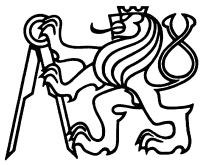
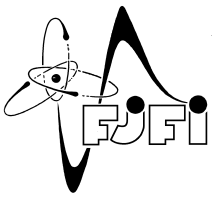


České vysoké učení technické v Praze
Fakulta jaderná a fyzikálně inženýrská



Vyhodnocení efektivity reklamy za použití detektoru lidí a odhadování směru pohledu

Evaluation of advertisement effectiveness using people detection and gaze estimation

Diplomová práce

Autor: **Bc. Thi Thu Hien Nguyenová**

Vedoucí práce: **Doc. Ing. Filip Šroubek, Ph.D. DSc.**

Vedoucí specialista: **Ing. Filip Naiser**

Konzultant: **Ing. Tomáš Kerepecký**

Akademický rok: 2022/2023

České vysoké učení technické v Praze, Fakulta jaderná a fyzikálně inženýrská

Katedra: matematiky

Akademický rok: 2022/2023

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student:	Bc. Thi Thu Hien Nguyenová
Studijní program:	Aplikované matematicko-stochastické metody
Název práce (česky):	Vyhodnocení efektivity reklamy za použití detektoru lidí a odhadování směru pohledu
Název práce (anglicky):	Evaluation of advertisement effectiveness using people detection and gaze estimation
Jazyk práce:	čeština

Pokyny pro vypracování:

1. Seznamte se s problematikou kalibrace a registrace kamery pro potřeby odhadu vzdálenosti.
2. Přizpůsobte a dotrénujte state of the art detektor, který bude detekovat chodce a predikovat pohlaví (jako bonus predikovat věk).
3. Seznamte se s metodami pro odhad směru pohledu a vhodnou metodu použijte nebo doplňte detektor o tento odhad.
4. Seznamte se s metodami pro sledování chodců a použijte vhodnou metodu nad výstupy detektoru.
5. Výstupy detektoru a trackingu použijte pro měření sledovanosti reklamy.
6. Seznamte se s metodami pro kvantizaci a zrychlení běhu neuronových sítí.
7. Natrénovaný detektor zoptimalizujte a připravte pro nasazení.
8. Vyhodnoťte přesnost měření statistik: 1. počet lidí, kteří zhlédli reklamu, 2. počet potenciálních diváků, 3. informace o délce sledování reklamy jednotlivcem.

Doporučená literatura:

1. I. Goodfellow, Y. Bengio, A. Courville, Deep learning. MIT press, 2016.
2. L. Liu, W. Ouyang, X. Wang, et al., Deep learning for generic object detection: A survey. International journal of computer vision 128, 2020, 261–318.
3. C. Y. Wang, A. Bochkovskiy, H. M. Liao, Scaled-yolov4: Scaling cross stage partial network. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, 13029-13038.

Jméno a pracoviště vedoucí diplomové práce:

Doc. Ing. Filip Šroubek, Ph.D. DSc.;
Ústav teorie informace a automatizace
Pod Vodárenskou věží 4
182 00, Praha 8

Jméno a pracoviště vedoucího specialisty:

Ing. Filip Naiser
iC Systems.ai, s.r.o.
Klimentská 1216/46
110 00, Praha 1

Jméno a pracoviště konzultanta:

Ing. Tomáš Kerepecký
Ústav teorie informace a automatizace
Pod Vodárenskou věží 4
182 00, Praha 8

Datum zadání diplomové práce: 31.10.2022

Datum odevzdání diplomové práce: 3.5.2023

Doba platnosti zadání je dva roky od data zadání.

V Praze dne 31.10.2022

garant oboru

vedoucí katedry

děkan

Poděkování:
Chtěla bych zde poděkovat především svému vedoucímu specialistovi Ing. Filipovi Naiserovi, který mi pomáhal v inženýrské části diplomové práce, školiteli Doc. Ing. Filipovi Šroubkovi, Ph.D. DSc. a konzultantovi Ing. Tomaši Kerepeckému za pečlivost, ochotu, vstřícnost a odborné i lidské zázemí při vedení mé práce. V neposlední řadě poděkování patří poděkování mé rodině a přátelům.

Čestné prohlášení:
Prohlašuji, že jsem tuto práci vypracovala samostatně a uvedla jsem všechnu použitou literaturu.

V Praze dne 1. května 2023 Bc. Thi Thu Hien Nguyenová

Název práce:

Vyhodnocení efektivity reklamy za použití detektoru lidí a odhadování směru pohledu

Autor: Bc. Thi Thu Hien Nguyenová

Program: Aplikované matematicko-stochastické metody

Druh práce: Diplomová práce

Vedoucí práce: Doc. Ing. Filip Šroubek, Ph.D. DSc., Ústav teorie informace a automatizace AV ČR, v.v.i., oddělení Zpracování obrazové informace, Pod Vodárenskou věží 4, 182 08 Praha 8

Vedoucí specialista: Ing. Filip Naiser, iC Systems.ai, s.r.o., Klimentská 1216/46, 110 00 Praha 1

Konzultant: Ing. Tomáš Kerepecký, Ústav teorie informace a automatizace AV ČR, v.v.i., oddělení Zpracování obrazové informace, Pod Vodárenskou věží 4, 182 08 Praha 8

Abstrakt: Cílem této práce bylo vytvořit algoritmus, který je schopen odhadovat, zda se osoba vyskytující se na vstupním obrázku dívá na reklamní baner (citylight). To zahrnovalo seznámit se s problematikou kalibrace kamery. Prostudovat si problematiku detekce a sledování lidského těla, odhad pozice hlavy a predikce věku a pohlaví pomocí metod hlubokého učení. Zvolit state-of-the-art detektor chodců, dotrénovat ho na predikci pohlaví a použít ho v prostředí zamyšlené aplikace. Nakonec jsme měli tento algoritmus zoptimalizovat a vyhodnotit přesnosti statistik jako je například počet diváků.

Klíčová slova: detekce chodců, detektory, hluboké učení, odhad pozice hlavy, odhad věku a pohlaví, strojové učení, reklama, kalibrace kamery

Title:

Evaluation of advertisement effectiveness using people detection and gaze estimation

Author: Bc. Thi Thu Hien Nguyenová

Abstract: The goal of this work was to create an algorithm that is able to estimate whether the person appearing in the input image is looking at an advertising banner (citylight). This included learning about camera calibration. Study the problem of human body detection and tracking, head position estimation and age and gender prediction using deep learning methods. Choose a state-of-the-art pedestrian detector, finetuned it to predict gender and use it in the environment of a thoughtful application. Finally, we had to optimize this algorithm and evaluate the accuracy of statistics such as the number of viewers.

Key words: pedestrian detection, detectors, deep learning, head pose estimation, age and gender estimation, machine learning, advertising, camera calibration

Obsah

Úvod **11**

1 Hluboké učení **13**
1.1 Umělé neuronové sítě . . . 13
1.2 Hluboké učení . . . 15
1.3 Konvoluční neuronové sítě . . . 17
1.4 Metody pro zrychlení neuronových sítí . . . 19

2 Detekce a sledování chodců **25**
2.1 Detekce chodců . . . 25
2.1.1 Metriky pro detekci objektů . . . 28
2.1.2 YOLO v7 . . . 29
2.2 Sledování chodců . . . 30
2.2.1 Algoritmy pro sledování chodců . . . 31
2.3 Pohlaví a věk . . . 32
2.3.1 Datasety . . . 32

3 Odhad pozice hlavy (pohledu) **33**
3.1 Metody pro odhady pozice hlavy . . . 34
3.2 img2pose . . . 36

4 Kalibrace kamery **37**
4.1 Fotogrammetrická kalibrace . . . 37
4.2 Auto-kalibrace . . . 37
4.3 Kombinace fotogrammetrické a auto-kalibrace . . . 38
4.3.1 Základní rovnosti a definice . . . 39
4.3.2 Vztah mezi rovinným modelem a jeho obrázkem . . . 40
4.3.3 Řešení kalibrace kamery . . . 41
4.3.4 Radiální distorze . . . 43

5 Praktická část **45**
5.1 Příprava dat . . . 46
5.2 Kalibrace kamery . . . 47
5.3 Detekce a sledování člověka . . . 48
5.3.1 Kvantizace . . . 49
5.3.2 Predikce pohlaví . . . 49
5.4 Mapování z 2D obrázku do 3D světa . . . 50
5.4.1 Pomocí projekční matice . . . 51

5.4.2 Pomocí perspektivní transformace 52
5.5 Odhad pozice hlavy 53
5.6 Funkce pro predikce dívání se/nedívání se 54
5.7 Vizualizace 55
5.8 Experiment a výsledky 56
5.9 Diskuze 58

Závěr **61**

Literatura **63**

Úvod

Perzonalizovaná reklama je výkonný nástroj, který zlepšuje relevanci reklam pro uživatele a zvyšuje inzerentům návratnost investic. Na internetu je jednodušší udělat personifikovanou reklamu než v reálném světě, tedy reklamy, které vidíme v obchodních centrech apod. Na internetu odhadují zájmy podle toho, jaké uživatel navštěvuje weby nebo jaké používá aplikace. To umožňuje inzerentům cílit kampaně na základě takových zájmů a přispívá to ke zvýšení spokojenosti uživatelů i inzerentů [19].

Bohužel v reálném světě je situace komplikovanější. Tato skutečnost nás motivovala k napsání této práce, jejíž cílem bylo získat na základě metod počítačového vidění relevantní informace, podle kterých by bylo možné rozhodnout, které reklamy v reálném světě mohou být považovány za relevantní. Pro takový úkol je mimo jiné příhodné získat informace o pozici, věku a pohlaví pozorovatele v daném čase. K detekci chodců, určení pohlaví, natočení hlavy využijeme metody strojového učení. Jedno z nejpokročilejších odvětví strojového učení se zabývá umělými neuronovými sítěmi. Jedná se o učící se výpočetní modely inspirované fungováním lidského mozku.

První čtyři kapitoly jsou zaměřené na teoretický úvod k problematice a poslední kapitola popisuje vlastní implementaci a analýzu provedeného řešení. V první kapitole si můžeme přečíst o základních pojmech hlubokého učení a kvantizaci neuronových sítí. V druhé části se setkáváme s problémem detekce a sledování lidského těla, predikce věku, pohlaví a potřebnými datasety pro naše aplikace. Ve třetí části se zabýváme metodami pro odhad pozice hlavy. Ve čtvrté kapitole se zabýváme hlavně kalibrací kamery a odhady příslušných parametrů. Poslední kapitola obsahuje praktickou část, která začíná kalibrací naší kamery, která bude důležitá pro další kroky. Pokusíme se o dotrénování YOLOv7, aby detekovala muže a ženy z celé postavy. Poté převádíme polohu z obrázku do reálného světa pomocí souřadnic nohou, predikujeme odhad pozice hlavy pomocí veřejného algoritmu img2pose. Vytvoříme funkci pro klasifikaci dívá/nedívá se, spustíme sledovač chodců. Pokusíme se o kvantizaci neuronové sítě YOLOv7. Výstupem tedy bude textový soubor s názvem obrázku, souřadnice ohraničujícího obrázku, zda se člověk dívá, nebo nedívá a nakonec i pohlaví. Vytvoříme vizualizaci tohoto výstupu. V závěru poslední kapitoly je zhodnocení provedeného řešení.

Kapitola 1

Hluboké učení

Umělá inteligence (angl. *artificial intelligence, AI*) je schopnost systému správně interpretovat externí data, učit se z takových dat a používat učení k dosažení konkrétních cílů a úkolů prostřednictvím flexibilní adaptace [43]. Důležitými podobory AI jsou strojové učení a hluboké učení.

Strojové učení (angl. *machine learning, ML*) je v poslední době velmi populární. Jedná se o algoritmy, které se snaží předpovídat výsledky pomocí učení z dat. K tomu tyto algoritmy potřebují vstupní data, očekávaný výstup ke vstupním datům a nástroj k měření přesnosti - počítá chybu mezi výstupem algoritmu a jeho očekávaným výstupem. Pomocí této chybovosti se algoritmus aktualizuje a přizpůsobení se problému nazýváme učením. Algoritmy strojového učení můžeme rozdělit do následujících skupin [13]:

1. Učení s učitelem (angl. *supervised learning*): V tomto případě tréninková data mají vstup i očekávaný výstup. Využíváme to například pro detekci tváří, těla nebo určitých předmětů.

2. Učení bez učitele (angl. *unsupervised learning*): Tréninková data ke vstupům nemají výstupy. Používáme například pro shlukovou analýzu (angl. *clustering*).

3. Kombinace učení s učitelem a bez učitele (angl. *semi-supervised learning*): Část vstupů má výstupy, ale většinou větší část výstupy nezná. Příkladem může být klasifikace, kdy se označená data používají k trénování modelu a neoznačená data se používají ke zlepšení výkonu modelu.

4. Zpětnovazebné učení (angl. *reinforcement learning*): Toto učení má více procesů a řídí se podle pravidel. Během plnění úkolu dostává zpětnou vazbu a podle těchto vazeb se rozhoduje, co bude dělat dále. Uplatňujeme to například v autonomních automobilech, kde se auto učí rozhodovat na základě zpětné vazby, kterou dostává ze svých senzorů a z okolí.

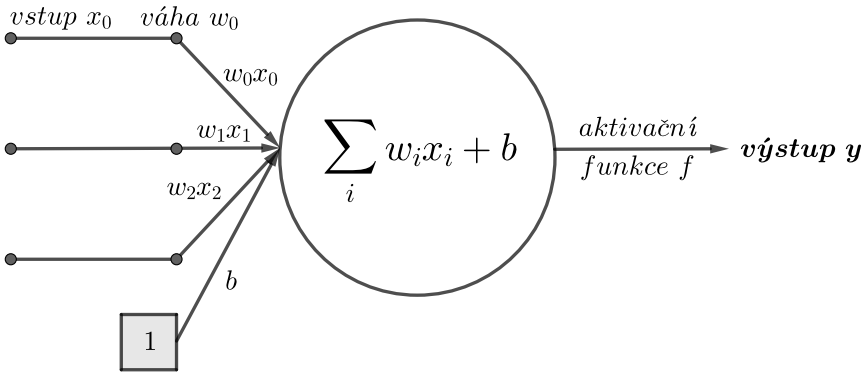
Strojové učení se využívá například ve zdravotnictví, ekonomii nebo v počítačovém vidění.

1.1 Umělé neuronové sítě

Umělé neuronové sítě (angl. *artificial neural network, ANN*, dále už jen neuronové sítě) jsou oblíbeným nástrojem ve strojovém učení. Tyto sítě jsou inspirovány biologickými neuronovými sítěmi. Jsou tvořeny z jednotlivých umělých neuronů, které jsou vzájemně propojeny a předávají si signály. Každé propojení je ovlivněno váhou (angl. *weights, w*), která nám říká, jak je spojení silné, či slabé (důležité/nedůležité). Práh (angl. *bias, b*) označuje hodnotu aktivace neuronu.

Každý vstup do neuronu je ovlivněn váhou a předchozím vstupem, které ovlivňují funkci (1.1) spočítanou v daném neuronu, což je vyobrazeno na obrázku 1.1. Celá umělá neuronová síť šíří vypočítané hodnoty ze vstupních neuronů do výstupních neuronů a využívá váhy a prahy jako parametry [12].

$$y = f(\sum_i w_i x_i + b) \tag{1.1}$$



Obrázek 1.1: Princip neuronu spočívá v přijímání vstupních dat, jejich transformaci pomocí vah a aktivační funkce a vytvoření výstupu.

K učení dochází změnou vah a prahů spojující neurony. Stejně jako biologické neuronové sítě, potřebují umělé neuronové sítě trénovací data, která obsahují vstup a výstup, pomocí kterých je síť schopna se učit. Vstupem může být například fotka zobrazující jednotlivé věci, jako je auto, kolo, člověk, a výstupem mohou být popisky toho, co je na obrázku. Učením se v neuronové síti mění parametry. Cílem změny vah a prahů je upravit vypočítanou funkci tak, aby byly předpovědi v budoucích iteracích přesnější.

