrázku v průběhu zpracování, HRNet zachovává vysoké rozlišení v hlavní větvi a k přechodu od vysokého rozlišení k nízkému dochází pouze v paralelních větvích. V architektuře sítě je rovněž využito slučování mezi nízkými a vysokými rozlišeními. Zachování vysokého rozlišení přispívá k prostorově přesnější teplotní mapě klíčových bodů.



Obrázek 3.8: Ilustrace architektury sítě HRNet. V hlavní větvi je zachováno rozlišení shodné se vstupním obrázkem, ke snižování rozlišení ("down sampling") a následně k jeho zvyšování ("up sampling") dochází ve vedlejších větvích. Převzato z [30].

3.4.2 Vlastnosti

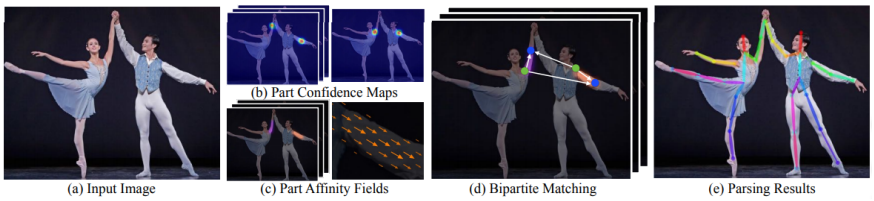
MMPose [28] podporuje 2D odhad poloh osob, 2D odhad poloh rukou, detekci orientačních bodů obličeje, odhad polohy člověka s až 133 klíčovými body, povrch lidského těla a odhad póz zvířat. Postup odhadování lze volit mezi oběma zmiňovanými metodami (shora-dolů, zdola-nahoru). Sada nástrojů podporuje řadu výstupních formátů jako jsou například COCO, AIC, MPII, MPII-TRB, OCHuman a jiné. Demonstrační skript nativně neobsahuje metodu pro uložení výsledných detekcí. Trackování lidí umožňuje separátní MMTracking [31] balíček.

3.5 OpenPose

OpenPose [32] je algoritmus obsahující detektor až 135 klíčových bodů osob v reálném čase. Systém s využitím metody zdola-nahoru slibuje konstantní čas potřebný pro detekci bez ohledu na počet osob na obrázcích.

3.5.1 Popis

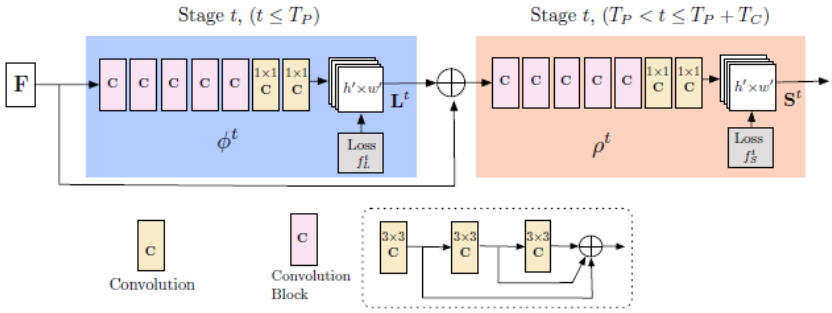
Metoda zdola-nahoru slibuje šetření časové náročnosti, díky zpracování celého vstupu najednou. Nejprve je implementována konvoluční neuronová síť (upravených prvních 10 vrstev VGG-19 sítě, což je konvoluční síť složená z 19 vrstev), která má za úkol vygenerovat mapy příznaků (*feature maps*).



Obrázek 3.9: Postup detekce OpenPose. (a) Obrázek vstupující do konvoluční sítě (b) Teplotní mapy pro jednotlivé části těla (c) PAFy (d) Bipartitní párování (e) Výsledná detekce. Převzato z [32].

Dále mapy s příznaky vstupují do vícestupňové konvoluční neuronové sítě. V první části sítě dochází ke generování 2D vektorů s názvem PAF (*Part Affinity Field*). Tyto vektory v sobě obsahují zakódované informace o orientaci a poloze jednotlivých "končetin"[1]. V druhé části tato síť předpovídá teplotní mapy.

Získané PAFy a teplotní mapy jsou využity k asociaci jednotlivých bodů za využití bipartitního párování a sestavení kostry osoby. Bipartitní párování spojuje klíčové body jedné části těla s klíčovými body jiné části těla, které s ní mohou sousedit (např. loket může být spojen se zápěstím nebo ramenem v COCO formátu, viz Podkapitola 2.1).



Obrázek 3.10: Ilustrace OpenPose architektury vícestupňové konvoluční sítě. První modrá část generuje PAFy a druhá oranžová část předpovídá teplotní mapy. Převzato z [32].

3.5.2 Vlastnosti

OpenPose obsahuje funkci pro ukládání výsledných detekcí. Součástí systému je volitelné trackování jedné osoby. Systém obsahuje naučené modely s výstupy ve formátech COCO nebo MPII.

[1] Označení pro všechny části klíčových bodů včetně nosu, očí a uší.

Kapitola 4

Metoda porovnání algoritmů

V rámci práce jsme otestovali 4 již představené algoritmy. Tři z nich byly testovány na stejném zařízení, zatímco jeden byl testován na jiném. V této kapitole jsou uvedeny specifikace obou strojů společně s konkrétními konfiguracemi algoritmů. Dále je zde blíže představen vybraný dataset pro testování a metriky, ve kterých byly vybrané algoritmy porovnány. V poslední části kapitoly jsou uvedeny kroky předcházející získání výsledných dat a následně způsob, jakým byly výsledky porovnány.

AlphaPose, Detectron2 a MMPose byly testovány na počítači Lenovo IdeaPad S540-15IWL s čtyř jádrovým procesorem Intel Core i7-8565U, 12 GB DDR4 RAM paměti a dedikovanou grafickou kartou NVIDIA GeForce GTX 1650 Mobile / Max-Q (TU117M) 4 GB. Na počítači je nainstalován operační systém Linux Mint ve verzi 20.2 Cinnamon 64bit.

Openpose byl testován na stroji Dell Inspiron 7577 s osmi jádrovým procesorem Intel Core i7-7700HQ, 16 GB DDR4 RAM a dedikovanou grafickou kartou NVIDIA GeForce GTX 1060 Max-Q 6 GB s operačním systémem Linux Ubuntu 16.04 64bit.

AlphaPose byl testován v konfiguraci se sítí Fast Pose (DCN) založené na ResNet50 - dcn, která dle specifikací disponuje druhou nejvyšší průměrnou přesností a rychlostí z ostatních konfigurací. FastPose je síť vytvořená autory AlphaPose, která dosud není oficiálně zdokumentovaná ani publikována[33].

Konfigurace pro Detectron2 byla zvolena s Keypoint R-CNN založené na ResNet50 s FPN. Model je v oficiálním repozitáři Detectron2 označen identifikátorem 137849621. Tato konfigurace byla zvolena díky nejmenšímu času potřebnému na zpracování jednoho obrázku a obdobnou průměrnou přesností jako ostatní konfigurace.

MMPose byl spuštěn s metodou Associative Embedding a sítí HRNet-W32. Konfigurace byla zvolena z důvodu otestování algoritmu s metodou zdola-nahoru, který představuje alternativu k OpenPose. Tato konfigurace disponuje nejrychlejším zpracováním obrázků za obdobné průměrné přesnosti s dalšími konfiguracemi.