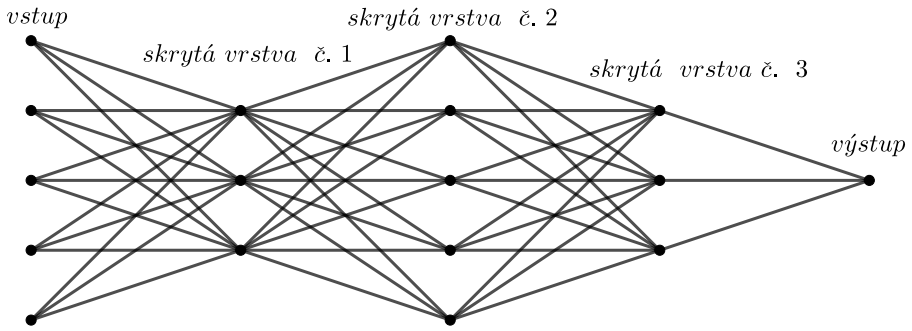
Pokud je tedy neuronová síť trénována s využitím mnoha různých obrázků aut, s velkou pravděpodobností bude nakonec schopna správně rozpoznat auta na obrázku, i ta, která předtím neviděla. Schopnost získat výstupy, které nikdy neviděla, se nazývá zobecnění a to je ideální vlastností modelů strojového učení.

Neuronové sítě můžeme rozdělit do dvou skupin podle počtu vrstev:

1. Jednovrstvé: Skládá se pouze ze vstupní a výstupní vrstvy, což můžeme vidět na obrázku 1.1. Tato jednoduchá neuronová síť se také nazývá perceptron.

2. Vícevrstvé: Ve vícevrstvých neuronových sítích jsou neurony uspořádány do vrstev, ve kterých jsou vstupní a výstupní vrstvy odděleny skupinou skrytých vrstev. Tato vrstvená architektura neuronové sítě se také nazývá dopředná síť a je vyobrazena na obrázku 1.2.

Neuronové sítě taky mohou využívat aktivační funkci. Aktivační funkce vnáší do sítě nelinearitu. To nám umožňuje modelovat výstupní proměnné, které se nelineárně mění s vysvětlujícími proměnnými. Různé aktivační funkce lze použít pro různé modely ve strojovém učení. Nejznámější aktivační funkce jsou [1]: sigmoid (1.2), ReLU (1.3) nebo tanh.



Obrázek 1.2: Vícevrstvá neuronová síť přijímá vstupní data, která projdou několika vrstvami skrytých neuronů s různými váhami a aktivačními funkcemi, aby byly vytvořeny abstraktní reprezentace vstupních dat. Výstupní vrstva pak kombinuje tyto reprezentace a generuje konečný výstup.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{1.2}$$

$$g(x) = \begin{cases} 0 & \text{pro } x < 0 \\ x & \text{pro } x \geq 0 \end{cases} \tag{1.3}$$

1.2 Hluboké učení

Hluboké učení (angl. *deep learning, DL*) je jedním z oborů strojového učení. Jedná se o učení, které klade důraz na vícevrstvé neuronové sítě. Počet skrytých vrstev v neuronových sítích nazýváme hloubkou modelu. Moderní hluboké učení často zahrnuje desítky nebo dokonce stovky po sobě jdoucích skrytých vrstev. Neuronové sítě, které mají jednu a nebo dvě vrstvy nazýváme mělké sítě (angl. *shallow network*). U hlubokého učení může nastat problém, že neuronová síť obsahuje hodně parametrů, třeba i miliony. A najít správnou hodnotu pro všechny parametry může být poměrně těžký úkol. Hlavně, když úprava jednoho parametru ovlivní všechny ostatní.

Chceme-li ovládat výstup neuronové sítě, musíme být schopni změřit, jak rozdílný je výstup modelu od toho, co jsme očekávali. To je úkolem ztrátové funkce. Mezi nejznámější ztrátové funkce patří například střední kvadratická chyba (angl. *mean squared error, MSE*), která je definována:

$$MSE = \frac{\sum_{i=0}^{n} (y_i - y_i^{'})^2}{n}, \tag{1.4}$$

střední absolutní chyba (angl. *mean absolute error, MAE*) definována:

$$MAE = \frac{\sum_{i=0}^{n} |y_i - y_i^{'}|}{n}, \tag{1.5}$$

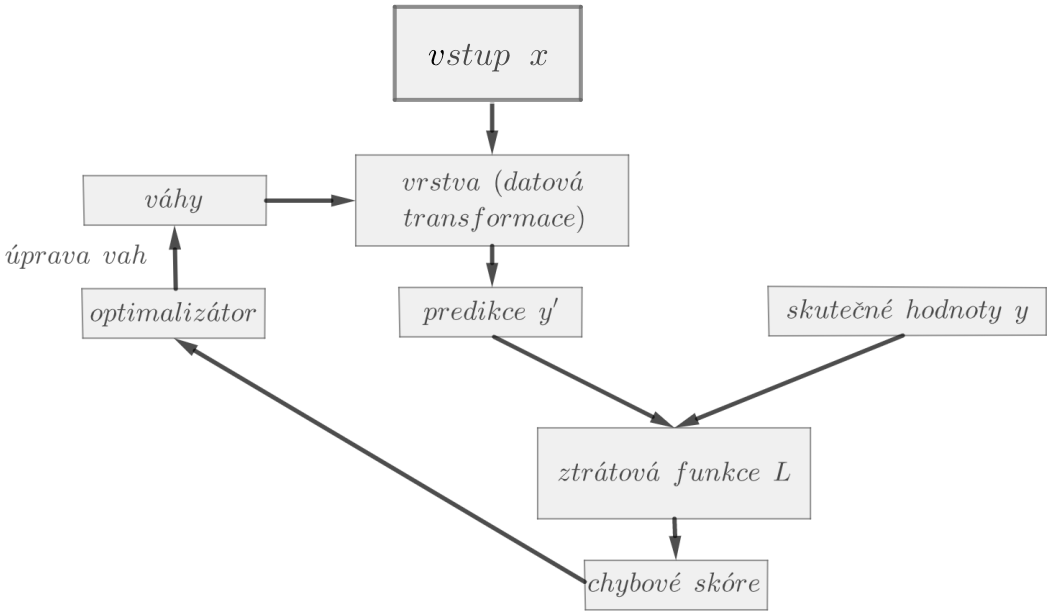
křížová entropie (angl. *cross entropy*) definována:

$$-y_i log(y_i^{'}) - (1 - y) log(1 - y_i^{'}) \text{ pro } i \in \hat{n}, \tag{1.6}$$

kde y_i je pravá hodnota výstupu a $y_i^{'}$ je odhad modelu, n počet dat.

Ztrátová funkce porovnává předpovědi neuronové sítě a skutečného výstupu a vypočítává skóre zachycující, jak dobře si síť vedla. Důležitým krokem v hlubokém učení je využití tohoto skóre jako zpětné vazby k úpravě hodnot parametrů sítě, který sníží skóre ztráty. Což je úkolem optimalizátoru. Optimalizátor je funkce nebo algoritmus, který upravuje váhy a prahy. Tím pomáhá tedy snížit celkovou ztrátu a zlepšit přesnost. Optimalizátor využívá algoritmus zpětného šíření chyb (angl. *backpropagation*) viz. obrázek 1.3, kde ztrátová funkce (MSE, MAE a další) je matematická funkce, která vyjadřuje, jak dobře model předpovídá výstup na základě vstupních dat. Cílem je minimalizovat hodnotu ztrátové funkce, aby byly predikce co nejblíže k požadovanému výstupu. Chybové skóre je míra, která vyjadřuje úspěšnost algoritmu na základě srovnání jeho výstupu s požadovanými výstupy. Tyto metriky mohou být různé v závislosti na typu problému a jsou zpravidla výstupem evaluace algoritmu na testovacích datech. Příkladem chybových metrik jsou přesnost (accuracy), přesnost v klasifikaci (precision), úplnost (recall) atd. A mezi nejznámější optimalizátory patří Adam, SGD nebo Adagrad [23].

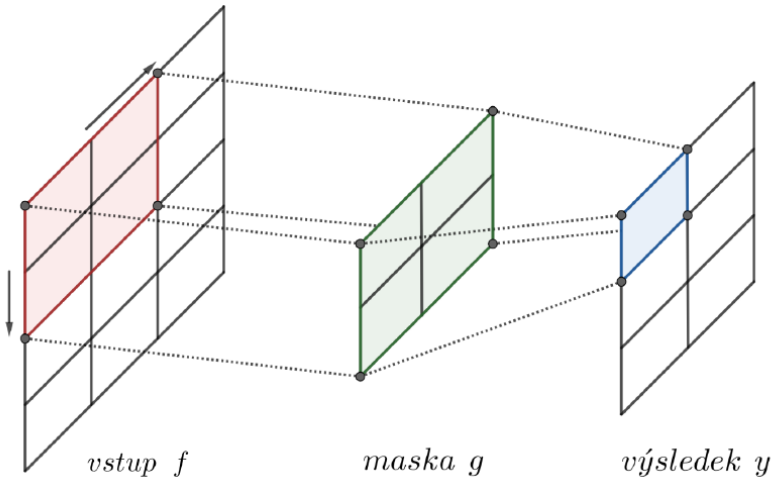
Na začátku mají váhy náhodné hodnoty, takže přesnost sítě je velmi malá, a proto ztrátové skóre je velmi vysoké. Ale s každým krokem síť v ideálním případě upravuje váhy správným směrem a skóre ztráty se sníží - tento celý proces nazýváme tréninkovou smyčkou (angl. *training loop*). Pokud se tento proces opakuje, lze minimalizovat ztrátové skóre. Síť s minimální ztrátou je taková, jejíž výstupy jsou co nejblíže k opravdovým výstupům.



Obrázek 1.3: Algoritmus zpětného šíření chyb a aktualizace parametrů v neuronové síti

1.3 Konvoluční neuronové sítě

Konvoluční neuronové sítě (angl. *convolutional neural network, CNN*) se zpravidla řadí do skupiny hlubokých neuronových sítí. Tento druh sítě má mnoho využití např. v detekci objektů na obrázku. Sítě využívají diskrétní konvoluční vrstvu, pro obrázky se většinou jedná o diskrétní konvoluci ve 2D s maskou **g** pro jednu vrstvu (což je vyobrazeno na obrázku 1.4 a ve vzorci (1.7), proto se tyto sítě nazývají konvoluční.

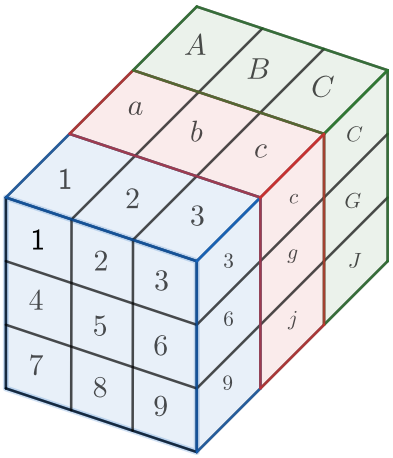


Obrázek 1.4: Znázornění diskrétní konvoluce ve 2D

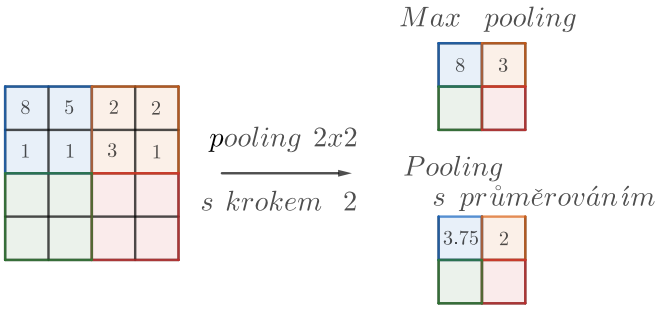
$$y(m,n) = (f * g)(m,n) = \sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} f(i,j)g(m-i,n-j) \tag{1.7}$$

Konvoluční vrstva nám umožní snížit velkosti jednotlivých vrstev a zjistit význačné body ve vrstvách. Obrázky se skládají z pixelů, což jsou nejmenší jednotky. Každý tento pixel je definován pozicí a barvou, společně vytvářejí pole 3 čísel. Barevný obrázek je reprezentován tenzorem 3. řádu, jedná se o 3 matice, které jsou zařazeny za sebou, což můžeme vidět na obrázku 1.5 [4].

Další důležitou vrstvou, která se používá ve spojení s konvoluční vrstvou je poolingová vrstva. Její hlavní funkcí je zmenšit velikost výstupu, což nám umožní rychlejší výpočet a ušetří nám paměť a čas. Je definována velikostí a krokem. V literatuře se běžně objevují 2 typy poolingových vrstev, a to - maximalizační a průměrované pooling vrstvy (obr. 1.6). Může se také použít aktivační funkce jako v ANN. Nakonec se často používá plně propojená vrstva, která nám z výsledků poolingové vrstvy dává výsledek, který požadujeme (například pro predikci z obrázku, zda je to žena, či muž) a funguje jako ANN [4].



Obrázek 1.5: Znázornění tenzoru 3. řádu



Obrázek 1.6: Max pooling a pooling s průměrováním

Mezi známé sítě, které využívají konvoluční vrstvy, patří reziduální neuronová síť (angl. *residual neural network, ResNet*), která patří do skupiny ANN, více v [67]. Jedná se o síť, jejíž základní myšlenkou je zavedení takzvaného „zkratkového připojení identity“, které přeskakuje jednu, nebo více vrstev. Mezi výhody této architektury patří snadnější optimalizace neuronových sítí [66] a zmírnění problému degradace [47].

1.4 Metody pro zrychlení neuronových sítí

V současné době mají modely strojového učení obrovské počty parametrů. Pro snížení počtu parametrů a zjednodušení modelů používáme kvantizaci. Kvantizace je proces, při kterém dochází ke snižování přesnosti vah a prahů. Tento proces snižuje vytížení paměti počítače. Například v neuronové síti lze nahradit číselnou reprezentaci parametrů z floatů (32 bitů) na inty (8 bitů), což výrazně sníží paměť. Nevýhodou kvantizace je ztráta přesnosti dané neuronové sítě, protože parametry nereprezentují informaci přesně. Ale existují kvantizace, které vedou k minimální ztrátě přesnosti. Nejčastěji se tyto procesy využívají při vytváření malých neuronových sítí, kde je třeba malý model, nízká paměťová zátěž nebo i snížení spotřeby energie při trénování apod.

Rozdělujeme kvantizaci na dva hlavní přístupy:

- Kvantizace po trénování (angl. *Post-Training Quantization, PTQ*): Kvantizace po trénování je technika, ve které je neuronová síť zcela trénována a poté je kvantizována. Po tréninku zamrazíme parametry sítě, poté parametry kvantizujeme, a tím vzniká kvantizovaná síť. Ačkoli je tato metoda jednoduchá, může vést k vyšší ztrátě přesnosti, protože všechny chyby související s kvantizací nastanou po dokončení tréninku, a proto je nelze kompenzovat [14].

- Trénink zaměřený na kvantizaci (angl. *Quantization Aware Training, QAT*): Trénink zaměřený na kvantizaci kompenzuje chyby související s kvantizací. Myšlenka je taková, že chyby související s kvantizací se budou hromadit v celkové ztrátě modelu během tréninku a optimalizátor tréninku bude pracovat na odpovídající úpravě parametrů a celkovém snížení chyb. Trénink zaměřený na kvantizaci má výhodu mnohem nižších ztrát než kvantizace po trénování [14].

Mimo kvantizaci lze zefektivnit modely i dalšími způsoby, tyto metody jsou z článku [40]:

- Navrhování efektivních architektur NN modelů: Můžeme měnit mikro-architekturu sítě nebo makro-architekturu sítě. U mikro-architektury se jedná o změny například typu jádra u konvoluce. Kdežto u makro-architektury jsou změny například v typu modulů, jako je reziduální. Klasické techniky se snažily nacházet novou architekturu pomocí "hrubé síly"(angl. *brute force*). Existují taky automatizované algoritmy, které se snaží vyhledávat architekturu sítě, jako jsou automatizované strojové učení (angl. *automated machine learning, AutoML*) [39] a vyhledávání neuronové architektury (angl. *Neural Architecture Search, NAS*) [35].

- Spolunavrhování architektury a hardwaru NN: Další nedávná práce [50] spočívala v přizpůsobení (a společném návrhu) architektury NN pro konkrétní cílovou hardwarovou platformu. To je důležité, protože komponenty NN jsou závislé na hardwaru.

- Prořezávání: Dalším přístupem ke snížení paměti a výpočetních nákladů NN je použití prořezávání. Při prořezávání jsou odstraněny neurony s malou důležitostí (citlivostí), což má za následek řídký výpočetní graf, a to minimálně ovlivní funkci výstupu/ztráty modelu. Metody prořezávání lze obecně rozdělit na nestrukturované prořezávání a strukturované prořezávání.

  S nestrukturovaným prořezáváním se odstraní neurony s malou významností, ať se vyskytují kdekoli. S tímto přístupem lze provádět agresivní prořezávání, odstranění většiny parametrů NN, s velmi malým dopadem na výkon zobecnění modelu. Tento přístup však vede k operacím s řídkými maticemi, o kterých je známo, že je obtížné je urychlit, a které jsou typicky vázané na paměť.

  Na druhou stranu při strukturovaném ořezávání je odstraněna skupina parametrů (např. celé konvoluční filtry). To má za následek změnu vstupních a výstupních tvarů vrstev a váhových matic, což stále umožňuje husté maticové operace. Agresivní strukturované prořezávání však často vede k výraznému zhoršení přesnosti.

- Vědomostní destilace: Tento typ zahrnuje trénování velkého modelu a jeho následné použití jako učitele k trénování kompaktnějšího modelu. Namísto používání „hard“ označení tříd během trénování modelu studenta je klíčovou myšlenkou modelové destilace využití „soft“ pravděpodobností vytvořených učitelem, protože tyto pravděpodobnosti mohou obsahovat více informací o vstupu. Kombinace znalostní destilace s předchozími metodami (tj. kvantizací a prořezáváním) dosahuje významného zefektivnění modelů strojového učení.

Neuronové sítě přinášejí do problému kvantizace jedinečné výzvy a příležitosti. Za prvé, trénování a inference neuronových sítí jsou výpočetně náročné, takže je důležité mít efektivní reprezentaci číselných hodnot. Za druhé, většina současných modelů je přeparametrizována, což znamená, že existuje mnoho příležitostí k redukci počtu parametrů bez významného dopadu na přesnost. Nicméně, výrazný rozdíl spočívá v tom, že neuronové sítě jsou velmi odolné vůči agresivní kvantizaci a extrémní diskretizaci, což znamená, že lze významně snížit počet bitů použitých k reprezentaci parametrů a aktivaci bez významného poklesu přesnosti sítě. To přináší příležitosti pro úsporu paměti, snížení výpočetní náročnosti a rychlejší inference na omezených zařízeních. Přeparametrizované modely jsou běžné, což znamená, že existuje mnoho různých konfigurací modelu, které mohou optimalizovat danou metriku, jako je kvalita klasifikace. Při používání vhodné kvantizace může model dosáhnout dobré přesnosti a generalizace [40].

Dříve se soustředilo na nalezení metod, které by signál příliš nezměnily, nebo na numerické metody, ve kterých existoval kontrolovaný rozdíl mezi „přesným“ a „diskretizovaným“ výpočtem. Neuronové sítě nabízí i další možnosti pro zrychlení běhu sítí [40].

Základní koncepty kvantizace

Předpokládejme, že NN má L vrstev s naučitelnými parametry označenými jako W_1, W_2, ..., W_L, přičemž θ budeme rozumět kombinaci všech těchto parametrů. Bez ztráty na obecnosti se zaměřujeme na problém učení s učitelem, kde cílem je optimalizovat následující empirickou funkci minimalizace rizika [40]:

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} l(x_i, y_i; \theta), \tag{1.8}$$

kde (x, y) jsou vstupní data, odpovídající označení $l(x, y; \theta)$ je ztrátová funkce (např. střední kvadratická chyba nebo ztráta křížové entropie) a N je celkový počet datových bodů. Označme také vstupní skryté aktivace i-té vrstvy jako h_i a odpovídající výstupní skrytou aktivaci jako a_i. Předpokládáme, že máme natrénované parametry modelu θ uložené s přesností float. Při kvantizaci je cílem snížit přesnost parametrů θ, stejně jako přechodných aktivačních map (tj. h_i, a_i) na nízkou přesnost, s minimálním dopadem na výkon/přesnost zobecnění modelu. K tomu potřebujeme definovat kvantizační operátor (popsaný dále), který mapuje float hodnotu na kvantizovanou, což je popsáno dále [40].

Základní koncepty kvantizace jsou součástí většiny kvantizačních algoritmů a jsou nezbytné pro pochopení a aplikaci existujících kvantizačních metod.

Uniformní kvantizace

Nejprve potřebujeme definovat funkci, která dokáže kvantizovat NN váhy a aktivace na konečnou množinu hodnot. Tato funkce přijímá skutečné float hodnoty a mapuje je do rozsahu s nižší přesností. Oblíbená volba pro kvantizační funkci je následující [40]:

$$Q(r) = Int(r/S) - Z, \tag{1.9}$$

kde Q je kvantizační operátor, r je vstup z oboru reálných čísel (aktivace nebo váha), S je faktor měřítka z reálných čísel a Z je celočíselný nulový bod. Kromě toho funkce Int mapuje hodnotu z oboru reálných čísel na celočíselnou hodnotu pomocí operace zaokrouhlení (např. zaokrouhlení na nejbližší hodnotu). V podstatě je tato funkce mapováním z reálných hodnot r na nějaké celočíselné hodnoty. Tato metoda kvantizace je také známá jako jednotná kvantizace, protože výsledné kvantizované hodnoty (tj. kvantizační úrovně) jsou rovnoměrně rozmístěny. Existují také nejednotné kvantizační metody, jejichž kvantizované hodnoty nemusí být nutně rovnoměrně rozmístěny [40].

Je možné obnovit původní hodnoty r z kvantizovaných hodnot $Q(r)$ pomocí operace, která se často nazývá dekvantizace [40]:

$$\widetilde{r} = S(Q(r) + Z), \tag{1.10}$$

kde Q je kvantizační operátor, r je reálný hodnotný vstup (aktivace nebo váha), S je reálný hodnotový faktor měřítka a Z je celočíselný nulový bod.

Symetrická a asymetrická kvantizace

Jedním důležitým faktorem při jednotné kvantizaci je volba měřítka S v rovnici (1.9). Tento škálovací faktor v podstatě rozděluje daný rozsah reálných hodnot r do několika oddílů [40]:

$$S = \frac{\beta - \alpha}{2^b - 1}, \tag{1.11}$$

kde $[\alpha, \beta]$ označuje ohraničený rozsah, kterým ořezáváme skutečné hodnoty, a b je kvantizační bitová šířka. Proto, aby bylo možné definovat faktor měřítka, je třeba nejprve určit rozsah oříznutí $[\alpha, \beta]$. Proces výběru rozsahu ořezu se často nazývá kalibrace. Přímou volbou je použití min/max signálu pro rozsah ořezu, tj. $\alpha = rmin$ a $\beta = rmax$. Tento přístup je asymetrickým kvantizačním schématem, protože ořezový rozsah nemusí být nutně symetrický vzhledem k počátku, tj. $-\alpha \neq \beta$. Je také možné použít symetrické kvantizační schéma výběrem symetrického ořezového rozsahu $\alpha = \beta$. Oblíbenou volbou je zvolit tyto parametry α, β na základě min/max hodnot signálu: $\alpha = \beta = max(|rmax|, |rmin|)$. Asymetrická kvantizace často vede k užšímu ořezovému rozsahu ve srovnání se symetrickou kvantizací. To je zvláště důležité, když jsou cílové váhy nebo aktivace nevyvážené, např. aktivace po ReLU, která má vždy nezáporné hodnoty. Použití symetrické kvantizace však zjednodušuje kvantizační funkci v rovnici (1.9) nahrazením nulového bodu Z = 0 [40]:

$$Q(r) = Int(r/S). \tag{1.12}$$

Zde jsou dvě možnosti pro faktor měřítka. V „plném rozsahu“ je symetrická kvantizace S zvolena jako:

$$S = \frac{2max(|r|)}{2^n - 1} \tag{1.13}$$

(se zaokrouhlováním nahoru), pro využití celého rozsahu INT8 z [-128,127]. V „omezeném rozsahu“ je však S zvoleno jako:

$$S = \frac{max(|r|)}{2^{n-1} - 1}, \tag{1.14}$$

které používá pouze rozsah [-127,127]. Přístup s plným rozsahem je přesnější. Symetrická kvantizace je v praxi široce používána pro kvantizaci vah, protože vynulování nulového bodu může vést ke snížení nákladů na výpočet během inference a také zjednodušuje implementaci [40].

Oblíbenou metodou je použití min/max signálu pro symetrickou i asymetrickou kvantizaci. Tento přístup je však citlivý na odlehlá data v aktivacích. Ta by mohla zbytečně zvětšit dosah a ve výsledku snížit rozlišení kvantizace. Jedním z přístupů, jak to vyřešit, je použít percentil místo min/max signálu. To znamená, že místo největší/nejmenší hodnoty se jako β, α použijí i-té největší/nejmenší hodnoty. Dalším přístupem je výběr α a β pro minimalizaci divergence KL (tj. ztráty informace) mezi reálnými hodnotami a kvantizovanými hodnotami [40].

Symetrická kvantizace rozděluje ořez pomocí symetrického rozsahu. To má výhodu snadnější implementace, protože to vede k Z = 0 v rovnici (1.9). To není však optimální pro případy, kdy by rozsah mohl být zkreslený a nesymetrický. Pro takové případy je preferována asymetrická kvantizace [40].

Statická vs dynamická kvantizace

Existují dva základní přístupy, jak stanovit ořezový rozsah: dynamická a statická kvantizace. Při dynamické kvantizaci je ořezový rozsah vypočítán pro každou aktivační mapu během běhu na základě statistik signálu. Aktivační mapy se však pro každý vstupní vzorek liší ($\boldsymbol{x}$ v rovnici (1.8)). Tento přístup často vede k vyšší přesnosti, ale vyžaduje výpočet statistik (min, max, percentil atd.) v reálném čase. Při statické kvantizaci je ořezový rozsah vypočítán předem a je použit během inference. Tento přístup je méně náročný na výpočetní výkon, ale obvykle vede k nižší přesnosti. Existuje několik různých metod pro výpočet ořezového rozsahu, včetně minimalizace střední kvadratické chyby mezi původní distribucí vah a odpovídajícími kvantizovanými hodnotami. Dalším přístupem je naučit se tento rozsah během tréninku neuronové sítě [40], [21].

Nerovnoměrná (neuniformní) kvantizace

Formální definice nerovnoměrné kvantizace je uvedena v rovnici (1.15), kde X_i představuje diskrétní kvantizační úrovně a Δi kvantizační kroky (prahové hodnoty) [40]:

$$Q(r) = X_i, \text{ if } r \in [\Delta i, \Delta i + 1). \tag{1.15}$$

Konkrétně, když hodnota reálného čísla r spadá mezi kvantizační krok Δi a $\Delta i+1$, kvantizér Q ji promítne na odpovídající kvantizační úroveň X_i. Všimněme si, že ani X_i, ani Δi nejsou rovnoměrně rozmístěny. Nejednotná kvantizace může dosáhnout vyšší přesnosti, protože je možné lépe zachytit distribuce tím, že se více zaměříme na oblasti důležitých hodnot nebo nalezneme vhodné dynamické rozsahy [40].

Typickou nejednotnou kvantizaci využívá logaritmické rozdělení, kde se kvantizační kroky a úrovně zvyšují exponenciálně namísto lineárně. Obecně nám nejednotná kvantizace umožňuje lépe zachytit informaci o signálu přiřazováním bitů a nerovnoměrnou diskretizací rozsahu parametrů. Nejednotná kvantizace se obtížně využívá v praxi kvůli hardwarové složce (GPU a CPU) [5].

Metody ladění

Často je nutné po kvantizaci upravit parametry v NN. To lze provést procesem, který se nazývá trénování zaměřené na kvantizaci, nebo procesem kvantizace po trénování. Tyto dva přístupy jsme již zmínili výše. Hlavní nevýhodou QAT jsou však výpočetní náklady na přetrénování modelu NN. PTQ je velmi rychlá metoda pro kvantizování NN modelů. To však často přichází za cenu nižší přesnosti ve srovnání s QAT.

Dále máme kvantizaci nazývanou Zero Shot, která ke kvantizaci nepotřebuje znát trénovací data, ani jiné apriorní informace, více v článku [28].

Stochastické kvantizování

Stochastická kvantizace zaokrouhluje číslo nahoru nebo dolů s pravděpodobností spojenou s velikostí aktualizace váhy. Například operátor Int v rovnici (1.9) je definován jako:

$$Int(x) = \begin{cases} \lfloor x \rfloor \text{ s pravděpodobností } \lceil x \rceil - x \\ \lceil x \rceil \text{ s pravděpodobností } x - \lfloor x \rfloor \end{cases}, \tag{1.16}$$

tuto definici však nelze použít pro binární kvantizaci, proto rozšíříme rovnici:

$$Binary(x) = \begin{cases} 1 \text{ s pravděpodobností } 1 - \sigma(x) \\ -1 \text{ s pravděpodobností } \sigma(x) \end{cases}, \tag{1.17}$$

kde Binary je funkce pro binarizaci reálné hodnoty x a $\sigma()$ je funkce sigmoid. Nedávno byla představena další metoda QuantNoise stochastické kvantizace, více v článku [36]. O pokročilejších metodách zabývajících se kvantizací pod 8 bitů se dočteme v článku [40].

Kvantizace konvolučních vrstev

V oblasti počítačového vidění se využívají konvoluční neuronové vrstvy, které obsahují mnoho různých konvolučních filtrů. Každý z těchto filtrů má jiný rozsah hodnot, a to může být problém při kvantizaci vah, tedy při ořezávání jejich rozsahu na pevně danou velikost. Existují různé způsoby, jak tuto kvantizaci provádět, jako například kvantizace po vrstvách, skupinová kvantizace, kvantizace přes kanály atd. Každý z těchto přístupů má své výhody a nevýhody a vybírá se podle konkrétních potřeb a vlastností vrstev, více v článku [40] a [72].

Kapitola 2

Detekce a sledování chodců

2.1 Detekce chodců

Detekce objektů je v oblasti počítačového vidění velmi náročným úkolem. Spočívá v nalezení objektů, kde na vstupu je obrázek a výstupem je množina objektů definována názvem objektu a souřadnice jejich ohraničujícího rámečku. Hluboké učení se ukázalo jako velmi účinná metoda pro detekci různých objektů, jako jsou například lidská těla, auta nebo zvířata. Detekce objektů slouží jako základ pro další úkoly, jako je segmentace, sledování objektů, rozpoznávání aktivit atd. Detekci objektů využíváme například v robotickém vidění nebo autonomním řízení. Existují dva typy detekce objektů - specifická detekce objektů, která se zaměřuje na konkrétní objekty (např. Leonardo DiCaprio nebo sousedovo auto), a obecná detekce objektů, která rozpoznává objekty a zařazuje je do předem definovaných kategorií nebo tříd (např. lidé, auta a psi).

Detekci objektů rozdělujeme na dvě období - tradiční detekce objektů a detekce založené na hlubokém učení. Většina tradičních detektorů byla postavena na základě manuálně navržených příznaků. Mezi tradiční detektory můžeme zařadit Viola Jones detektor [68], histogram orientovaných gradientů (angl. *histogram of oriented gradients, HOG*) [30] a deformovatelný model založený na částech (angl. *deformable part-based model, DPM*) [38]. Tyto detektory jsou často omezené v detekci složitých objektů a mají nižší přesnost než detektory založené na hlubokém učení.

Algoritmy založené na hlubokém učení dále rozdělujeme na jednofázové a dvoufázové. Mezi dvoufázové detektory lze zařadit FPN (angl. *feature pyramid networks*) [54], SPPNet (angl. *spatial pyramid pooling networks*) [46], Fast RCNN [41] a další. Tyto detektory pracují ve dvou krocích - zahrnují krok předzpracování, takže se nejprve vytvoří oblasti zájmu, a poté se provede detekce objektů v těchto oblastech. Jsou složitější na trénování, ale z historických dat mají lepší výsledky než jednofázové detektory. Mezi jednofázové detektory lze zařadit You Only Look Once (YOLO) [61], Single Shot MultiBox Detector (SSD) [57] a RetinaNet [55], [38]. Jednofázové detektory mají jen jednu část, nerozděluje předzpracování a detekci. Tyto detektory jsou rychlejší, ale méně přesné.

Metody počítačového vidění pro detekci chodců

Následující metody jsou převzaty z článku [27]. První přístupy pro detekci a sledování pohybujících se objektů ve videu pořízeném statickou kamerou spočíval v odečítání pozadí. Tato technika umožňuje detekci a rozlišení pohybujících se objektů uvnitř scény pomocí vhodného modelu pozadí. I když jsou algoritmy založené na odečítání pozadí poměrně jednoduché na implementaci, tento přístup není odolný vůči různému osvětlení, dynamickému pozadí, stínům nebo šumu, což omezuje jeho použití.