OpenPose běžel ve výchozí konfiguraci s doporučenými parametry. Parametr, který je doporučeno změnit pro zlepšení výkonu OpenPose je snížit počet detekovaných osob na 1 nebo používat model BODY_25. Tyto změny však v našem kontextu nešlo použít.

4.1 Dataset

Pro případy, kdy je známý cíl nasazení detektoru, je možné vytvořit vlastní dataset zaměřený na konkrétní scény. Porovnání algoritmů na takovém datasetu by představovalo výsledky bližší následné reálné implementaci. Sběr dostatečného počtu obrázků a následná manuální anotace jsou ovšem časově náročné a předpokládáme, že by specializovaný dataset neměl na porovnání algoritmů vliv.

COCO dataset s kostrou člověka sestavenou ze 17 klíčových bodů se již v laboratoři využíval, a proto byl zvolen pro otestování algoritmů. Obsahuje 5 000 validačních obrázků s 11 004 anotovanými osobami ve variabilních prostředích. Formát kostry se 17 klíčovými body je nativně rozšířen mezi všemi testovanými algoritmy. COCO formát pokrývající tělo člověka 17 klíčovými body považujeme z hlediska využití v interakci člověka a robota za dostatečný.

4.2 Metriky

Neboť cílem této práce je nalezení nejvhodnějšího algoritmu pro detekci klíčových bodů při interakci člověka a robota, zajímá nás jak jejich složitost zprovoznění a náročnost provozu, rozebrané v Oddílech 4.2.1 a 4.2.2, tak spolehlivost jejich detekcí, viz Oddíl 4.2.3.

4.2.1 Instalace

AlphaPose, Detectron2 a MMPose byly nainstalovány na počítači s operačním systémem Linux Mint 20.2 Cinnamon. Instalace jednotlivých algoritmů proběhla v rámci individuálních prostředí pomocí správce prostředí Conda [34].

Hodnocení obtížnosti instalace je rozděleno na tři úrovně: lehká, střední a těžká obtížnost. Nejlepší, lehká obtížnost, označuje přímé postupování dle oficiálního návodu bez výskytu problému. Střední náročnost představuje výskyt problému během instalace, který je vyřešen v rámci "Issues" fóra nebo podobného oficiálního zdroje. Těžká obtížnost značí potřebu odchýlení se od přiloženého postupu instalace, chybějící dokumentaci instalace nebo výskyt problému, který dosud nebyl oficiálně řešen.

4.2.2 Časová a paměťová náročnost

Paměťová náročnost je v případě omezeného výpočetního výkonu důležitým faktorem. Zajímá nás, zda je nutné mimo robota užít také externí stroj na zpracování zdrojových dat z kamery.

Časová náročnost je rovněž důležitá k ověření, zda jsou algoritmy schopny detekovat v reálném čase bez prodlev při interakci s robotem. Předpokládáme, že kvalitní algoritmus bude schopen zpracovat data za krátký čas s přiměřenými paměťovými prostředky.

Obě metriky byly zaznamenány pomocí Python modulu Memory Profiler [35]. Modul zaznamenává čas a využití paměti hlavním procesem a jím

vytvářených vedlejších procesů. Hodnoty využití paměti byly porovnány se systémovým monitorem a `htop` příkazem v operačním systému Linux. Zaznamenané časy běhů byly porovnány s časovači, které jsou součástí testovaných algoritmů. Díky srovnatelným hodnotám v obou případech považujeme Memory Profiler za věrohodnou metodu pro záznam dat k porovnání.

4.2.3 Hodnocení detekcí

Výsledky jednotlivých detektorů byly porovnány vůči anotacím z datasetu. Porovnáním získáváme informace o tzv. "binárních" detekcích, průměrných chybách klíčových bodů v pixelech, průměrných relativních chybách v procentech a OKS hodnotách (viz rovnice 4.4).

Součástí každého výsledného klíčového bodu je také skóre, které označuje míru, se kterou si je algoritmus jistý, že se daný klíčový bod nachází na daných souřadnicích. Skóre nebylo použito v porovnání neboť není unifikované mezi algoritmy.

Binární detekce. Rozděluje výsledek algoritmů na naše tři základní hodnoty:

Falešně negativní detekce představují anotované osoby, které k sobě nemají detekovaný protějšek.

Pravdivě pozitivní detekce jsou detekce, které obsahují alespoň jeden klíčový bod uvnitř ohraničujícího boxu anotace a jsou s ní spárovány.

Falešně pozitivní detekce jsou detekce, které nemají vhodný protějšek mezi anotacemi.

Z těchto hodnot lze následně získat naše dvě metriky:

Průměrná přesnost je poměr správně určených binárních detekcí vůči všem detekcím daného algoritmu. Získá se ze vztahu:

$$\text{Průměrná přesnost} = (\sum_{i=1}^{n} \frac{TP_i}{TP_i + FP_i}) \cdot \frac{1}{n}, \tag{4.1}$$

kde TP_i označuje pravdivě pozitivní detekce v i-tém obrázku, FP_i jsou falešně pozitivní detekce v i-tém obrázku a n je v našem případě počet validačních obrázků.

Průměrná senzitivita je poměr správně určených binárních detekcí proti všem osobám na obrázcích. Získá se ze vztahu:

$$\text{Průměrná senzitivita} = (\sum_{i=1}^{n} \frac{TP_i}{TP_i + FN_i}) \cdot \frac{1}{n}, \tag{4.2}$$

kde FN_i je počet falešně negativních detekcí v i-tém obrázku.

Pokud součty $TP_i + FP_i$ nebo $TP_i + FN_i$ jsou rovny 0, dochází k záměně hodnoty přesnosti, případně senzitivity, za hodnotu 1 značící bezchybné detekování obrázku.

Naše metriky průměrné přesnosti a senzitivity nevyužívají hodnot překrytí ohraničujících bodů (IoU [36]), které je rozebráno níže. Jelikož se chceme primárně soustředit na porovnání jednotlivých klíčových bodů, vytvořili jsme

vlastní alternativu průměrné přesnosti a senzitivity, která je zde popsána a využívána.

Průměrná absolutní chyba klíčových bodů. Tato představuje součet Euklidovských vzdáleností v pixelech mezi anotovanými a detekovanými klíčovými body, vydělený součtem osob. Euklidovská vzdálenost je získána ze vztahu

$$\mathrm{d} = \sqrt{(x_i - x_{ai})^2 + (y_i - y_{ai})^2}, \tag{4.3}$$

kde x_i a y_i jsou detekované i-té klíčové body, x_{ai} a y_{ai} jsou i-té klíčové body anotace.

Absolutní chyba detekovaného klíčového bodu od anotace není vhodná metrika na porovnání. V případě, že bude osoba zabírat většinu obrázku, může větší odchýlení od správných souřadnic stále znamenat správné určení části těla. V opačném případě, kdy se bude osoba nacházet na malé ploše v pozadí, může stejná absolutní chyba znamenat nesprávnou detekci klíčového bodu.