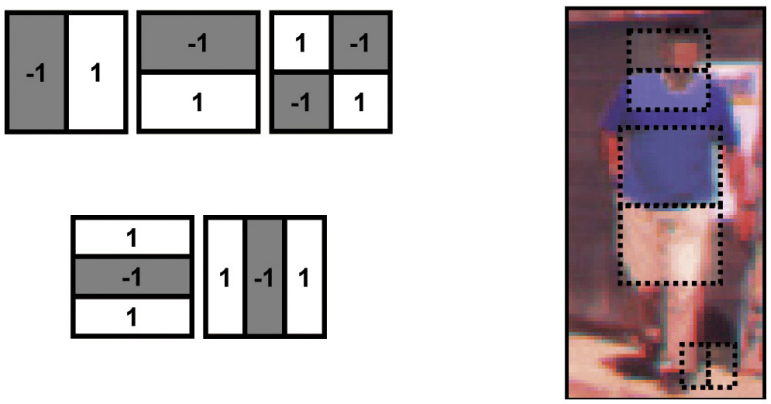
V posledních letech bylo vyvinuto a testováno velké množství algoritmů pro provádění detekce a sledování lidí, většina z nich je založena na následujících přístupech pro extrakci a detekci příznaků:

1. Histogramy orientovaných gradientů: Metoda využívá gradienty intenzity a směrů hran k charakterizaci vzhledu a tvaru lokálních objektů, což je vyobrazeno na obrázku 2.1. Snímek videa je rozdělen na malé oblasti a v každé oblasti se vypočítává 1-D histogram směrů gradientu nebo orientace hran. Pro normalizaci histogramů se používá vylepšená verze algoritmu, která řeší problémy související s osvětlením nebo stínováním. Histogramy jsou považovány za reprezentaci obrazu.

Obrázek 2.1: Výstupní obrázek ukazující superpozici deskriptorů HOG na vstupní obrázek [27]

2. Haarovy příznaky: 2-rozměrné Haarovy vlnky zahrnují základní funkce, které zachycují změnu intenzity podél horizontálního, vertikálního a diagonálního (nebo rohového) směru, obr. 2.2. Stejně jako v předchozím případě je každá reprezentace použita jako vstup do klasifikátoru.

3. Viola-Jones příznaky: Tento přístup je rozšířenou verzí obdélníkových filtrů prezentovaných Violou a Jonesem pro detekci statické tváře; konkrétně tento přístup bere v úvahu konkrétní filtry založené na Haarových vlnkách/příznacích. V tomto případě využíváme informace o směru i intenzitě, a to i s ohledem na sekvence snímků.



Obrázek 2.2: Reprezentace Haarových příznaků na snímku [27]

4. Texturní příznaky: Extrakce vlastností z textury znamená získání informací o distribuci textury v obrázku a je poměrně jednoduchá. Avšak použití pouze texturních příznaků pro klasifikaci chodců je náročné, protože třídy se mohou výrazně lišit, například kvůli různým světelným podmínkám. Proto se texturní příznaky často kombinují s jinými druhy příznaků, jako jsou tvary, barvy a další, aby se zvýšila úspěšnost klasifikace.

5. Místní binární vzor (angl. *local binary pattern, LBP*): Tato technika umožňuje popsat obrázky na základě jejich textury vhodným zvážením okolí každého pixelu. LBP přístup se stal velmi populární díky své lepší odolnosti vůči změnám pozice a osvětlení než u jiných metod. Vektory příznaků LBP se velmi často používají v kombinaci s příznaky HOG k dosažení vyššího výkonu při detekci chodců.

6. Strojové učení:

   (a) Tradiční metody: Pro detekci a sledování chodců používají jednoduché klasifikátory, jsou-li jim dané příznaky, jako jsou Support Vectors Machines (SVM) [63] nebo rozhodovací stromy a další metody.

   (b) Metody hlubokého učení: V nedávné době se v oblasti umělé inteligence rozšířily moderní techniky strojového učení, které využívají hluboké sítě. Nejčastěji používanou architekturou hlubokého učení jsou konvoluční neuronové sítě. Vzhledem k tomu, že trénování konvolučních neuronových sítí je velmi náročné na čas a výpočetní zdroje, tak pro konvoluční neuronové sítě máme mnoho přístupů, mezi nejznámější patří přeučení (angl. *transfer learning*). Jedná se o techniku, která dokáže využít před-trénované modely pro řešení nových úkolů, ale podobných.

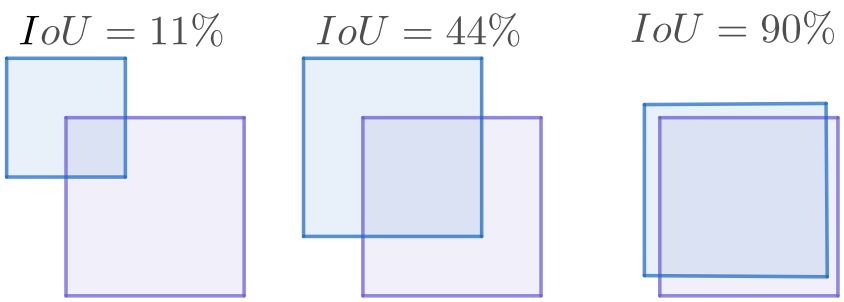
2.1.1 Metriky pro detekci objektů

Základní pojmy

Pro kvalitu detekce se využívá metriky zvané "průnik ku sjednocení"(angl. *intersection over union, IoU*), což můžeme vidět znázorněno na obrázku 2.3. Uvažujme správný ohraničující rámeček $\boldsymbol{g}$ spojený s třídou y a hypotézu detekce $\boldsymbol{x}$ ohraničujícího rámečku $\boldsymbol{b}$. Protože $\boldsymbol{b}$ obvykle zahrnuje objekt a určité plochy pozadí, může být obtížné určit, zda je detekce správná, nebo ne. To je obvykle řešeno IoU metrikou:

$$IoU(\boldsymbol{b}, \boldsymbol{g}) = \frac{\boldsymbol{b} \cap \boldsymbol{g}}{\boldsymbol{b} \cup \boldsymbol{g}} \tag{2.1}$$

[29].



Obrázek 2.3: Metrika IoU pro různé hodnoty

S touto metrikou dále můžeme definovat [6]:

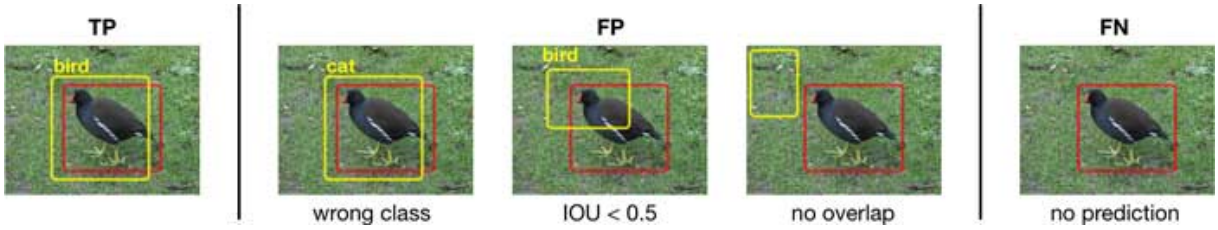
- Správná zařazení do výsledků (angl. *True Positive, TP*): Správná detekce. Detekce s prahem IOU ≥ prahová hodnota, kterou si stanovíme.
- Nesprávná zařazení (angl. *False Positive, FP*): Špatná detekce, byl nalezen objekt, který tam není. Detekce s IOU < prahová hodnota.
- Nesprávná vynechání (angl. *False Negative, FN*): Správný ohraničující rámeček není detekován.
- Správná vynechání (angl. *True Negative, TN*): Představoval by nesprávnou detekci. V úloze detekce objektů existuje mnoho možných ohraničovacích rámečků, které by neměly být detekovány v rámci obrázku. TN by tedy byly všechny možné ohraničující boxy, které nebyly detekovány (tolik možných krabic v rámci obrázku). Proto ji metriky nepoužívají.

Tyto pojmy jsou znázorněny v obrázku 2.4. Pomocí výše nadefinovaných pojmů lze definovat přesnost (angl. *precision, P*) a úplnost (angl. *recall, R*):

$$P = \frac{TP}{TP + FP} \tag{2.2}$$

$$R = \frac{TP}{TP + FN} \tag{2.3}$$

Můžeme tedy definovat metriky pro detekci objektů.



Obrázek 2.4: Znázornění pojmů TP, FP a FN [17]

Křivka přesnosti a úplnosti

Křivka přesnosti a úplnosti je užitečný nástroj pro vyhodnocení výkonu detektoru objektů. Pro každou třídu objektů máme vlastní křivku. Detektor objektů určité třídy je považován za dobrý, pokud jeho přesnost zůstává vysoká a zároveň neklesá jeho úplnost. To znamená, že při změně prahové hodnoty zůstanou přesnost a úplnost stále vysoké [6].

Průměrná přesnost

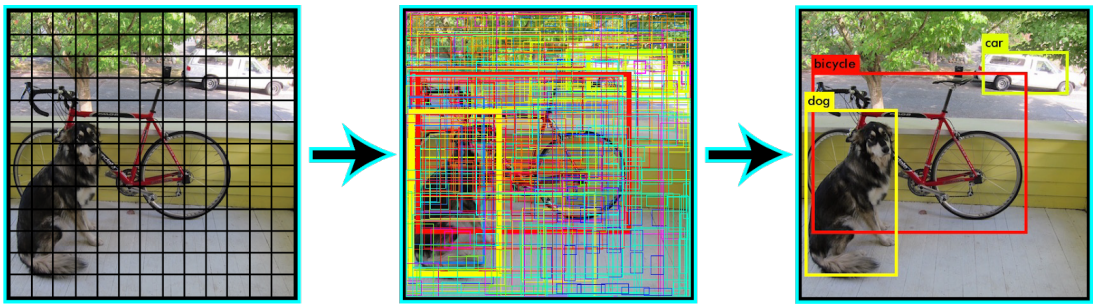
K vyhodnocení detekce běžně používáme křivku přesnosti a úplnosti, ale průměrná přesnost dává numerické hodnoty. Pak je snadné porovnat výkon s jinými modely. Průměrná přesnost (angl. *Average Precision, AP*) je váženým průměrem přesností pro každou prahovou hodnotu se zvyšující se úplností. Tento výpočet se provede pro každý objekt.

Střední hodnota průměrné přesnosti

Střední průměrná přesnost (angl. *Mean Average Precision, mAP*) je rozšíření průměrné přesnosti. V průměrné přesnosti vypočítáme pouze jednotlivé objekty, ale mAP dává přesnost pro celý model. Chceme-li najít procento správných předpovědí v modelu, tak používáme mAP [6].

2.1.2 YOLO v7

Algoritmus YOLO verze 7 patří mezi nejlepší detektory objektů. Je to jednofázový detektor, který dokáže nejen detekovat objekty, ale také určit jejich pozici a segmentaci. Díky své rychlosti je využíván v aplikacích v reálném čase. YOLO navrhuje použití end-to-end neuronové sítě, která umožňuje provádět predikce ohraničujících rámečků a pravděpodobností tříd současně. I když se YOLO jeví jako nejlepší algoritmus pro detekci objektů, stále má některá omezení. Například malé objekty, které se vyskytují přirozeně ve skupinách jsou pro YOLO obtížné detekovat a lokalizovat. Navíc YOLO dosahuje nižší přesnosti v porovnání s mnohem pomalejšími algoritmy detekce objektů, jako je Fast RCNN.



Obrázek 2.5: Popis YOLO v7 - rozdělí obrázek na mřížku, v každé buňce proběhne regrese i klasifikace, nakonec potlačí duplikáty

Architektura YOLO v7

YOLOv7 se skládá ze čtyř částí: hluboké konvoluční sítě, tzv. páteř sítě (angl. *backbone*), rozšířené příznakové pyramidy (angl. *Feature Pyramid Network, FPN*), hlavní sítě a dekodéru. Páteř sítě se skládá z několika konvolučních vrstev, které extrahují příznaky z různých úrovní vstupního obrazu. FPN se používá ke propojení jednotlivých úrovní, aby se zlepšila přesnost detekce. Díky FPN je síť robustní vůči měřítku. Hlavní síť se skládá z několika konvolučních a lineárních vrstev, jejichž výstup jsou pravděpodobnostní mapy, které obsahují predikce pro každou třídu a souřadnice ohraničujícího rámečku. Dekodér pak zpracovává výstupy hlavní sítě a vytváří výsledné predikce, konkrétněji probíhá NMS (angl. *Non-Maximum Suppression, NMS*). NMS je algoritmus, jehož hlavním účelem je snížení počtu duplikátních detekcí objektů v obrázku, více v článku [59]. Autoři YOLOv7 se snažili vylepšit architekturu sítě YOLOv6 [51], aby byla rychlejší a přesnější než její předchozí verze, více v článku [24].

2.2 Sledování chodců

Sledování objektů znamená, že algoritmus monitoruje pohyb objektu ve videu. Tento proces zahrnuje odhadnutí polohy a dalších důležitých informací o pohybujících se objektech. Jedná se o algoritmy jejíž vstupem jsou informace o detekcích a výstupem je přiřazení každé detekci informace o identitě. Sledování objektů obvykle začíná detekcí objektů, kterou jsme popsali v kapitole 2.1. Existují dva typy sledování objektů: sledování objektu v jednom videu a sledování objektů v sérii snímků. Tento proces lze také rozdělit na sledování jediného objektu nebo více objektů najednou. Existují problémy, kterým zpravidla musíme čelit při sledování objektů, příkladem je okluze objektů ve videích. Aby se tomu zabránilo, lze implementovat i citlivost na okluzi do daného algoritmu.

Sledování objektu se skládá z 4 hlavních kroků. Prvním krokem je detekce objektu, kde se určí, který objekt má být sledován. Poté musí sledovač odhadnout nebo předpovědět polohu objektu v dalších snímcích a vykreslit ohraničující rámeček kolem něj. Druhým krokem je modelování vzhledu, které se zaměřuje na zachycení vizuálního vzhledu sledovaného objektu. Když objekt prochází různými scénami, může se jeho vzhled změnit a to může vést k chybám v algoritmu sledování. Modelování vzhledu musí být schopné zachytit různé změny a deformace, které se mohou vyskytnout při pohybu objektu. Skládá se ze dvou složek [3]:

1. Vizuální reprezentace: Zaměřuje se na konstrukci robustních prvků a reprezentací, které mohou popisovat objekt.
2. Statistické modelování: Využívá techniky statistického učení k efektivnímu vytváření matematických modelů pro identifikaci objektů.

Třetím krokem sledování objektu je odhad pohybu objektu. To znamená, že algoritmus předpovídá, kam se objekt bude pohybovat v budoucích snímcích. Poté algoritmus aproximuje oblast, kde by se objekt mohl s největší pravděpodobností nacházet. Nakonec se použije vizuální model k přesnému zaměření na polohu cíle [3].

2.2.1 Algoritmy pro sledování chodců

IoU Tracker

Sledovač objektů s využitím metriky IoU používá hodnoty překryvu mezi rámečky objektů detekovaných mezi dvěma po sobě jdoucími snímky k propojení těchto rámů nebo k přiřazení nového ID, pokud není nalezena shoda [16].

Přesné detekce a využití videozáznamu s vysokou snímkovou frekvencí může značně zjednodušit úlohu sledování. Metoda je založena na předpokladu, že detektor vytváří detekci na každém snímku pro každý objekt, který má být sledován, tj. v detekcích nejsou žádné, nebo jen několik „mezer“. Dále předpokládáme, že detekce objektu v po sobě jdoucích snímcích mají nezaměnitelně vysoký přesah IoU, což se běžně stává při použití dostatečně vysokých snímkových frekvencí. Pokud jsou splněny oba požadavky, sledování se stává triviálním a lze jej provádět i bez použití obrázkových informací. Tento jednoduchý sledovač, který v podstatě pokračuje ve sledování tím, že přiřazuje detekci s nejvyšší IoU k poslední detekci v předchozím snímku, pokud je splněna určitá prahová hodnota σ_{IoU}. Všechny detekce nepřiřazené k existující detekci zahájí novou. Všechny detekce bez přiřazené detekce skončí [26].

Vstupem je textový soubor, který na první pozici má název snímku, další jsou název třídy objektů, souřadnice ohraničujícího obrázku (x_1,y_1, x_2, y_2), práh detekce.

DeepSORT

Je přístup ke sledování více objektů, který se zaměřuje na jednoduché a efektivní algoritmy. Pro sledování používá Kalmanovo filtrování v obrázkovém prostoru a asociaci dat snímek po snímku pomocí metody s asociační metrikou, Kalmanovo filtrování je vysvětleno v článku [71]. Metrika měří překrytí hraničních rámečků a tímto způsobem dosahuje vysokého výkonu při vysokých frekvencích snímků. SORT tak dokáže kombinovat informace o poloze a pohybu objektů. Autoři algoritmu DeepSort vylepšili původní sledovací algoritmus tzv. SORT (angl. *Simple Online and Realtime Tracking, SORT*) integrací informací o vzhledu objektů, což jim umožňuje sledovat objekty po delší dobu i při okluzích a tím snižovat počet chybějících identit [16].

BYTE and ByteTrack

Většina metod sledování objektů přiděluje identitu objektům na základě detekčních rámečků s vysokým skóre. Avšak objekty s nízkým skóre jsou ignorovány, což vede k chybějícím objektům a chybným trajektoriím. Autoři navrhují novou asociační metodu, která přiděluje identitu téměř každému detekčnímu rámečku. Využívají podobnosti s předchozími trajektoriemi, tzv. tracklety, k obnovení skutečných objektů a filtrování falešných detekcí. O trakletech více v článku [64]. ByteTrack kombinuje tuto asociační metodu s výkonným detektorem YOLOX [16].

Mezi další algoritmy pro sledování patří například FairMOT [74] a StrongSORT [34].

2.3 Pohlaví a věk

Predikce pohlaví a věku představuje další výzvu v oblasti počítačového zpracování obrazu. Existuje mnoho způsobů, jak predikovat pohlaví a věk, nicméně nejčastěji se využívá hlubokého učení. K predikci lze využít různé vizuální příznaky, které by algoritmus mohl používat, jako jsou oblečení, tvary těla, šperky a účesy. Identifikace pohlaví na základě obličeje je také možná, avšak v případech jako jsou chodci, kdy jsou k dispozici fotografie nízké kvality, obličeje jsou špatně natočené nebo jsou zakryté rouškou, může selhat. Proto je vhodné použít pro predikci pohlaví celé tělo.

Mezi ženami a muži existuje značný rozdíl z biologického hlediska, např. tvar těla je dost odlišný. Při detekci pohlaví a věku z celého těla může volné oblečení jednotlivých chodců dělat problémy.

Využití predikcí pohlaví a věku má mnoho využití:

1. Sběr informací pro efektivitu reklam.

2. Zkoumání jednání nebo chování lidí v obchodním centru.

Objevily se různé klasifikační techniky pro identifikaci pohlaví a věku - většina jich je založená na predikci z lidské tváře a nebo na lidském těle. Predikce pohlaví a věku na základě obličeje je těžká hlavně v případech, kde obrázky mají nízké rozlišení, jsou daleko od kamery apod. Predikce pohlaví a věku podle lidského těla je taky těžká, ale dalo by se využít informací, jak už bylo zmíněno: oblékání, tvaru těla. Tyto informace mohou být užitečné a cenné i v situacích s obrázky, které mají nízké rozlišení [33].

2.3.1 Datasety

Díky obrovskému rozvoji v oblasti strojového učení, existuje velké množství datasetů. Lze najít obrovské datasety, které jsou veřejně dostupné. Umožňuje nám to vytvářet datasety dostatečně velké a rozmanité, což může vylepšit výkon algoritmu. Pro rozpoznávání objektů datasety nefungují jen jako data k trénování, ale používají se pro měření a porovnávání výkonů různých modelů a algoritmů. Většina algoritmů vyhodnocuje svůj výkon na veřejně dostupných datech. Pro dotrénování pohlaví v našem detektoru budeme potřebovat dataset lidí a rozřazení do jednotlivých pohlaví. Veřejně dostupné datasety pro predikci pohlaví z celého těla jsou uvedeny v tabulce 2.1. Hledali jsme pouze dataset pohlaví, protože to bude využito v praktické části, tyto datasety mohou sloužit i jako datasety pro detekci chodců.

Název datasetu	Počet obrázků	-	Dostupnost
Market1501-Attributes	1.500	Muž, žena	[56]
MIAP	100.000	Muž, žena a neznámé	[18]
Richly Annotated Pedestrian (RAP)	41.585	Muž, žena	[52]
Pedestrian attribute recognition at far distance (PETA)	19.000	Muž a žena	[32]

Tabulka 2.1: Veřejně dostupné datasety pro predikci pohlaví z celého těla

Kapitola 3

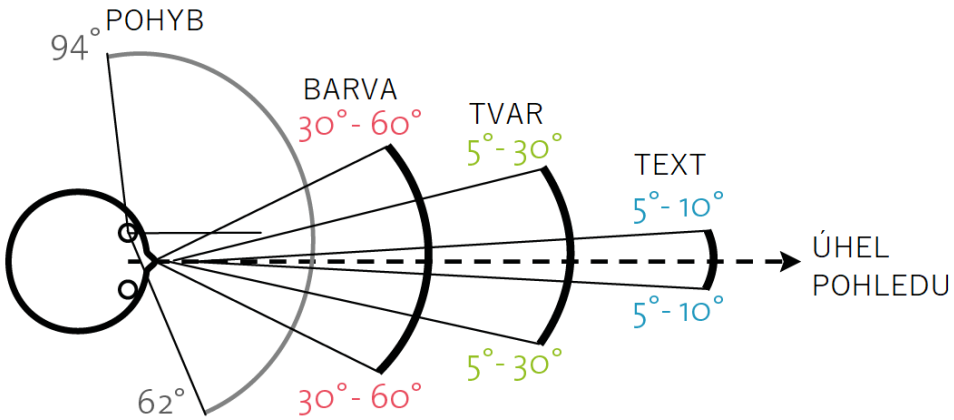
Odhad pozice hlavy (pohledu)

Sledování lidského těla a jeho natočení představuje zajímavou úlohu, kterou mnoho výzkumníků řeší. Pro různé účely je třeba lépe porozumět lidským pohybům. Jedním z těchto pohybů je odhad pozice hlavy (angl. *head pose estimation, HPE*). Formálně tento problém je definován třemi úhly - předklon/záklon, otočení a úklon, co můžeme vidět na obrázku 3.2. V současnosti se většina algoritmů zaměřuje na detekci obličeje a následně odhaduje jeho natočení. Bohužel tyto algoritmy mají určitá omezení, např. při nízkém rozlišení se odhad směru pohledu stává náročným problémem.

Dále existují algoritmy, které odhadují pozici hlavy bez nutnosti předchozí detekce obličeje nebo lokalizace orientačních bodů na obličeji. Tento algoritmus funguje i na malé obličeje, mezi tyto algoritmy patří například img2pose [25].

Odhad pozice hlavy a směru pohledu se využívá v různých aplikacích například:

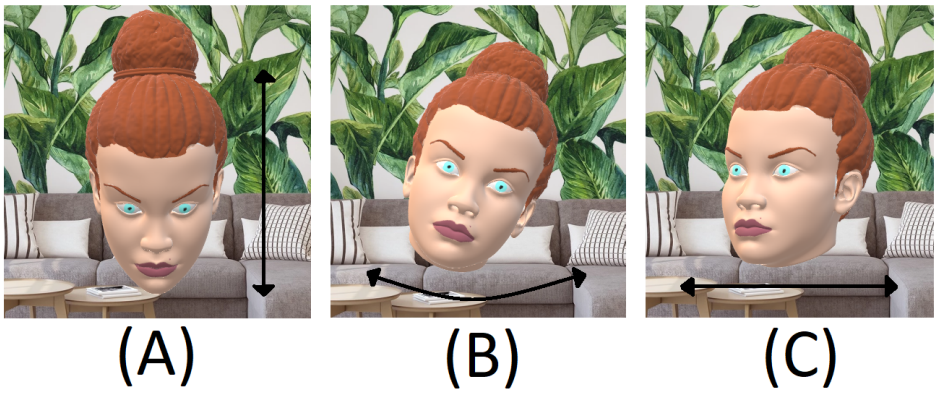
- Bezpečnost jízdy: Tyto algoritmy jsou zvláště užitečné pro řidiče, neboť jim poskytují výstražné signály, když se v okolí pohybuje člověk v nebezpečné vzdálenosti a nevěnuje jim pozornost.
- Cílená reklama: Jedná se o systémy, které se snaží vyhodnocovat efektivitu reklam na reklamních bannerech. Tento systém může určovat specifické informace o daném člověku, jako jsou například pohlaví, věk, emoce atd. Navíc můžeme určovat, zda člověk má danou reklamu ve svém zorném poli. Omezení zorného pole je znázorněno na obrázku 3.1.



Obrázek 3.1: Citlivost na různé jevy v závislosti na zorném poli člověka [7]

3.1 Metody pro odhady pozice hlavy

Následující metody pro odhad pozice hlavy jsou převzaty z článku [49]. Metody pro odhad pozice hlavy se snaží predikovat 3 úhly - úhel předklon/záklon (A), úklon (B) a otočení (C). Tyto úhly jsou postupně vyobrazeny v obrázku 3.2.



Obrázek 3.2: Úhly pro odhad pozice hlavy (A) předklon/záklon (B) úklon (C) otočení

1. Metody založené na 2D vzhledu: Tyto algoritmy předpokládají, že mezi 3D pozicí obličeje a vlastnostmi 2D obrazu obličeje vždy existuje určitý vztah. K odhalení tohoto vztahu se využívá velké množství obrázků a technik statistického učení. Jedná se o druh učení, které nezasahuje do trénovacích dat po dokončení trénování a není schopen se učit z nových dat během běhu.

   Tyto přístupy jsou poměrně jednoduché a vhodné pro obrázky s nízkým i vysokým rozlišením. Navíc během tréninkového procesu nejsou potřeba žádná tréninková data neobsahující obličeje. Model lze snadno upravit, což umožňuje architektuře v případě potřeby přizpůsobit se měnícím se podmínkám.

   Lze také uvést některé závažné nedostatky jako u jiných metod. Detekce hlavy musí být spolehlivá. V případě chybné lokalizace hlavy to vede k vážnému zhoršení přesnosti. Podobně se tyto metody stávají nespolehlivé, když se objeví změny ve vzhledu - s brýlemi, vousy apod.

2. Geometrické metody: Tyto techniky vyžadují lokalizaci klíčových bodů obličeje, jako jsou oči, obočí, nos, rty atd. Následně se odhaduje pozice na základě relativních pozic těchto klíčových bodů, například pomocí strojového učení. Tento přístup se opírá o rozdíly v různých pozicích, jako je odchylka symetrie hlavy atd. Výhodou geometrických metod je, že jsou výpočetně efektivní a většina extrahovaných klíčových bodů je poměrně odolná vůči translační a rotační invarianci. Problém nastává, pokud se klíčové body obličeje detekují chybně, což způsobí špatné určení natočení hlavy.

3. Víceúlohové metody: Víceúlohové metody jsou také známé jako hybridní metody. Hlavní cílem je spojit odhad pozice hlavy s dalšími problémy analýzy obrazu obličeje - rozpoznávání pohlaví, rozpoznávání výrazu obličeje, klasifikace ras atd.

   Metody hlubokého učení jsou také velmi účinné při řešení víceúkolového přístupu. Například metoda pod názvem HyperFace [60] nejprve extrahuje rysy pomocí konvolučních neuronových sítí a poté provádí několik úkolů v jediném frameworku: detekce obličeje, lokalizace orientačních bodů, klasifikace pohlaví a odhad pozice hlavy.

   Jelikož víceúlohové metody získávají informace z více podnětů a spojují odhady z každého systému nezávisle, to zvyšuje přesnost odhadu. Navíc jsou takové frameworky často navrženy bez potřeby inicializace nebo omezení posunu, takže lze překonat omezení jedné konkrétní kategorie pozice hlavy.

4. Sledovací metody: Jedná se o výkonné metody, které využívají časové informace již sledovaných částí hlavy. Dosahují přiměřené přesnosti, obvykle využívající afinní transformace k odhadu změn polohy hlavy, například je možné předpovídat relativní změnu polohy jako posun, který minimalizuje chybu odhadu ve smyslu nejmenších čtverců, pomocí stereo kamerové soupravy.

5. Regresní (nelineární) metody: Tyto algoritmy odhadují polohu hlavy tím, že se naučí funkční mapování z prostoru obrazu do různých pozic hlavy. Hlavní síla těchto metod spočívá v tom, že poskytnutím sady označených tréninkových snímků lze generovat nový model, jež je schopen předpovídat pozici hlavy pro nové vzorky dat.

   Neuronové sítě se používají v literatuře často jako nelineární regresní nástroj pro úlohu HPE. Využívají učení s učitelem a hlubokých neuronových sítí. Jakmile se neuronová síť natrénuje, může být velmi výpočetně efektivní.

6. Metody založené na hlubokém učení s konvolučními neuronovými sítěmi: Přístupy hlubokého učení, které využívají konvoluční neuronové sítě, dokáží řešit různé obtížné vizuální úkoly.

7. Mezi další metody pro odhad pozice hlavy patří například metody pomocí variet, 3D registrace hlavy založená na modelování, detekčního pole, více v článku [49].

3.2 img2pose

Tento algoritmus budeme používat v praktické části této práce. Algoritmus img2pose používá konvoluční neuronové sítě k zpracování obrazu a poskytuje odhad orientace obličeje v šesti stupních volnosti (6DoF), což zahrnuje tři úhly rotace a tři posuny vůči kameře. Tento odhad je jednodušší než detekce orientačních bodů obličeje a zahrnuje pouze rotaci a posun obličeje v prostoru bez deformace obličeje. Algoritmus byl trénován na WIDER FACE datasetu. Nabízí snadno trénovatelný a účinný model založený na rychlém R-CNN, který odhaduje pózu pro všechny tváře na fotografii bez předběžné detekce tváře. Póza se převádí a přitom udržuje konzistentní výstup mezi vstupní fotografií a libovolnými výstřižky fotografie vytvořenými při trénování a vyhodnocování [25].

Máme-li obrázek **I** ve kterém odhadujeme 6DoF pro každou tvář. Pro každý i-tý obličej dostaneme popisek ozn. $\boldsymbol{h}_i \in \mathbb{R}^6$:

$$\boldsymbol{h}_i = (r_x, r_y, r_z, t_x, t_y, t_z), \tag{3.1}$$

kde (r_x, r_y, r_z) představuje vektor rotace a (t_x, t_y, t_z) je vektor translace. Je dobře známo, že 6DoF pozice tváře $\boldsymbol{h}$, může být převedena na vnější kamerovou matici pro promítání 3D tváře do 2D obrazové roviny. Za předpokladu známých vnitřních parametrů kamery lze 3D obličej zarovnat s obličejem na fotografii [25].

Tento detektor je dvoufázový založeným na Rychlejší R-CNN. První fází je síť pro návrh regionů (angl. *region proposal network, RPN*) s pyramidou příznaků, navrhuje potenciální umístění obličejů na snímku. Druhou fází je extrakce příznaků z každého návrhu regionů se sdruženou oblastí zájmu (angl. *region of interest, ROI*) a poté to předá dvěma různým hlavám (obvykle používá pro označení horní části sítě): standardnímu klasifikátoru obličeje/neobličeje a novému regresoru pozice obličeje 6DoF [25]. Více informací se dočtete v článku [25].

Kapitola 4

Kalibrace kamery

Kalibrace kamery je nezbytným krokem ve 3D počítačovém vidění, aby bylo možné získat metrické informace z 2D obrázků. V našem případě budeme chtít vědět, kde osoba stojí v reálném světě. Kalibrace kamery nám řekne, jak se promítá reálný obraz ve 3D do obrázku ve 2D. Jedná se o hledání mapovací funkce z reálného prostředí do obrázku.