Relativní chyba. Tato je získaná vydělením absolutní chyby úhlopříčkou ohraničujícího boxu osoby získaného z anotace. Tato operace řeší zmíněný problém rozdílných interpretací absolutních chyb. Úhlopříčka byla zvolena, neboť není závislá na tom, zda je detekční box vertikální nebo horizontální a má stejnou jednotku jako chyba.

OKS metrika. Metrika představuje obdobu IoU [36] (*Intersection over Union*) mezi anotovanými a detekovanými klíčovými body, viz také Podkapitola 2.1.

OKS je definováno jako:

$$OKS = \frac{\sum_i [\exp(\frac{-{d_i}^2}{2s^2{k_i}^2})\delta(v_i > 0)]}{\sum_i [\delta(v_i > 0)]}, \tag{4.4}$$

kde d_i je Euklidovská vzdálenost mezi jednotlivými klíčovými body anotace a detekce, v_i je viditelnost klíčového bodu v obrázku, s je měřítko ohraničujícího boxu a k_i je konstanta daného klíčového bodu. Perfektní detekci představuje hodnota $OKS = 1$. Měřítko s lze získat ze vztahu

$$\mathrm{s} = \sqrt{\text{šířka} \cdot \text{výška}}, \tag{4.5}$$

kde *šířka* představuje rozměr šířky ohraničujícího boxu anotované osoby a *výška* je rozměr výšky ohraničujícího boxu anotované osoby.

Vztah 4.4 bere v potaz zabíranou plochu osobou a velikost jednotlivých částí lidského těla (každá část těla má vlastní konstantu).

V našem porovnání bereme průměrnou hodnotu OKS metriky vzhledem k validačním obrázkům, které obsahují anotované osoby s klíčovými body a zároveň na nich jednotlivé algoritmy nalezly alespoň jednu osobu.

IoU. Metrika je používaná v objektové detekci a porovnává ohraničující boxy detekce a anotace. Vypočítá se ze vztahu

$$\text{IoU} = \frac{\text{oblast průniku}}{\text{oblast sjednocení}}, \tag{4.6}$$

kde *oblast průniku* je průnik zmíněných ohraničujících boxů a *oblast sjednocení* je jejich sjednocení. V nejlepším případě dojde k vzájemnému překryvu boxů s hodnotou $IoU = 1$. V našem porovnání IoU nevyužíváme, neboť nám jde o preciznost klíčových bodů.

4.3 Úprava výsledků

Pro export výsledků detekcí bylo nutné upravit Detectron2 a MMPose demonstrační skripty. U Detectron2 představuje úprava jeden přidaný cyklus přes počet osob na uložení detekovaných osob do Python listu a následný zápis do souboru. U MMPose bylo přidáno pouze uložení Python listu do souboru.

Před porovnáním bylo třeba sjednotit formát všech výsledků detekcí. Za tímto účelem vznikly tři skripty pro úpravu AlphaPose, MMPose a OpenPose výsledků, které se svým formátem lišily. Skripty jsou dostupné v projektovém repositáři na Gitlabu [37].

4.4 Zpracování dat

Na porovnání výsledků jednotlivých algoritmů vůči anotacím z datasetu byl vytvořen Python skript, který je volně dostupný v projektovém repozitáři na Gitlabu [37]. Pro práci s anotacemi využívá COCO API [38]. Skript počítá a ukládá počet anotovaných osob, nadbytečných detekcí, falešně negativních detekcí, průměrnou Euklidovskou vzdálenost klíčových bodů od anotací v pixelech, průměrnou relativní chybu v procentech, počet chybějících klíčových bodů vůči anotacím a hodnoty OKS metriky v rámci každého z validačních obrázků.

Za *nadbytečné detekce* je považována absolutní hodnota rozdílu počtu detekovaných osob s počtem anotovaných osob v případě, kdy je počet detekovaných osob větší než počet anotovaných osob v obrázku.

V případě, že je v obrázku počet detekovaných osob menší než počet anotovaných osob, je jejich absolutní hodnota rozdílu přičtena k hodnotě *falešně negativních detekcí*.

K hodnotě *falešně negativních detekcí* jsou zároveň přičítány případy, kdy k anotaci neexistuje dostupná detekce s klíčovým bodem uvnitř ohraničujícího boxu anotace.

Z těchto detekcí a počtu anotovaných osob lze zjistit hodnoty pravdivě pozitivních a falešně pozitivních detekcí.

Pravdivě pozitivní detekce lze určit ze vztahu

$$\mathrm{TP} = \mathrm{GT} - \mathrm{FN}, \tag{4.7}$$

kde TP představuje pravdivě pozitivní detekce, GT je počet anotovaných osob a FN je počet falešně negativních detekcí.

Falešně pozitivní detekce lze určit ze vztahu

$$\mathrm{FP} = \text{nadbytečné detekce} + \mathrm{FN}. \tag{4.8}$$

Ke zjištění odchylky detekce je potřeba najít správný pár detekované a anotované osoby. Ve skriptu dochází k vypočtení průměrné absolutní a relativní chyby u všech možných párových kombinací detekcí a anotací v rámci jednoho obrázku.

Algoritmus Párování anotací a detekcí

```
pole = []
while anotované osoby do
  while detekované osoby do
    d ← 0
    if viditelnost KB[1]anotace > 0 then
      if skóre KB detekce > 0 then
        d ← d + Euklidovská vzdálenost KB
      end if
    end if
    d_abs ← d/počet KB
    d_rel ← d_abs/velikost úhlopříčky
    if d_rel > 0.3 or anotovaná osoba bez KB then
      if not KB uvnitř ohraničujícího boxu then
        break
      end if
    end if
    pole append d_abs, d_rel
  end while
end while
pole sort d_abs
```

V případě, že anotace obsahuje pouze ohraničující box nebo při relativní chybě větší než 30 %, dojde ke kontrole, zda se v boxu nachází alespoň jeden klíčový bod detekce. Pokud ani jeden klíčový bod neleží uvnitř ohraničujícího boxu, daný pár je zahozen a pokračuje párování s další kombinací. V opačném případě dojde k uložení páru s jejich metrikami.

Po vyčerpání všech kombinací párů dojde k vzestupnému seřazení uložených párů dle jejich hodnoty absolutní chyby a dochází k výběru dvojic s nejnižšími chybami, které určují finální pár. Při výběru párů nemůže dojít k vybrání již zahrnutého páru, tj. obsahujícího anotaci nebo detekci v již vybrané dvojici.

[1]Klíčový bod.

Při vybírání finálních dvojic jsou upřednostňovány detekce vůči anotovaným osobám s klíčovými body.

Výpočet Euklidovských vzdáleností detekovaných klíčových bodů od anotací funguje pouze v případě, že skóre detekovaného klíčového bodu není 0 a zároveň stav viditelnosti je 1 (není viditelný, ale označený) nebo 2 (viditelný a označený).

Pokud se anotovaná osoba neobjevuje v žádném páru, je označena a zanesena do statistik jako nedetekována.

AlphaPose, Detectron2 a MMPose predikují všechny klíčové body, včetně situací, kdy jsou dané klíčové body zakryty. Viditelnost se odrazí v nízkém, ale stále nenulovém skóre klíčového bodu. Pouze v případě OpenPose došlo během testování k občasnému vynechávání klíčových bodů při nalezených detekcích. Z tohoto důvodu Python skript na porovnání výsledků zaznamenává počet vynechaných klíčových bodů vůči anotacím pouze pro OpenPose algoritmus.