Kalibrační techniky můžeme klasifikovat do těchto kategorií [76]:

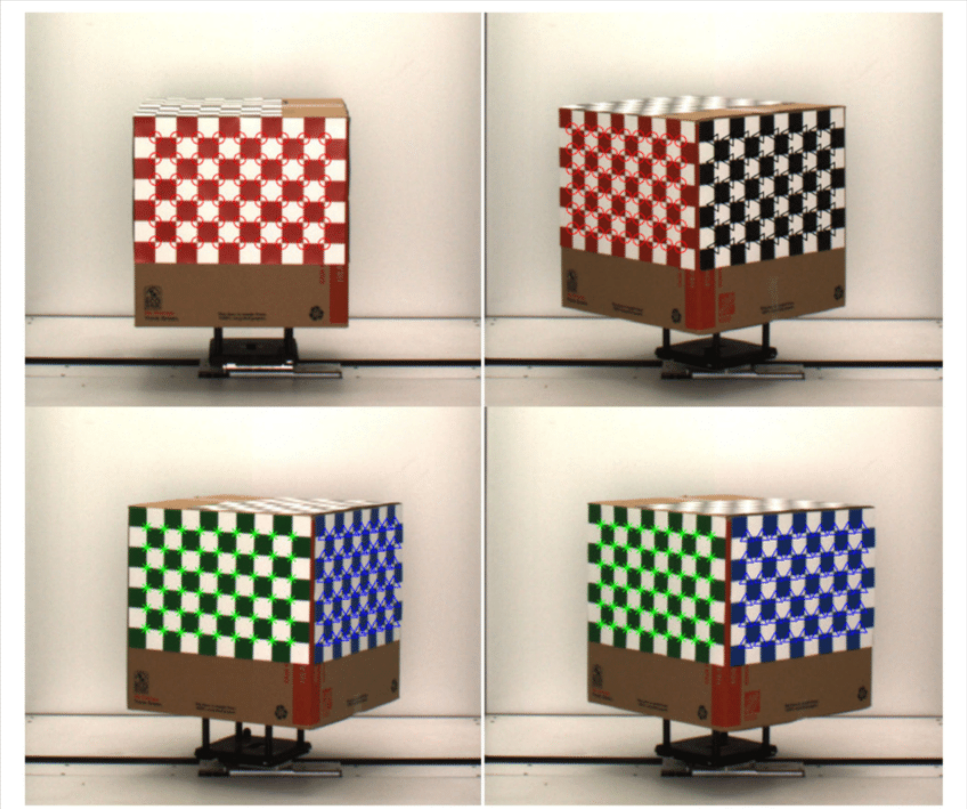
- fotogrammetrická kalibrace [62]
- auto-kalibrace [75]
- využití úběžníků pro ortogonální směry [44]
- kalibrace z čisté rotace [53]

4.1 Fotogrammetrická kalibrace

Ve fotogrammetrické kalibraci využíváme například třírozměrnou referenční objektovou kalibraci. Jedná se o způsob kalibrace kamery, která pozoruje kalibrační objekty, jehož geometrie ve 3D prostoru je známá s velmi dobrou přesností. Tento druh kalibrace je velmi efektivní. Kalibrační objekt se obvykle skládá ze dvou nebo tří k sobě kolmých rovin, příklad takového objektu můžeme vidět na obrázku 4.1 [76].

4.2 Auto-kalibrace

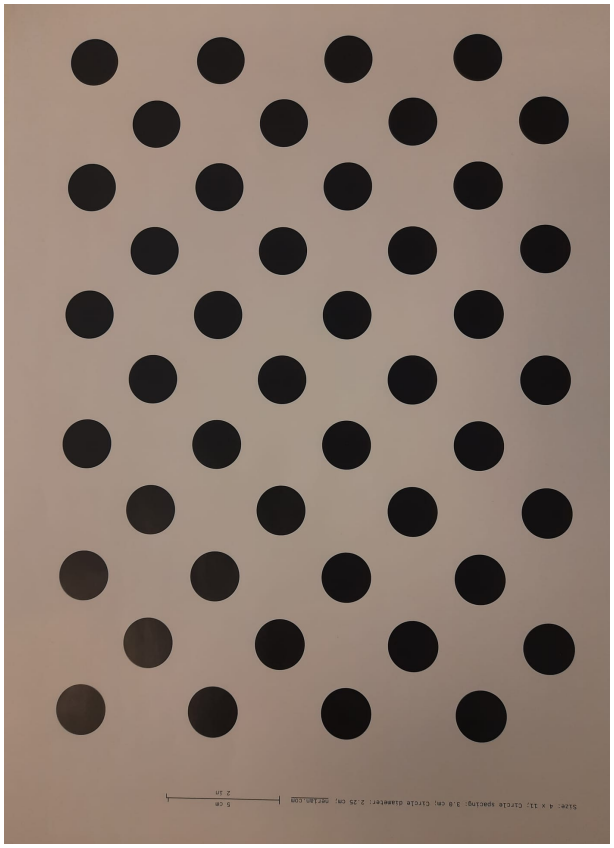
Auto-kalibrace nepoužívají žádné kalibrační objekty. Pouhým pohybem kamery ve statické scéně poskytuje informaci o scéně. Obecně máme dvě omezení - vnitřní a vnější parametry. Pokud jsou tedy snímky pořízeny stejnou kamerou s pevnými vnitřními parametry, korespondence mezi třemi snímky stačí k odhadnutí vnitřních i vnějších parametrů, které nám umožňují rekonstruovat 3D strukturu a podobnosti. I když je tento přístup velmi flexibilní, ještě není dostatečně přesný. Protože existuje mnoho parametrů k odhadu, nemůžeme vždy získat spolehlivé výsledky [76].



Obrázek 4.1: Kalibrační krychle [31]

4.3 Kombinace fotogrammetrické a auto-kalibrace

Další přístup využívá informace 2D metriky, tento přístup bychom zařadili mezi fotogrammetrickou kalibraci, která používá explicitní 3D model, a auto-kalibrací, která využívá pohybovou scénu, nebo ekvivalentně implicitní 3D informace. Navrhovaná technika vyžaduje pouze, aby kamera pozorovala rovinný vzor zobrazený v několika (alespoň dvou) různých směrech. Vzor lze vytisknout na laserové tiskárně a nalepit k rovnému povrchu (např. pevný knižní obal). Využívá tedy 2D plochu s geometrickými tvary, jedním z mnoha vzorů můžeme vidět 4.2. Ručně pochybujeme kamerou nebo rovinným vzorem. Pohyb nemusí být znám. V porovnání s klasickými technikami je navrhovaná technika značně flexibilnější: Každý si může vytvořit kalibrační obrazec sám a nastavení je velmi snadné. Ve srovnání s auto-kalibrací získává značnou míru robustnosti [76].



Obrázek 4.2: Kalibrační deska s kruhovým vzorem

4.3.1 Základní rovnosti a definice

Bod ve 2D rovině označíme:

$$\boldsymbol{m} = [u, v]^\top, \tag{4.1}$$

bod ve 3D rovině označíme:

$$\boldsymbol{M} = [X, Y, Z]^\top, \tag{4.2}$$

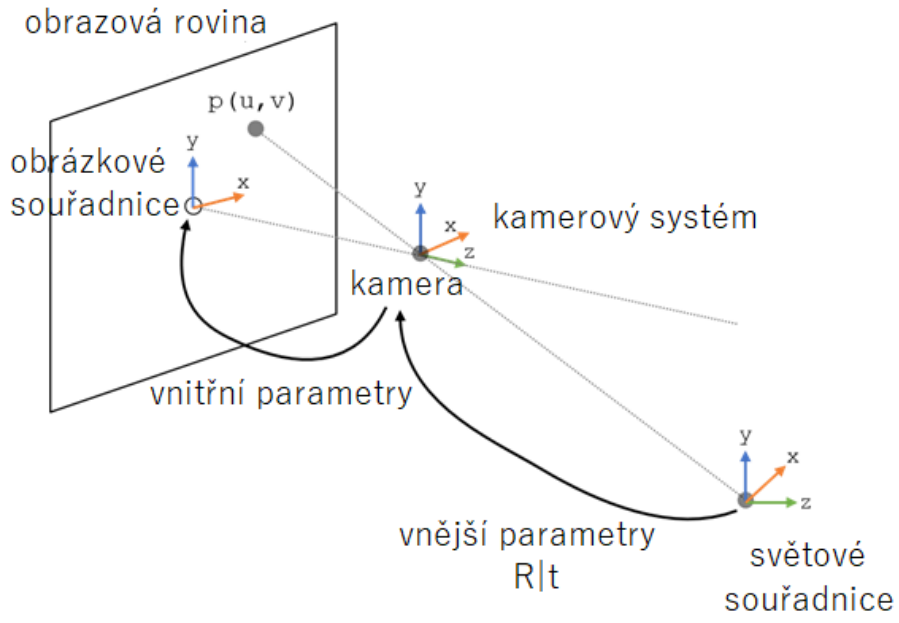
dále homogenní souřadnice:

$$\widetilde{\boldsymbol{m}} = [u, v, 1]^\top, \qquad \widetilde{\boldsymbol{M}} = [X, Y, Z, 1]^\top. \tag{4.3}$$

Vztah mezi 3D body v $\widetilde{\boldsymbol{M}}$ a jeho obrázkové projekce v $\widetilde{\boldsymbol{M}}$ je dána vztahem:

$$s\widetilde{\boldsymbol{m}} = \mathbf{A}[\mathbf{R}, \boldsymbol{t}]\widetilde{\boldsymbol{M}}, \tag{4.4}$$

kde $\mathbf{A} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$, s je libovolný měřítkový faktor, $(\mathbf{R}, \boldsymbol{t})$ nazýváme vnější parametry. Vnější parametry jsou to rotace a posunutí (translace), které uvádí do vztahu reálný souřadnicový systém a kamerový souřadnicový systém, $\mathbf{A}$ je vnitřní matice parametrů. $(u_0, v_0) = (W/2, H/2)$ a W, H jsou šířka a výška daného kalibračního obrázku, α a $\beta = k\alpha$ jsou ohniskové vzdálenosti a k konstanta pro zachování proporcí a poměru stran z 3D do 2D, γ je parametr popisující zakřivení (zkreslení) dvou os obrázku. Tato rovnice nám popisuje transformaci na obrázku 4.3 [76], [20].



Obrázek 4.3: Vztah mezi 3D světem a obrázkem [2]

4.3.2 Vztah mezi rovinným modelem a jeho obrázkem

Projektivní transformace je invertibilní transformace mezi dvěma projektivními perspektivami, kde zásadní vlastností je mapování přímek na přímky. Invertibilita platí pouze v případech, kdy vstupní body neleží na jedné přímce. Pokud body leží na jedné přímce, projektivní transformace není invertibilní. Dalšími názvy pro projektivní transformaci, se kterými je možné se v odborné literatuře setkat, jsou projektivita a homografie. Projektivita vyjadřuje, jak se mění vjem pozorovaného předmětu. Příkladem projektivní transformace je středové promítání se středem [8].

Bez újmy na obecnosti budeme předpokládat, že rovinný model má souřadnici $Z = 0$ v reálném světě. Označme i-tý sloupec rotační matice $\mathbf{R}$ jako r_i. Potom z rovnice (4.4) vychází [76]:

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{A}[\boldsymbol{r_1}, \boldsymbol{r_2}, \boldsymbol{r_3}, \boldsymbol{t}]\begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \mathbf{A}[\boldsymbol{r_1}, \boldsymbol{r_2}, \boldsymbol{t}]\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}, \tag{4.5}$$

kde levá strana rovnice jsou souřadnice v obrázku, matice $\mathbf{A}$ udává vnitřní parametry (optický střed, fokální délka, koeficient zkreslení), $\mathbf{R}$ udává vnější parametry (otočení a translace kamery) a poslední vektor udává souřadnice reálného prostoru. Dále můžeme psát, že $\boldsymbol{M} = [X, Y]^\intercal$ a $\widetilde{\boldsymbol{M}} = [X, Y, 1]^\intercal$. Modelový bod $\boldsymbol{M}$ a jeho obraz m souvisí s homografií $\mathbf{H}$:

$$s\widetilde{\boldsymbol{m}} = \mathbf{H}[\mathbf{R}, \boldsymbol{t}]\widetilde{\boldsymbol{M}} \qquad \text{, kde } \mathbf{H} = \mathbf{A}[\boldsymbol{r_1}, \boldsymbol{r_2}, \boldsymbol{t}], \tag{4.6}$$

$\mathbf{H}$ je matice 3x3 a je definována až do měřítka [76], [9].

Omezení vnitřních parametrů

Na základě obrázku modelové roviny lze odhadnout homografii. Tuto homografii označme jako $\mathbf{H} = [\boldsymbol{h_1}, \boldsymbol{h_2}, \boldsymbol{h_3}]$. Z rovnosti (4.6) dostaneme:

$$[\boldsymbol{h_1}, \boldsymbol{h_2}, \boldsymbol{h_3}] = \lambda \mathbf{A}[\boldsymbol{r_1}, \boldsymbol{r_2}, \boldsymbol{t}], \tag{4.7}$$

kde λ je libovolný skalár. Víme, že $\boldsymbol{r_1}$ a $\boldsymbol{r_2}$ jsou ortonormální (ON) dostaneme rovnosti:

$$\boldsymbol{h_1}^\intercal \mathbf{A}^{-\intercal} \mathbf{A}^{-1} \boldsymbol{h_2} = \mathbf{0} \tag{4.8}$$

$$\boldsymbol{h_1}^\intercal \mathbf{A}^{-\intercal} \mathbf{A}^{-1} \boldsymbol{h_1} = \boldsymbol{h_2}^\intercal \mathbf{A}^{-\intercal} \mathbf{A}^{-1} \boldsymbol{h_2} \tag{4.9}$$

Toto jsou dvě základní omezení vnitřních parametrů, dávající jednu homografii. Protože homografie má 8 stupňů volnosti a existuje šest vnějších parametrů (tři pro rotaci a tři pro translaci), můžeme získat pouze dvě omezení vnitřních parametrů [76].

4.3.3 Řešení kalibrace kamery

Tato část poskytuje podrobnosti, jak efektivně vyřešit problém s kalibrací kamery. Začneme analytickým řešením, po kterém následuje nelineární optimalizační technika založená na kritériu maximální věrohodnosti. Nakonec vezmeme v úvahu zkreslení čočky, což poskytuje analytická i nelineární řešení [76].

Řešení v uzavřené formě

Označme si symetrickou matici $\boldsymbol{B}$:

$$\mathbf{B} = \mathbf{A}^{-\intercal}\mathbf{A}^{-1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} = \begin{bmatrix} \frac{1}{\alpha^2} & -\frac{\gamma}{\alpha^2\beta} & \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} \\ -\frac{\gamma}{\alpha^2\beta} & \frac{\gamma^2}{\alpha^2\beta^2} + \frac{1}{\beta^2} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} \\ \frac{v_0\gamma - u_0\beta}{\alpha^2\beta} & -\frac{\gamma(v_0\gamma - u_0\beta)}{\alpha^2\beta^2} - \frac{v_0}{\beta^2} & \frac{(v_0\gamma - u_0\beta)^2}{\alpha^2\beta^2} + \frac{v_0^2}{\beta^2} + 1 \end{bmatrix}, \tag{4.10}$$

a $\boldsymbol{b} = [B_{11}, B_{12}, B_{13}, B_{22}, B_{23}, B_{33}]^\intercal$ je šestidimenzionální vektor. I-tý sloupec matice $\mathbf{H}$ označíme jako $\boldsymbol{h_i} = [h_{i1}, h_{i2}, h_{i3}]$ z těchto definic dostáváme:

$$\boldsymbol{h_i}^\intercal \mathbf{B} \boldsymbol{h_j} = \boldsymbol{v_{ij}}^\intercal \boldsymbol{b}, \tag{4.11}$$

kde $\boldsymbol{v_{ij}} = [h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}]^\intercal$. Z omezovacích rovnice (4.8) a (4.9) a dané homografie $\boldsymbol{H}$ můžeme napsat dvě homogenní rovnice pro $\boldsymbol{b}$:

$$\begin{bmatrix} \boldsymbol{v_{12}}^\intercal \\ (\boldsymbol{v_{11}} - \boldsymbol{v_{12}})^\intercal \end{bmatrix} \boldsymbol{b} = \mathbf{0} \tag{4.12}$$

Máme-li napozorováno n obrázků modelové roviny, skládáním n takových rovnic jako je (4.12) dostaneme:

$$\mathbf{V}\boldsymbol{b} = \mathbf{0}, \tag{4.13}$$

kde $\mathbf{V}$ je matice o rozměrech $2n$x6. Je-li $n \geq 3$ budeme mít obecně jednoznačné řešení pro $\boldsymbol{b}$ definované až do měřítka. Je-li $n = 2$ můžeme zavést omezení distorze (zkroucení) $\gamma = 0$, například $[0, 1, 0, 0, 0, 0]\boldsymbol{b} = 0$, což přidává podmínku navíc pro řešení rovnice (4.13). Je-li $n = 1$, můžeme řešit pouze dva vnitřní parametry kamery, např. α a β, za předpokladu, že jsou známé body u_0 a v_0 (např.

ve středu obrazu) a $\gamma = 0$. Rovnice (4.13) je známé jako vlastní vektor matice $\mathbf{V}^\intercal\mathbf{V}$ pro nejmenší vlastní vlastní číslo (ekvivalentně pravý singulární vektor $\mathbf{V}$ spojený s nejmenší singulární hodnotou). Je-li $\boldsymbol{b}$ odhadnut, můžeme vypočítat všechny vnitřní parametry kamery následovně. Matice $\mathbf{B} = \lambda\mathbf{A}^{-\intercal}\mathbf{A}$ s λ libovolným měřítkem. Bez problémů můžeme jednoznačně extrahovat vnitřní parametry z matice $\mathbf{B}$ [76].

$$\begin{aligned} v_0 &= (B_{12}B_{13} - B_{11}B_{23})/(B_{11}B_{22} - B_{12}^2) \\ \lambda &= B_{33} - [B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})]/B_{11} \\ \alpha &= \sqrt{\lambda/B_{11}} \\ \beta &= \sqrt{\lambda B_{11}/(B_{11}B_{22} - B_{12}^2)} \\ \gamma &= -B_{12}\alpha^2\beta/\lambda \\ u_0 &= \gamma v_0/\alpha - B_{13}\alpha^2/\lambda \end{aligned} \tag{4.14}$$

Je-li $\mathbf{A}$ známé, vnější parametry pro každý obrázek jsou jednoduše vypočítány. Z rovnice (4.6) dostaneme:

$$\begin{aligned} \boldsymbol{r}_1 &= \lambda\mathbf{A}^{-1}\boldsymbol{h_1} \\ \boldsymbol{r}_2 &= \lambda\mathbf{A}^{-1}\boldsymbol{h_2} \\ \boldsymbol{r}_1 &= \boldsymbol{r}_1 \times \boldsymbol{r}_2 \\ \boldsymbol{t} &= \lambda\mathbf{A}^{-1}\boldsymbol{h_3} \end{aligned} , \tag{4.15}$$

kde $\lambda = 1/\|\mathbf{A}^{-1}\boldsymbol{h_1}\| = 1/\|\mathbf{A}^{-1}\boldsymbol{h_2}\|$. Kvůli šumu v datech, takto vypočítaná matice $\mathbf{R} = [\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3]$ obecně nesplňuje vlastnosti rotační matice. Nejlepší rotační matici pak lze získat například rozkladem singulární hodnoty více informací nalezneme v článku [42]; [76].