Kapitola 5
Vyhodnocení

V této kapitole jsou uvedena vyhodnocení jednotlivých metrik, naměřené výsledky testovaných algoritmů a jejich vzájemné porovnání.

5.1 Instalace

V této podkapitole jsou popsány problémy, které se vyskytly během instalace jednotlivých algoritmů a zhodnocení jejich náročnosti zprovoznění dle již zmíněné stupnice v Oddíle 4.2.1.

5.1.1 AlphaPose

Instalace postupovala podle Conda návodu v oficiálním repositáři AlphaPose [4]. Během instalace se vyskytl problém kompatibility instalovaných balíčků s Python verzí 3.6, ač dle návodu by měla být podporována. Po instalaci novější verze Pythonu 3.7.13 byl problém vyřešen.

Další problém nastal v momentu, kdy byla vyžadována instalace balíčku Pycocotools. Tento problém je zmíněný na oficiálním "Issues" fóru [39]. Po odstranění řádku přidávajícího balíček mezi vyžadované, proběhla instalace v pořádku.

Z důvodu nekompatibility balíčků s verzí Python uvedené v oficiálním návodu a výskytu zdokumentovaného problému je instalace hodnocena jako těžká.

5.1.2 Detectron2

Detectron2 [26] byl nainstalován ze zdrojového kódu podle návodu v dokumentaci na Python verzi 3.8.12. Instalační návod obsahuje 3 řádky terminálových příkazů.

Během instalace se nevyskytla žádná komplikace a z tohoto důvodu je obtížnost hodnocena jako lehká.

5.1.3 MMPose

Postup instalace MMPose balíčku následovala podrobný návod z oficiálního repositáře na Github [28]. Instalace proběhla s Python verzí 3.7.11.

Hodnocení obtížnosti instalace vzhledem k důkladné dokumentaci postupu bez výskytu problému je hodnocena jako lehká.

5.1.4 OpenPose

OpenPose je již řadu let implementován v laboratoři a tedy jeho obtížnost instalace nebyla zkoumána přímo. Existuje nyní již přenosná demoverze pro Windows, avšak standardní způsob instalace je projekt zkompilovat.

Vzhledem k závislosti na externích knihovnách (jako je Caffe a další) by dle zkušenosti na katedře byla instalace hodnocena jako těžká.

5.2 Postup testování

Jak již bylo zmíněno, testování AlphaPose, Detectron2 a MMPose proběhlo na stejném zařízení a OpenPose byl spuštěn na výkonnějším zařízení.

Aby bylo docíleno srovnatelných podmínek, bylo zařízení zapnuto 5 minut před spuštěním jednotlivých testovaných algoritmů. Tuto dobu považujeme za dostatečnou ke spuštění operačního systému a dodatečných programů. Následně byly algoritmy spuštěny pomocí příkazu v terminálu viz níže.

Při testování AlphaPose a OpenPose bez vykreslování výsledků do obrázků jsme pozorovali shodnou paměťovou i časovou náročnost jako při zapnutém vykreslování. Detectron2 a MMPose se nepodařilo spustit bez vykreslování výsledků, proto jsou všechny algoritmy testovány s vykreslováním výsledků do obrázků.

Během testování bylo zařízení testující AlphaPose, Detectron2 a MMPose umístěno na chladící desce se dvěma větráčky.

```
AlphaPose
mprof run --include-children --multiprocess \
    python demo_inference.py \
    --cfg 256x192_res50_lr1e-3_2x-dcn.yaml \
    --checkpoint fast_dcn_res50_256x192.pth \
    --save_img --detbatch 1 --posebatch 40

Detectron2
mprof run --include-children --multiprocess \
    python demo.py \
    --config-file keypoint_rcnn_R_50_FPN_3x.yaml \
    --opts MODEL.WEIGHTS key_rcnn_R_50_FPN_3x.pkl

MMPose
mprof run --include-children --multiprocess \
    python demo/bottom_up_img_demo.py \
    hrnet_w32_coco_512x512.py \
    hrnet_w32_coco_512x512-bcb8c247_20200816.pth \
    vis_results
```

```
OpenPose
mprof run --include-children --multiprocess \
    ./build/examples/openpose/openpose.bin --model_pose COCO \
    --display 0
```

5.3 Časová a paměťová náročnost

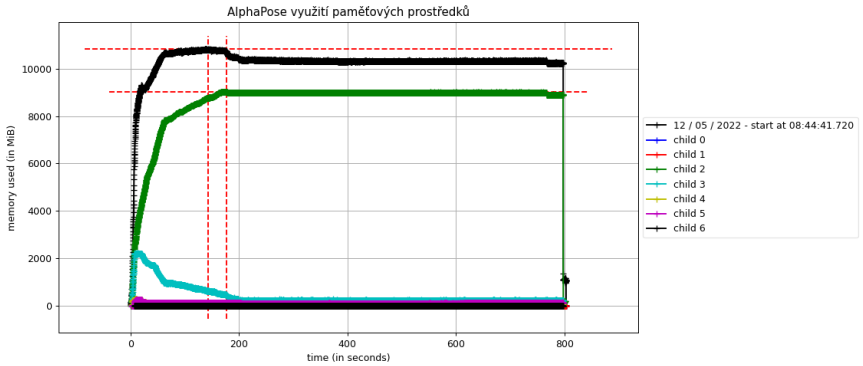
V Tabulkách 5.1 a 5.2 jsou uvedeny maximální hodnoty využití paměti RAM a grafické karty a doby běhů jednotlivých algoritmů. Průběh využití paměti RAM v průběhu testování je zobrazen v Obrázcích 5.1- 5.4

Algoritmus	Max. využití paměti RAM [GB]	Využití paměti grafické karty [GB]
AlphaPose	10.8	3.0
Detectron2	3.7	2.7
MMPose	4.0	2.0
OpenPose	0.9	2.5

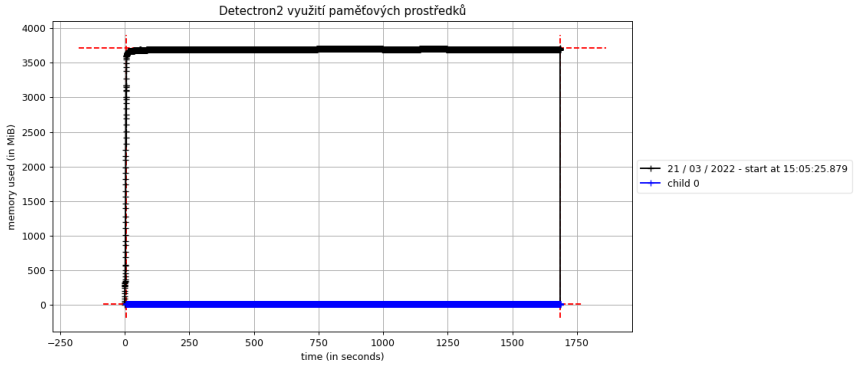
Tabulka 5.1: Využití RAM a grafické paměti při testování.

Algoritmus	Doba běhu	Počet obrázků za sekundu
AlphaPose	13 minut	6.26
Detectron2	26 minut	3.19
MMPose	1 hodina 6 minut	1.25
OpenPose	14 minut	5.71

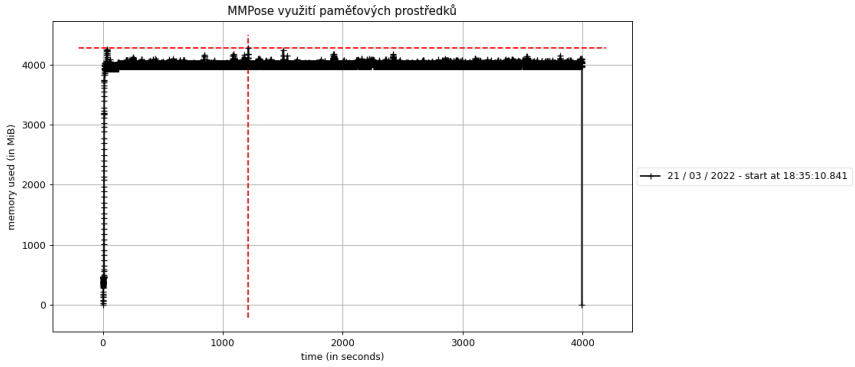
Tabulka 5.2: Doby běhů jednotlivých algoritmů a odhad zpracovaných obrázků za sekundu vzhledem k době běhů a počtu obrázků.



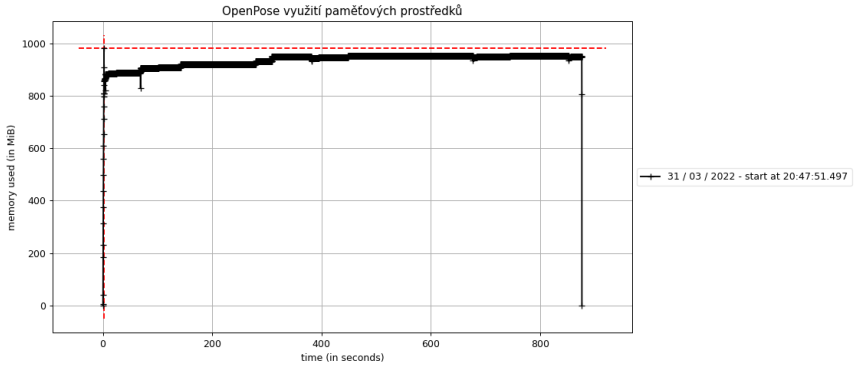
Obrázek 5.1: Graf využití paměti procesy vytvořenými AlphaPose algoritmem.



Obrázek 5.2: Graf využití paměti procesy vytvořenými Detectron2 algoritmem.



Obrázek 5.3: Graf využití paměti procesy vytvořenými MMPose algoritmem.



Obrázek 5.4: Graf využití paměti procesy vytvořenými OpenPose algoritmem.

AlphaPose využíval během testování maximálně 10.8 GB paměti RAM a 3.0 GB paměti grafické karty. Zároveň dokázal využít všechna vlákna procesoru najednou. Testování zabralo 13 minut.

Detectron2 využíval až 3.7 GB paměti RAM a dokázal zatěžovat pouze jedno celé vlákno procesoru. Ke zpracování obrázků využíval 2.7 GB paměti grafické karty. Testování trvalo 26 minut.

MMPose během detekování využíval až 4.0 GB paměti RAM a pouze jedno celé vlákno procesoru. Ke zpracování dat využíval 2.0 GB paměti grafické karty. Testování trvalo 1 hodinu a 6 minut.

OpenPose využíval maximálně 0.9 GB paměti RAM a 2.5 GB paměti grafické karty. Ke zpracování využíval pouze jednoho vlákno procesoru. Testování výsledků trvalo 14 minut.

Z tabulek 5.1 a 5.2 vyplývá, že AlphaPose zpracoval data za nejkratší čas 13 minut s největším využitím paměťových prostředků. Oproti tomu nejméně náročný OpenPose využíval pouze 0.9 GB paměti RAM a 2.5 GB grafické paměti.

5.4 Hodnocení detekcí

5.4.1 Binární porovnání

V Tabulkách 5.3 až 5.5 jsou uvedeny počty obrázků s pravdivě pozitivními detekcemi, počty falešně negativních a pravdivě pozitivních detekcí a výsledné průměrné hodnoty přesností a senzitivit jednotlivých algoritmů vůči všem validačním obrázkům. Hodnota průměrné přesnosti a průměrné senzitivity jednotlivých algoritmů je zároveň vynesena do souřadné soustavy v Obrázku 5.5.

Algoritmus	Počet obrázků s pravdivě pozitivními detekcemi
AlphaPose	2630
Detectron2	2640
MMPose	2371
OpenPose	2289
Počet obrázků s osobami	2693

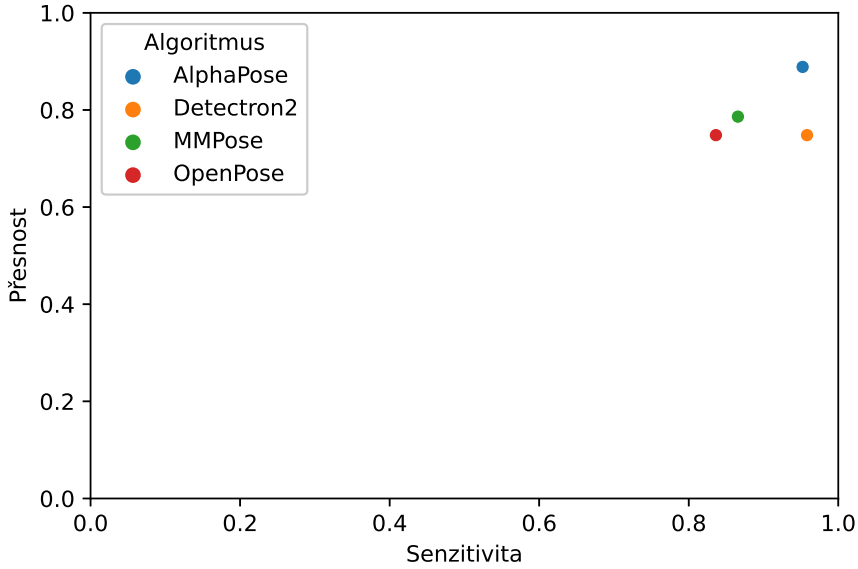
Tabulka 5.3: Počet obrázků s pravdivě pozitivními detekcemi jednotlivých algoritmů.

Algoritmus	Falešně negativní	Pravdivě pozitivní
AlphaPose	1263	9741
Detectron2	1253	9751
MMPose	3990	7014
OpenPose	4463	6541
Celkem osob na obrázcích	11004	

Tabulka 5.4: Počet falešně negativních a pravdivě pozitivních detekcí jednotlivými algoritmy ze všech obrázků.

Algoritmus	Průměrná přesnost	Průměrná senzitivita
AlphaPose	0.89	0.95
Detectron2	0.75	0.96
MMPose	0.79	0.87
OpenPose	0.75	0.84

Tabulka 5.5: Hodnoty průměrné přesnosti a průměrné senzitivity jednotlivých algoritmů ze všech validačních obrázků.



Obrázek 5.5: Graf hodnot průměrné přesnosti a průměrné senzitivity jednotlivých algoritmů ze všech validačních obrázků. Souřadnice [1;1] představují ideální hodnotu průměrné přesnosti a průměrné senzitivity.