Řešení pomocí maximálních věrohodnostních odhadů (MLE)

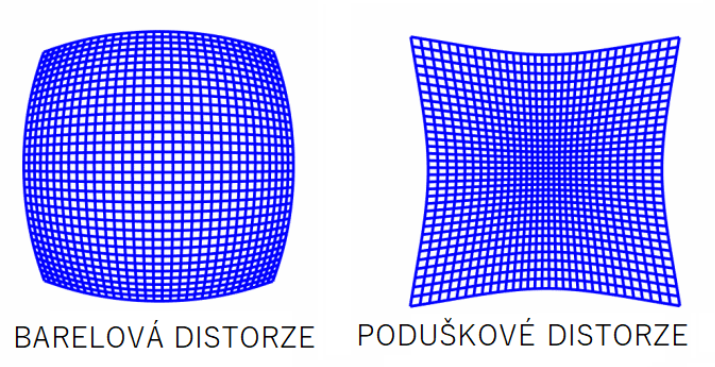
Výše uvedené řešení je získáno minimalizací algebraické vzdálenosti, která není fyzikálně smysluplná. Můžeme to upřesnit pomocí odvození s odhadem maximální věrohodnosti. Nechť je dáno n obrázků modelové roviny a na modelové rovině jsou body. Předpokládejme, že obrazové body jsou poškozeny nezávislým a identicky distribuovaným šumem. Odhad maximální věrohodnosti lze získat minimalizací následujících funkcí:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} \|m_{ij} - \hat{\boldsymbol{m}}(\mathbf{A}, \mathbf{R}_i, \boldsymbol{t}_i, \boldsymbol{M}_j)\|^2, \tag{4.16}$$

kde $\hat{m}(\mathbf{A}, \mathbf{R}_i, \boldsymbol{t}_i, \boldsymbol{M}_j)$ je projekce bodu $\boldsymbol{M}_j$ do obrázku i. Rotace $\mathbf{R}$ je parametrizována vektorem tří parametrů, označeným $\mathbf{r}$, který je rovnoběžný s osou rotace a jehož velikost je rovna úhlu rotace. $\mathbf{R}$ a $\mathbf{r}$ jsou příbuzné podle Rodriguesova vzorce, podrobnosti k tomuto tématu můžeme najít v [37]. Minimalizace rovnice (4.16) je nelineární problém minimalizace, který je řešen pomocí Levenberg-Marquardtova algoritmu implementovaného v Minpacku, který je blíže popsán v článku [58]. Vyžaduje počáteční odhad $\mathbf{A}$, $\mathbf{R}_i, t_i | i = 1...n$, které lze získat pomocí techniky popsané v předchozí části [76].

4.3.4 Radiální distorze

Existují dva hlavní druhy zkreslení (resp. distorze) - radiální zkreslení a decentrované zkreslení [69]. Typy radiálního zkreslení můžeme vidět na obrázku 4.4. Dosud jsme neuvažovali zkreslení objektivu fotoaparátu. Fotoaparát však obvykle vykazuje značné zkreslení čočky, zejména radiální složky. Projevuje se tak, že se rovné čáry zdají křivé. Pro propracovanější modely se odkazujeme na [48]. Pro zjednodušení odvozování, budeme pouze uvažovat první dva koeficienty radiální distorze.



Obrázek 4.4: Typy radiální distorze [22]

Nechť (u, v) jsou ideální (bez zkreslení) souřadnice obrazu pixelu a $(\widetilde{u}, \widetilde{v})$ odpovídají skutečné souřadnice pozorovaného obrazu. Ideálními body jsou projekce bodů podle dírkového modelu (angl. *pinhole model*). Podobně (x, y) a $(\widetilde{x}, \widetilde{y})$ jsou ideální (bez zkreslení) a skutečné (zkreslené) normalizované souřadnice obrazu. Dostaneme [76], [69]:

$$\begin{aligned} \widetilde{x} &= x + x[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \\ \widetilde{y} &= y + y[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2], \end{aligned} \tag{4.17}$$

kde k_1 a k_2 jsou koeficienty radiálního zkreslení. Střed radiálního zkreslení je stejný jako bod ve středu obrázku (u_0, v_0). Z $(\widetilde{u} = u_0 + \alpha\widetilde{x}, \widetilde{v} = v_0 + \beta\widetilde{y})$ a s předpokladem $\gamma = 0$:

$$\begin{aligned} \widetilde{u} &= u + (u - u_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \\ \widetilde{v} &= v + (v - v_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \end{aligned} . \tag{4.18}$$

Radiální zkreslení lze popsat pro 3 členy a je popsáno následujícími rovnicemi:

$$\begin{aligned} \widetilde{x} &= x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \\ \widetilde{y} &= y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6), \end{aligned} \tag{4.19}$$

kde k_1, k_2 a k_3 jsou koeficienty radiální distorze, $r = \sqrt{x^2 + y^2}$. Podobně dochází k decentrované zkreslení, protože čočka snímající obraz není vyrovnána dokonale rovnoběžně se zobrazovací rovinou. Některé oblasti na obrázku tedy mohou vypadat blíže, než se očekávalo. Velikost decentrovaného zkreslení lze znázornit následovně:

$$\begin{aligned} \widetilde{x} &= x + [2p_1xy + p_2(r^2 + 2x^2)] \\ \widetilde{y} &= y + [2p_2xy + p_1(r^2 + 2y^2)], \end{aligned} \tag{4.20}$$

kde p_1 a p_2 jsou koeficienty decentrované distorze. Dohromady dostaneme koeficient distorze $(k_1, k_2, p_1, p_2, k_3)$ [2], [76], [69].

Odhad radiálního zkreslení

Pro odhad radiálních koeficientů použijeme maximální věrohodnostní odhady. Jednou ze strategií je odhadnout k_1 a k_2 po odhadu ostatních parametrů, které nám poskytnou ideální pixelové souřadnice (u, v). Potom z rovnic (4.18) máme dvě rovnice pro každý bod v každém obrázku:

$$\begin{bmatrix} (u-u_0)(x^2+y^2) & (u-u_0)(x^2+y^2)^2 \\ (v-v_0)(x^2+y^2) & (v-v_0)(x^2+y^2)^2 \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} \widetilde{u}-u \\ \widetilde{v}-v \end{bmatrix}. \tag{4.21}$$

Je-li dáno m bodů v n obrázcích, můžeme skládat všechny rovnice dohromady, abychom získali celkem $2mn$ rovnic, nebo v maticové formě jako $\mathbf{D}\boldsymbol{k} = \mathbf{d}$, kde $\boldsymbol{k} = [k_1; k_2]^\intercal$. Lineární řešení nejmenších čtverců je dáno vztahem:

$$\boldsymbol{k} = (\mathbf{D}^\intercal \mathbf{D}^{-1})\mathbf{D}^\intercal \boldsymbol{d}. \tag{4.22}$$

Jakmile jsou k_1 a k_2 odhadnuty, je možné zpřesnit odhad ostatních parametrů řešením (4.16) s $\hat{\boldsymbol{m}}(\mathbf{A}, \mathbf{R}_i, \boldsymbol{t}_i, M_j)$ nahrazením (4.18). Tyto dvě procedury provádíme dokud nedokonverguje [76].

Úplný maximální věrohodnostní odhad

Experimentálně jsme zjistili, že konvergence výše uvedené techniky je pomalá. Přirozeným rozšířením z rovnice (4.16) je pak odhadnout kompletní sadu parametrů minimalizací následujících funkcí:

$$\sum_{i=1}^{n} \sum_{j=1}^{m} \|m_{ij} - \widetilde{\boldsymbol{m}}(\mathbf{A}, k_1, k_2, \mathbf{R}_i, \boldsymbol{t}_i, \boldsymbol{M}_j)\|^2, \tag{4.23}$$

kde $\widetilde{\boldsymbol{m}}(\mathbf{A}, k_1, k_2, \mathbf{R}_i, \boldsymbol{t}_i, \boldsymbol{M}_j$ je průmět bodu M_j v obraze i podle rovnice (4.4), následovaný zkreslením podle rovnic (4.18). Toto je problém nelineární minimalizace, který je řešen pomocí Levenberg-Marquardtova algoritmu implementovaného v Minpacku, již zmíněna předtím. Rotace je opět parametrizována 3-vektorem $\boldsymbol{r}$. Počáteční odhad $\mathbf{A}$ a $\mathbf{R}_i, \boldsymbol{t}_i | i = 1...n$ lze získat pomocí techniky popsané řešení pomocí MLE. Počáteční odhad k_1 a k_2 lze získat technikou popsanou v posledním odstavci nebo jednoduše jejich nastavením na 0 [76].

Kapitola 5

Praktická část

Praktická část této práce je zaměřena na detekci chodců, predikci jejich pohlaví, odhad pozice chodcovi hlavy a určení zda se člověk dívá na určitý objekt (v našem případě se jedná o reklamu na citylightu). Pro detekci chodců jsme si vybrali jeden z nejnovějších state-of-the-art detektorů, kterým je YOLOv7. Jedná se o detektor jednofázový, který je vhodný pro detekci v reálném čase. Vyzkoušeli jsme také jiný detektor chodců Pedestron [45], který má i různé backbony, mezi niž patří například VGG [65], ResNeXt [73], HRNet [70].

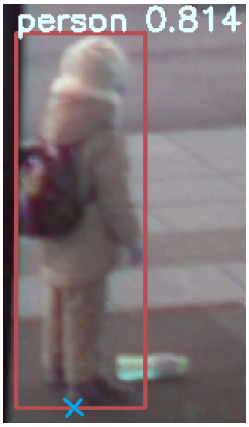
Provedli jsme kalibraci kamery, kterou budeme používat pro celou praktickou část. YOLOv7 jsme dotrénovali pro predikci pohlaví za použití veřejného a vlastního datasetu. Kvůli nepřesným výsledkům dotrénování, jsme nadále YOLOv7 používali pouze jako detektor chodce. Pro predikci pohlaví jsme našli alternativu ve formě umělé neuronové sítě od ONNX. Pro odhad pozice hlavy jsme zkusili i state-of-the-art algoritmus, který se nazývá img2pose.

Vytvořili jsme funkci, která nám říká, zda se člověk dívá, nebo nedívá. Tato funkce má na vstupu odhad pozice hlavy a souřadnice člověka v reálném světě.

Pro odhadnutí, kde se člověk nachází v reálném světě jsme zkusili 2 přístupy - pomocí projekční matice a pomocí perspektivní transformace. Nad všemi detekcemi nakonec spustíme sledování chodců. Nakonec jsme vytvořili vlastní vizualizační funkci.

Všechny části detektorů a predikcí jsme spojili a vytvořili algoritmus, který nám říká, jestli se člověk dívá, nebo nedívá na zkoumaný objekt.

Začínáme detekcí člověka na obrázku pomocí YOLOv7. Dále převádíme polohu nohou z obrázku do 3D prostoru pomocí souřadnic, kde pozice nohou definujeme jako na obrázku 5.1. Následuje predikce odhadu pozice hlavy pomocí img2pose a pohlaví pomocí ONNX sítě [10]. Vytvořili jsme funkci pro klasifikaci dívá/nedívá se. Nad výstupy detektoru jsme spustili sledování chodce. Výstupem tedy bude textový soubor s názvem obrázku, souřadnice ohraničujícího obrázku, zda se člověk dívá, nebo nedívá a nakonec i pohlaví.

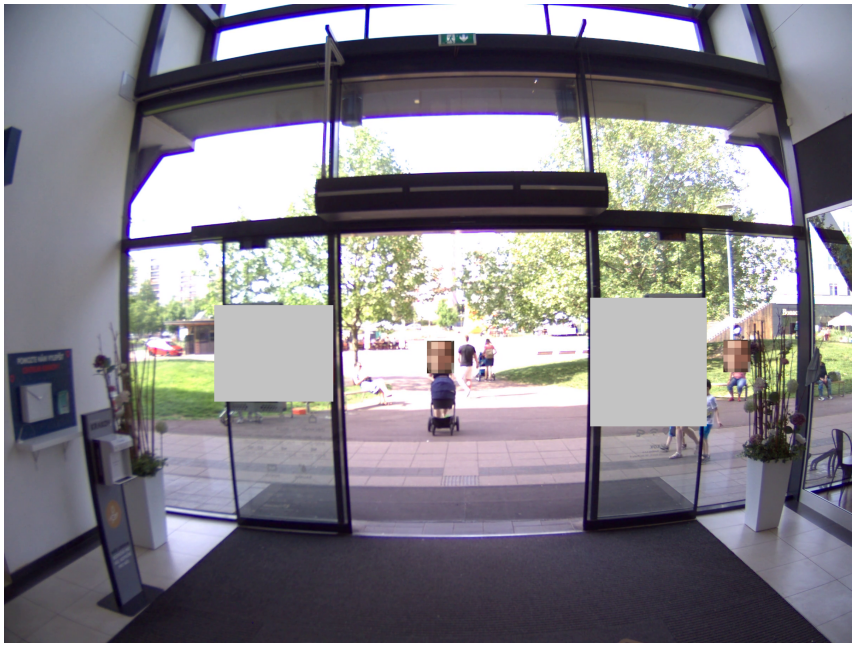


Obrázek 5.1: Predikce chodce a definice polohy nohou označeno křížkem

5.1 Příprava dat

K dispozici jsme měli videa ze zamýšlené aplikace, ukázka snímků je na obrázku 5.2 a na 5.3. Snímky jsou snímané z kamery, která se nachází nad reklamou. Toto video mělo snímkovou frekvenci 25 FPS o rozlišení 2400 x 1808 pixelů. Z našeho videa jsme si zvolili 100 náhodných snímků, které jsme poté anotovali pomocí ohraničujícího rámečku a dodali jsme informaci o pohlaví. Pro dotrénování YOLOv7 k predikci pohlaví jsme využili veřejný dataset PETA, zmíněný již ve 2. kapitole, a náš vlastní dataset 5.2. V datasetu pro určení zda se člověk dívá, nebo nedívá byla komplikace v tom, že to nešlo u některých lidí přesně říci, jestli se dívají.