Z hodnot v uvedených tabulkách vyplývá, že AlphaPose je nejlepší binární detektor osob. Na Obrázku 5.5 je vidět, že se AlphaPose nejvíce přibližuje souřadnicím [1;1] představujícím ideální hodnoty průměrné přesnosti a průměrné senzitivity. AlphaPose průměrně určoval 0.91 falešně pozitivní detekce a průměrně vykázal 0.25 falešně negativních detekcí na každém validačním obrázku. Celkem detekoval pravdivě pozitivní osoby na 2630 obrázcích z 2693 a detekoval falešně pozitivní osoby na 115 obrázcích bez osob.

Druhým nejúspěšnějším binárním algoritmem podle vzdálenosti dvojice hodnot průměrné přesnosti a senzitivity od souřadnic [1;1] je MMPose. Průměrné v každém obrázku detekoval 1.32 falešně pozitivních detekcí a vykázal 0.80 falešně negativních detekcí na všech obrázcích. MMPose detekoval pravdivě pozitivní osoby na 2371 obrázcích. Zároveň detekoval falešně pozitivní osoby na 191 obrázcích bez osob.

Třetím binárním detektorem podle vzdálenosti dvojice hodnot průměrné přesnosti a senzitivity od souřadnic [1;1] je Detectron2. V průměrné přesnosti

zaostal o 0.04 za MMPose s hodnotou 0.75, ovšem v průměrné senzitivitě předčil MMPose o 0.09 s nejvyšší hodnotou 0.96. Celkem detekoval pravdivě pozitivní osoby na 2640 obrázcích. Průměrně určoval 1.18 falešně pozitivních detekcí a vykázal 0.25 falešně negativních detekcí na každém validačním obrázku. Detectron2 detekoval falešně pozitivní osoby na 604 obrázcích, které neobsahovaly ani jednu osobu.

Nejhůře skončil OpenPose s průměrnou přesností 0.75 a průměrnou senzitivitou 0.84. Detekoval pravdivě pozitivní osoby na 2289 obrázcích. V každém obrázku nalezl 1.13 falešně pozitivních detekcí a vykázal 0.89 falešně negativních detekcí. Detekoval falešně pozitivní osoby na 352 obrázcích bez osob.

Z těchto dat je vidět, že v případě AlphaPose a Detectron2 algoritmů fungujících metodou shora-dolů je hodnota falešně negativních detekcí méně jak třetinová oproti algoritmům s metodou zdola-nahoru.

Porovnáním Detectron2 a MMPose výsledků je vidět, že Detectron2 i přes průměrně nižší hodnotu falešně pozitivních detekcí častěji detekoval osoby na obrázcích, které neobsahovali anotované osoby, oproti MMPose.

5.4.2 Výsledky porovnání klíčových bodů

V Tabulkách 5.6-5.8 jsou uvedeny průměrné hodnoty absolutních chyb s průměrným počtem vynechaných klíčových bodů, relativních chyb a OKS metriky s rozptyly. V Obrázcích 5.6 a 5.7 jsou zobrazeny vzestupně seřazené relativní chyby a sestupně seřazené hodnoty OKS testovaných obrázků.

Algoritmus	Průměrná absolutní chyba [px]	Rozptyl	Průměr VKB[1]
AlphaPose	9.55	14.84	0
Detectron2	11.36	16.80	0
MMPose	10.26	17.51	0
OpenPose	8.65	11.38	2.10

Tabulka 5.6: Hodnoty absolutních chyb s rozptyly ze všech validačních obrázků.

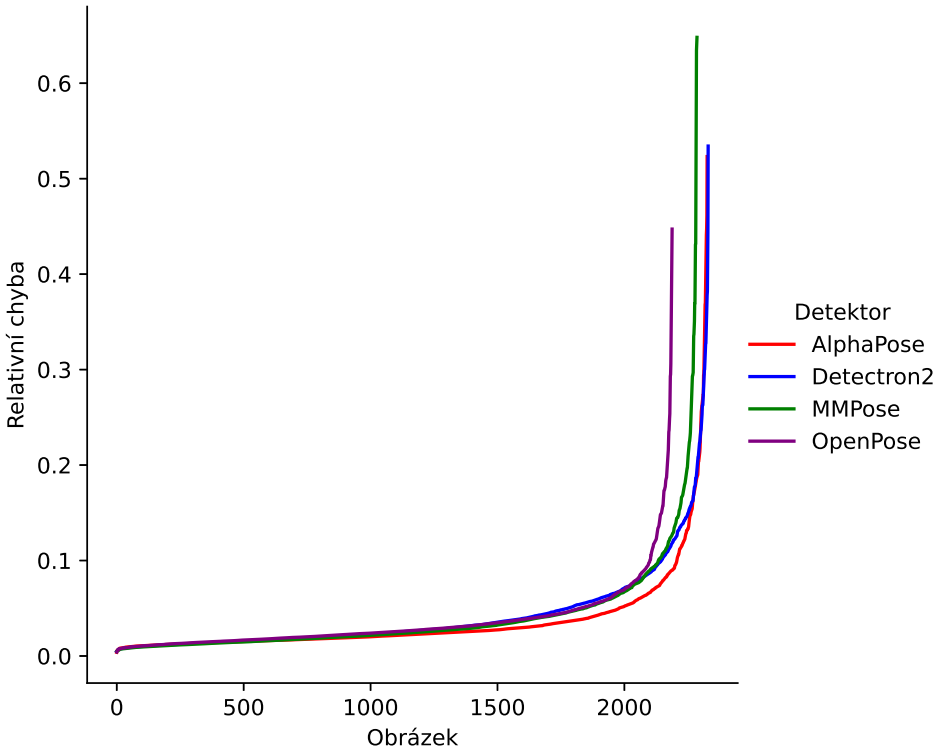
Algoritmus	Průměrná relativní chyba [%]	Rozptyl relativní chyby
AlphaPose	3.52	4.42
Detectron2	4.20	4.73
MMPose	3.93	5.11
OpenPose	3.56	3.60

Tabulka 5.7: Hodnoty průměrných relativních chyb s rozptylem ze všech validačních obrázků.

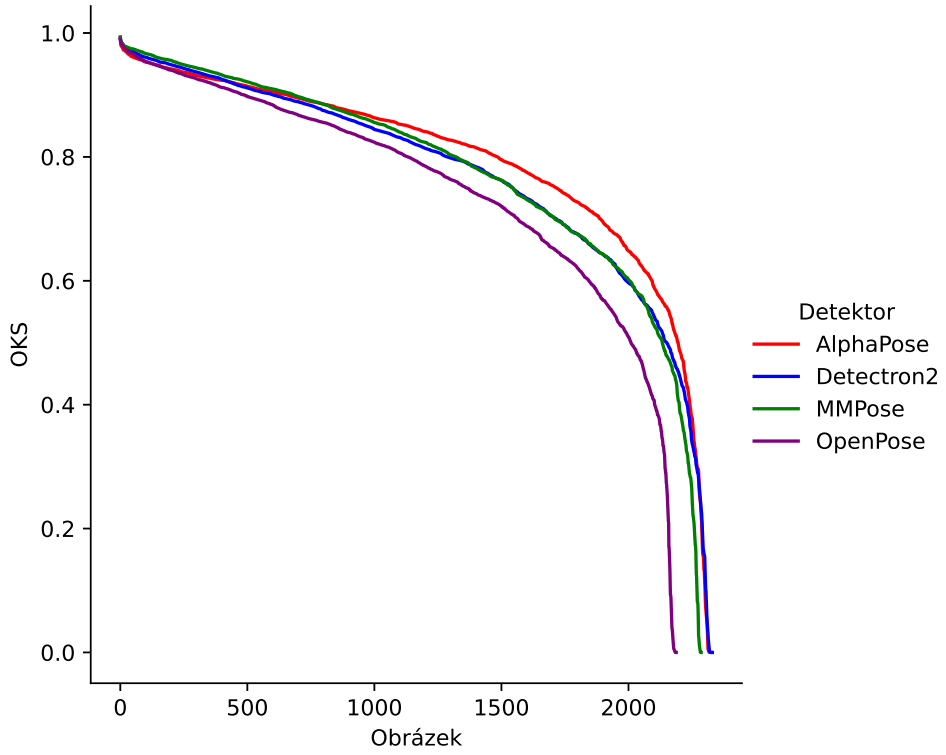
[1] Vynechaných klíčových bodů.

Algoritmus	Průměrná hodnota OKS metriky	Rozptyl OKS
AlphaPose	0.80	0.17
Detectron2	0.77	0.18
MMPose	0.79	0.17
OpenPose	0.76	0.17

Tabulka 5.8: Průměrné hodnoty OKS metriky s rozptylem vzhledem k obrázkům obsahující anotace s klíčovými body, na kterých byla úspěšně detekována alespoň jedna osoba jednotlivými algoritmy.



Obrázek 5.6: Graf vzestupně uspořádaných relativních chyb na obrázcích s detekovanými osobami jednotlivými algoritmy.



Obrázek 5.7: Graf sestupně uspořádaných OKS hodnot získaných na obrázcích obsahující anotace s klíčovými body, na kterých byla úspěšně nalezena alespoň jedna osoba jednotlivými algoritmy.

Z Obrázku 5.6 je vidět, že Detector2 a AlphaPose rozpoznaly osoby s klíčovými body na větším počtu testovacích obrázků než MMPose a OpenPose. Zároveň MMPose zaznamenal nejvyšší relativní chybu přes 60 %.

Z hodnot uvedených v Tabulce 5.7 vyplývá, že AlphaPose dosáhl nejmenší průměrné relativní chyby. Zároveň z Obrázku 5.7 a hodnoty z Tabulky 5.8 je vidět, že disponuje nejvyšší hodnotou OKS ze všech testovaných algoritmů. Z tohoto důvodu je považován za nejvhodnější algoritmus pro určení klíčových bodů lidského těla.

Druhým nejlepším algoritmem na určování klíčových bodů je MMPose, který dosáhl druhé nejlepší hodnoty v metrice OKS a třetí nejmenší průměrnou hodnotu relativní chyby.

Kapitola 6

Diskuze

I přes výsledky porovnání jsme si stále vědomi, že testované algoritmy poskytují mnoho konfigurací s různými modely, které mohou nabídnout větší průměrnou přesnost, vyšší rychlost či menší náročnost na provoz. K porovnání byly vybrány konfigurace, které dle specifikací disponují nejlepšími hodnotami průměrné přesnosti a nejmenšího času potřebného na zpracování jednoho obrázku, nebo které představovaly vhodný kompromis (Detectron2, MMPose). Jak již bylo zmíněno, MMPose konfigurace byla vybrána s metodou zdola-nahoru za účelem prozkoumání přímé alternativy k OpenPose.

Při testování paměťové náročnosti byl v případě OpenPose objeven značný rozdíl vůči ostatním algoritmům. Domníváme se, že OpenPose těchto výsledků docílil díky spuštění zkompilovaného projektu, který nevyžaduje vysoký výpočetní výkon jako ostatní Python skripty.

COCO dataset byl zvolen z důvodu vysoké variability a jeho širokého rozšíření. Rovněž byl formát COCO již využíván v laboratoři s OpenPose. Na Obrázku 6.1 a 6.2 jsou vidět nedostatky datasetu, kdy obrázek obsahuje málo viditelné neoznačené osoby, ale algoritmy dokázaly tyto osoby detekovat (konkrétně v tomto případě AlphaPose). Detekce potom byla označena jako falešně pozitivní a tím byla úspěšnost degradována než by byla v případě dokonalého datasetu.

Testovací dataset obsahuje 3 stupně viditelnosti klíčových bodů. V případě, kdy byl klíčový bod označen hodnotou 0, porovnání bylo vynecháno bez penalizací, pokud se detekce nacházela uvnitř ohraničujícího boxu. S ostatními hodnotami klíčových bodů docházelo k jejich porovnání. AlphaPose, Detectron2 a MMPose v případě detekování člověka označily vždy všechny klíčové body, i když s malým skóre. OpenPose oproti tomu klíčové body vynechával a takto vynechané body nebyly penalizovány ve výsledném porovnání, ale pouze zaznamenán jejich počet.

Poslední faktor o kterém víme, že může ovlivňovat výsledky je způsob párování detekcí a anotací. To upřednostňuje páry s anotací obsahující alespoň jeden klíčový bod. V případě, že osoba byla anotována pouze ohraničujícím boxem bez klíčových bodů (klíčové body s hodnotou stavu viditelnosti 0), mohla být detekce přiřazena nesprávně k jiné anotaci s klíčovými body.

Obrázek 6.1: Obrázek 000000167486.jpg z validačního balíčku COCO datasetu detekovaný AlphaPose algoritmem. Obrázek převzat z [8].

Obrázek 6.2: Obrázek 000000167486.jpg z validačního balíčku COCO datasetu s vyznačenými anotacemi. Obrázek převzat z [8].

Kapitola 7

Závěr

Cílem této práce bylo porovnat vybrané algoritmy detekující klíčové body lidí na zvoleném, respektive vytvořeném, datasetu.

Algoritmy byly úspěšně zprovozněny a otestovány na validační sbírce COCO datasetu s kostrou v COCO formátu složenou ze 17 klíčových bodů. K vytvoření vlastního datasetu nedošlo z důvodu neznámého prostředí, ve kterém by měl robot fungovat. I přesto porovnání na obecném datasetu, vytvořeném pro detekci klíčových bodů osob, zahrnulo širokou škálu různorodých scén a výsledky tak pokrývají variabilní prostředí, ve kterých by robot mohl být nasazen.

Porovnáním fungování jednotlivých algoritmů jsme zjistili, že AlphaPose zpracoval data za 13 minut a zároveň využíval většinu dostupného výpočetního výkonu. OpenPose, i přes testování na výkonnějším zařízení, dosáhl výsledků za 14 minut, ovšem s minimálním zatížením pamětí. Detectron2 vyžadoval dvojnásobek času na zpracování dat oproti AlphaPose. MMPose běžel 1 hodinu a 6 minut, což představuje nepoužitelný algoritmus pro naše cílové využití.

Algoritmy byly v rámci spolehlivosti porovnány binárně s ohledem na ohraničující boxy osob, OKS metrikou a vlastní relativní metrikou vůči úhlopříčce ohraničujícího boxu.

Nejlepší variantou pro binární detekci je AlphaPose, který dosáhl nejvyšší dvojice hodnot průměrné přesnosti 0.89 a průměrné senzitivity 0.95. MMPose obsadil druhé místo s dvojicí hodnot průměrné přesnosti 0.79 a průměrné senzitivity 0.87. Oproti tomu OpenPose dosáhl nejnižší dvojice hodnot průměrné přesnosti 0.75 a průměrné senzitivity 0.84. Představuje tak nejhorší z testovaných algoritmů v rámci binární detekce.

V porovnání klíčových bodů z detekovaných obrázků považujeme AlphaPose za nejlepší algoritmus s průměrnou relativní chybou 3.52 % a hodnotou OKS 0.80. Druhým skončil MMPose, který dosáhl 3.93 % průměrné relativní chyby a hodnoty 0.79 v OKS metrice.

Po zvážení všech testovaných metrik považujeme AlphaPose jako nejlepší detektor klíčových bodů člověka. Představuje nejspolehlivější algoritmus v binární detekci a určování klíčových bodů z testovaných algoritmů.

Vytvořený Python skript postupně porovnává výsledky jednoho algoritmu vůči anotacím z obrázku. Jeho malou úpravou může probíhat porovnávání

mezi jednotlivými algoritmy. Porovnávání výsledků jednotlivých algoritmů potom může sloužit jednak k anotaci nových obrázků nebo k syntéze algoritmů v podobě tzv. mixture of experts modelu. I když skript poskytuje základ pro takové rozvedení, plné rozpracování však již bylo nad rámec možností této práce a je ponecháno jako navazující projekt.

Literatura

[1] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D Human Pose Estimation: New Benchmark and State of the Art Analysis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[2] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[3] S. Jin, L. Xu, J. Xu, C. Wang, W. Liu, C. Qian, W. Ouyang, and P. Luo, "Whole-Body Human Pose Estimation in the Wild," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[4] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "AlphaPose," https://github.com/MVIG-SJTU/AlphaPose, 2017, navštíveno: 2022-04-12.

[5] Y.-L. Li, L. Xu, X. Liu, X. Huang, Y. Xu, S. Wang, H.-S. Fang, Z. Ma, M. Chen, and C. Lu, "PaStaNet: Toward Human Activity Knowledge Engine," in *CVPR*.

[6] J. Brooks, "COCO Annotator," https://github.com/jsbroks/coco-annotator/, 2019, navštíveno: 2022-04-21.

[7] B. Sekachev, N. Manovich, M. Zhiltsov, A. Zhavoronkov, D. Kalinin, B. Hoff, TOsmanov, D. Kruchinin, A. Zankevich, DmitriySidnev, M. Markelov, Johannes222, M. Chenuet, a andre, telenachos, A. Melnikov, J. Kim, L. Ilouz, N. Glazov, Priya4607, R. Tehrani, S. Jeong, V. Skubriev, S. Yonekura, vugia truong, zliang7, lizhming, and T. Truong, "opencv/cvat: v1.1.0," Aug. 2020, navštíveno: 2022-04-21. [Online]. Available: https://doi.org/10.5281/zenodo.4009388

[8] T.-Y. Lin, M. Maire, G. Patterson, M. Ronchi R., Y. Cui, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "COCO Dataset," https://cocodataset.org/, navštíveno: 2022-04-21.

[9] R. Li, X. Dong, Z. Cai, D. Yang, H. Huang, S.-H. Zhang, P. Rosin, and S.-M. Hu, "Pose2Seg: Human Instance Segmentation Without Detection," 03 2018.

[10] M. Li, Z. Zhou, J. Li, and X. Liu, "Bottom-up pose estimation of multiple person with bounding box constraint," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 115–120.

[11] Clear Image AI, "Pose Estimator 2," https://aws.amazon.com/marketplace/pp/prodview-eunnr3xwggw42, navštíveno: 2022-05-15.

[12] "Kinect for Windows," https://developer.microsoft.com/en-us/windows/kinect/, navštíveno: 2022-05-15.

[13] B. Raj, "An Overview of Human Pose Estimation with Deep Learning," https://medium.com/beyondminds/an-overview-of-human-pose-estimation-with-deep-learning-d49eb656739b, 2019, navštíveno: 2022-04-12.

[14] G. Ning, J. Pei, and H. Huang, "Lighttrack: A generic framework for online top-down human pose tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 1034–1035.

[15] S. Leijnen and F. Veen, "The Neural Network Zoo," *Proceedings*, vol. 47, p. 9, 05 2020.

[16] I. C. Education, "Convolutional Neural Networks," 08 2020, navštíveno: 2022-05-12.

[17] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 764–773.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[19] M. Wang, T. Sun, K. Song, S. Li, J. Jiang, and L. Sun, "An efficient sparse pruning method for human pose estimation," *Connection Science*, vol. 34, no. 1, pp. 960–974, 2022.

[20] Jeblad, "Canonical form of residual neural nets," https://commons.wikimedia.org/w/index.php?curid=64298839, 2017, navštíveno: 2022-05-16.

[21] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[22] W. Abdulla, "Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow," https://github.com/matterport/Mask_RCNN, 2017, navštíveno: 2022-05-11.

[23] C. Patil and V. Gupta, "Human Pose Estimation using Keypoint RCNN in PyTorch," https://learnopencv.com/human-pose-estimation-using-keypoint-rcnn-in-pytorch/#overview, 2021, navštíveno: 2022-05-11.

[24] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," *Advances in neural information processing systems*, vol. 28, 2015.

[25] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "Rmpe: Regional multi-person pose estimation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2334–2343.

[26] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019, navštíveno: 2022-04-14.

[27] R. A. Güler, N. Neverova, and I. Kokkinos, "DensePose: Dense Human Pose Estimation in the Wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[28] M. Contributors, "OpenMMLab Pose Estimation Toolbox and Benchmark," https://github.com/open-mmlab/mmpose, 2020, navštíveno: 2022-04-18.

[29] A. Newell, Z. Huang, and J. Deng, "Associative embedding: End-to-end learning for joint detection and grouping," *Advances in neural information processing systems*, vol. 30, 2017.

[30] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep High-Resolution Representation Learning for Human Pose Estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[31] M. Contributors, "MMTracking: OpenMMLab video perception toolbox and benchmark," https://github.com/open-mmlab/mmtracking, 2020, navštíveno: 2022-05-11.

[32] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[33] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "AlphaPose," https://github.com/MVIG-SJTU/AlphaPose/blob/master/docs/MODEL_ZOO.md, 2017, navštíveno: 2022-05-15.

[34] "Anaconda Software Distribution," 2020. [Online]. Available: https://docs.anaconda.com/

[35] F. Pedregosa and P. Gervais, "Memory Profiler," https://github.com/pythonprofilers/memory_profiler, navštíveno: 2022-02-14.

[36] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized Intersection over Union," June 2019.

[37] M. Mísař, "GitLab repozitář," https://gitlab.fel.cvut.cz/body-schema/pose-estimation-comparison, navštíveno: 2022-05-15.

[38] "COCO API," https://github.com/cocodataset/cocoapi, 2014, navštíveno: 2022-04-28.

[39] "installation failed," https://github.com/MVIG-SJTU/AlphaPose/issues/572, 2020, navštíveno: 2022-04-28.