Obrázek 5.2: Snímek ze zamýšlené aplikace



Obrázek 5.3: Snímek ze zamýšlené aplikace

5.2 Kalibrace kamery

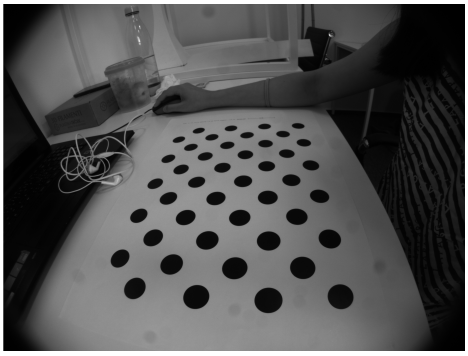
K pořízení záběrů jsme používali kameru od společnosti Basler model daA2500-14um. Snímky z této kamery můžeme vidět na obrázcích 5.4. Můžeme vidět, že kamera má barelovou distorzi. Pro naše budoucí účely jsme si nakalibrovali kameru a zjistili si jednotlivé parametry pomocí programu v pythonu a knihovnou OpenCV. Získali jsme vnitřní parametry (ozn. ve 4. kapitole jako **A**), nazýváno také jako matice kamery:

$$M_{cam} = \begin{bmatrix} 1528.52 & 0.0 & 1296 \\ 0.0 & 1433.13 & 972 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \tag{5.1}$$

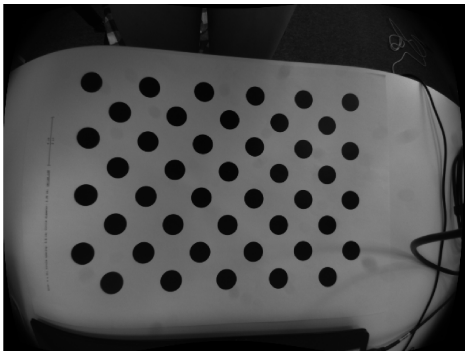
a koeficient zkreslení čočky, nebo-li distorze:

$$C_{dis} = [-0.28, 0.06, 0.002, 0.00045, 0.013] \tag{5.2}$$

S využitím odhadnutých parametrů jsme provedli kalibraci kamery. Výsledek kalibrace můžete vidět na obrázku 5.5.

Obrázek 5.4: Snímky z kamery pro kalibraci

Obrázek 5.5: Porovnání původního obrázku z kamery a narovnaného obrázku pomocí získaných parametrů

5.3 Detekce a sledování člověka

Jedním z cílů práce bylo zkusit dotrénovat klasifikaci pohlaví na detektoru chodce. Při detekci člověka jsme vyzkoušeli 2 detektory YOLOv7 a Pedestron. Při použití detektoru Pedestron trvala detekce jednoho snímku z videa 3 sekundy, zatímco pro detekci pomocí YOLOv7 detekce trvala pouze 15−20 ms. YOLOv7 je navíc mnohem robustnější vůči prostředí a světelnému prostředí. Nakonec jsme se rozhodli pro detekci člověka využít jednofázový detektor YOLOv7 popsaný ve 2. kapitole, který má vysokou přesnost a je rychlý. Tento detektor funguje dobře v naší aplikaci.

Tento vybraný detektor jsme zkusili dotrénovat na predikci pohlaví, což jsme popsali v podkapitole 5.3.2. Výsledky dotrénování nás vedly k tomu, že tento detektor nebudeme používat k predikci a detekci pohlaví z celého těla, ale pouze k detekci chodců. YOLOv7 predikuje ohraničující rámeček detekovaného člověka. Dále budeme chtít detekovat přibližně jeho nohy pro pozdější využití. Souřadnice nohou budeme brát jako střed spodní hrany ohraničujícího rámečku. Pro sledování chodců jsme využili již zmíněný algoritmus IoU tracker.

5.3.1 Kvantizace

Cílem bylo také kvantizovat část algoritmu a proto jsme se rozhodli kvantizovat neuronovou síť YOLOv7. Kvantizování jsme provedli pomocí knihovny ONNX RUNTIME [15]. Každá váha v neuronové síti YOLOv7 je reprezentována jako číslo s desetinnou čárkou, které má 32 bitů. Při převodu modelu YOLOv7 na ONNX se pevná desetinná čárka obvykle kvantizuje na 16-bitovou reprezentaci s jedním znaménkovým bitem a 15 bity pro celou a desetinnou část. Tato reprezentace byla zvolena pro efektivnější využití paměti a výpočetních zdrojů při zachování dostatečné přesnosti pro detekci objektů. Pevná desetinná čárka může být výhodná pro nasazení modelů na zařízeních s omezenými výpočetními zdroji, ale zároveň může mít vliv na přesnost modelu. ONNX model je přibližně 1.2 krát rychlejší než YOLOv7, více v článku [11].

5.3.2 Predikce pohlaví

Pro YOLOv7 jsme se pokusili o dotrénování sítě, aby predikovala pohlaví z celého těla. Pro tyto účely jsme využili veřejně dostupný dataset PETA a náš vlastní anotovaný dataset - celkově jsme měli 5500 obrázků. PETA dataset má nevýhodu v tom, že jednotlivé snímky obsahují jen jednoho člověka. Tento celkový dataset byl rozdělen na 700 testovacích obrázků, 3600 trénovacích a validačních 1300. Při trénování jsme byli omezeni grafickou kartou a velikostí její paměti. Predikce jednotlivých pohlaví (muž modrý ohraničující rámeček, žena oranžový ohraničující obrázek) můžeme vidět na obrázku 5.7 a její chybovou matici na obrázku 5.6 (FN, FP, TN, TP zmíněno již v 2. kapitole). Tato matice nám říká jak si síť vedla na testovací sadě: 48 procent mužů bylo predikováno správně jako muži, 0.35 procent můžu bylo predikováno jako ženy a 0.17 mužů bylo predikovaných jako pozadí. Přesnost této dotrénované sítě je pro muže 0.36 a pro ženy 0.15. Detekce pro snímek trvala 4 sekundy.

	GT muž	žena	pozadí
PREDIKOVANÉ muž	0.48	0.25	0.57
žena	0.35	0.14	0.43
pozadí	0.17	0.60	

Obrázek 5.6: Chybová matice

Jelikož se nám nepodařilo dostatečně natrénovat neuronovou síť nalezli jsme alternativu, kterou se stala předtrénovaná síť od ONNX s páteří GoogleNet, více na GitHubu [10].

Obrázek 5.7: Vlevo máme správně anotované obrázky a vpravo predikci dotrénovanou sítí YOLOv7 na datasetu PETA

5.4 Mapování z 2D obrázku do 3D světa

Pro funkci zda se člověk dívá, nebo nedívá na reklamu, budeme chtít vědět, kde se člověk nachází v reálném světě. Využijeme znalost toho, kde člověk má nohy v obrázku z YOLOv7 a pomocí projekčních rovnic kamer a nebo pomocí perspektivní transformace se pokusíme určit, kde se člověk nachází v reálné světě.

5.4.1 Pomocí projekční matice

Projekční matice se používá k převodu z 3D souřadnic světa do souřadnic 2D obrazu a to pomocí rovnic 5.3. K odhadu projekční matice 3x4 je vypočítaná jako součin vnitřních a vnějších vlastností obrazu.

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \tag{5.3}$$

kde $[u, v]$ jsou souřadnice bodu v obrázku, s je škálování, f_x, f_y, c_x, c_y jsou vnitřní vlastnosti obrazu, r_{ij} pro $i, j = 1, 2, 3$ jsou prvky rotační matice a t_i pro $i = 1, 2, 3$ jsou prvky translačního vektoru, $[X, Y, Z, 1]$ jsou souřadnice v reálném světě.

Zkusíme promítnout body z reálného světa do obrázku 5.8. Z obrázku vidíme, že tato projekce je celkem přesná. Proto se pokusíme vytvořit inverzi, která by převáděla souřadnice z 2D do 3D.

Pro naše účely se matice rotace zjednoduší, protože máme kameru bez rotace a to na tvar:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.4}$$

A navíc se zjednoduší i translační vektor:

$$\begin{bmatrix} 0.4 & 2.09 & 0 \end{bmatrix}, \tag{5.5}$$

kde 2.09 je výška kamery od podlahy. Budeme pouze předpovídat X, Z, protože lidi stojí na podlaze, což znamená, že $Y = 0$. Dostáváme rovnice:

$$\begin{aligned} Z &= (f_y t_2 + c f_y - b c_y)/(bv + c f_y - b c_y) \\ s &= Zb + t_3 \\ X &= (su - Zbc_y - t_1 f_y - c_x t_3)/(f_y a) \\ Y &= 0 \end{aligned} \quad ,$$

kde $a, b, c = 1$ jsou diagonální prvky rotační matice. S těmito vnitřními i vnějšími parametry máme jednoznačné řešení. Při této inverzní transformaci, jsme nedostali přesné výsledky a proto jsme vyzkoušeli i jiný přístup.

Obrázek 5.8: Projekce 3D bodů do obrázku pomocí projekčních rovnic

5.4.2 Pomocí perspektivní transformace

Transformace perspektivy je technika používaná v počítačovém vidění a zpracování obrazu k transformaci obrazu nebo videa z jedné perspektivy do druhé. Této transformace je dosaženo mapováním každého pixelu v původním obrázku do nového umístění ve výstupním obrázku. K této transformaci je nutné znát 4 body, aby bylo možné odhadnout matici transformace. Nejprve jsou tyto body normalizovány tak, aby byly ve stejném měřítku a aby jeden bod byl v počátku souřadnicového systému. Poté se použije metoda nejmenších čtverců pro nalezení matice, která nejlépe odpovídá těmto bodům. Výsledná transformační matice je použita k transformaci bodů v původním obrazu do cílového prostoru. Perspektivní transformace je definována:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & b_1 \\ a_3 & a_4 & b_2 \\ c_1 & c_2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \tag{5.6}$$

kde (x, y) jsou souřadnice v původním obrazu (vstup), (x', y') jsou souřadnice transformovaného bodu a a_i pro $i = 1, 2, 3, 4$ jsou parametry definující transformaci jako je škálování, rotaci a další, b_i pro $i = 1, 2$ je prvky translačního vektoru, c_i pro $i = 1, 2$ jsou prvky projekční vektoru. Pro afinní transformaci je projekční vektor roven $\mathbf{0}$. Afinní transformaci lze tedy považovat za konkrétní případ perspektivní transformace. My se pokusíme o to, transformovat podlahu tak, abychom z ní mohli zjistit souřadnice stojících lidí na ní. Zkusíme si tedy odhadnout matici a zadáme známé body rohů podlahy. Víme, že v realitě šířka vchodu je 7.2 metrů a viditelná délka podlahy je 3.6 metrů.

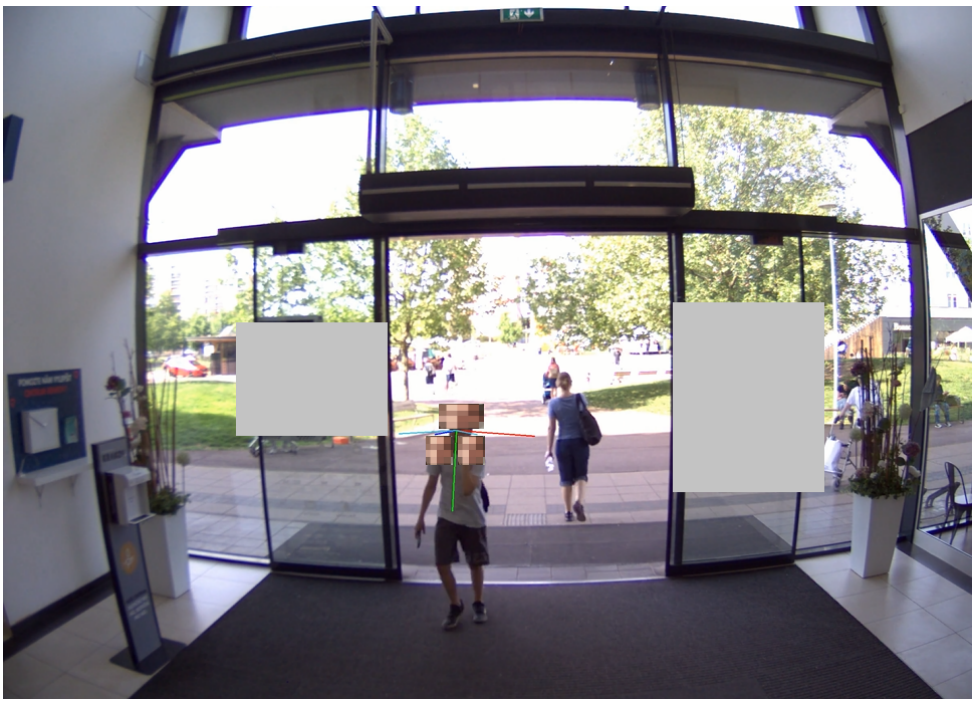
Z obrázku jsme se snažili odhadnout souřadnice 3 bodů, u kterých známe souřadnice v reálném světě. Přístupy jsme porovnali na těchto 3 známých bodech, výsledky jsou v tabulce 5.1.

Transformace	Reálné souřadnice (v metrech)	Predikované souřadnice (v metrech)
Projekční matice	[0, 5.6]	[0.56, 2.82]
Perspektivní transformace	[0, 5.6]	[-0.51, 6.38]
Projekční matice	[0, 3.8]	[0.8, 2.04]
Perspektivní transformace	[0, 3.8]	[-0.25, 3.58]
Projekční matice	[0.7, 5.6]	[1.32, 2.74]
Perspektivní transformace	[0.7, 5.6]	[1.13, 6.03]

Tabulka 5.1: Porovnání dvou transformací z 2D do 3D souřadnic

5.5 Odhad pozice hlavy

Pro odhad pozice hlavy jsme využili algoritmus img2pose, který jsme popsali v 3. kapitole. Výstupem tedy byl textový soubor, který nám dal vektor $[r_x, r_y, r_z, t_x, t_y, t_z]$. První tři čísla patří k rotačnímu vektoru a další tři jsou translační vektor. Vizualizace odhadu pohledu z našeho prostředí můžeme vidět na obrázku 5.9, kde předklon a záklon je modrá osa, otáčení je červená osa, úklon je zelená osa a navíc máme znázorněno světle modrou barvou FOV daného člověka.



Obrázek 5.9: Odhad pozice hlavy pomocí img2pose

5.6 Funkce pro predikce dívání se/nedívání se

Pro predikci zda se člověk dívá, nebo nedívá budeme využívat vlastní funkci. Tato funkce bere vstup úhly pozice hlavy z img2pose a pozici ve 3D světě z transformace 2D do 3D. Pomocí transformace z 2D do 3D dostaneme souřadnice (u, v). Převádíme tedy pozici nohou z obrázku do 3D světa, to můžeme, protože daná reklamní plocha je 70 cm nad zemí a je do výšky 200 cm. Představme si, že se díváme do prostoru ze shora, nákres tohoto pohledu je vyobrazeno na obrázku 5.10 se souřadnicovým systémem u, v.

Úhel α_1 dostaneme pomocí vztahu $90 + uhel_{uklonu}$, kde $uhel_{uklonu}$ je v intervalu $[-45, +45]$, pokud chodec stojí v kladné části osy u. V opačném případě $\alpha_1 = 90 - uhel_{uklonu}$.

Pomocí sinové věty, můžeme zjistit velikost úseček a_1 a a_2:

$$a_1 = a_3 \frac{sin(\alpha_1)}{sin(\alpha_3)} \tag{5.7}$$

$$a_2 = a_3 \frac{sin(\alpha_2)}{sin(\alpha_3)} \tag{5.8}$$

Pomocí těchto informací lze dopočítat bod p z nákresu a to pomocí vzdálenosti 2 bodů - p souřadnice označíme jako $(0, n)$ a souřadnice člověka označíme jako (x, y), p a bod $(0, v)$.

$$a_2^2 = (x - 0)^2 + (y - n)^2, \tag{5.9}$$

A pro druhou vzdálenost platí:

$$a_1^2 = (0 - 0)^2 + (y - n)^2. \tag{5.10}$$

Vidíme, že je třeba jen jedna z těchto rovnic pro výpočet souřadnic bodu p. Pro n nám vždy vyjdou 2 řešení, a my chceme, aby toto řešení bylo $< v$. Máme-li tedy souřadnice bodu p, můžeme vypočítat směrový vektor přímky z, což je přímka, která je definována body p a bodem člověka.

$$z = (u - 0, v - n). \tag{5.11}$$

Z tohoto dostaneme normálový vektor $\boldsymbol{N}$:

$$\boldsymbol{N} = (-(v - n), u - 0) \tag{5.12}$$

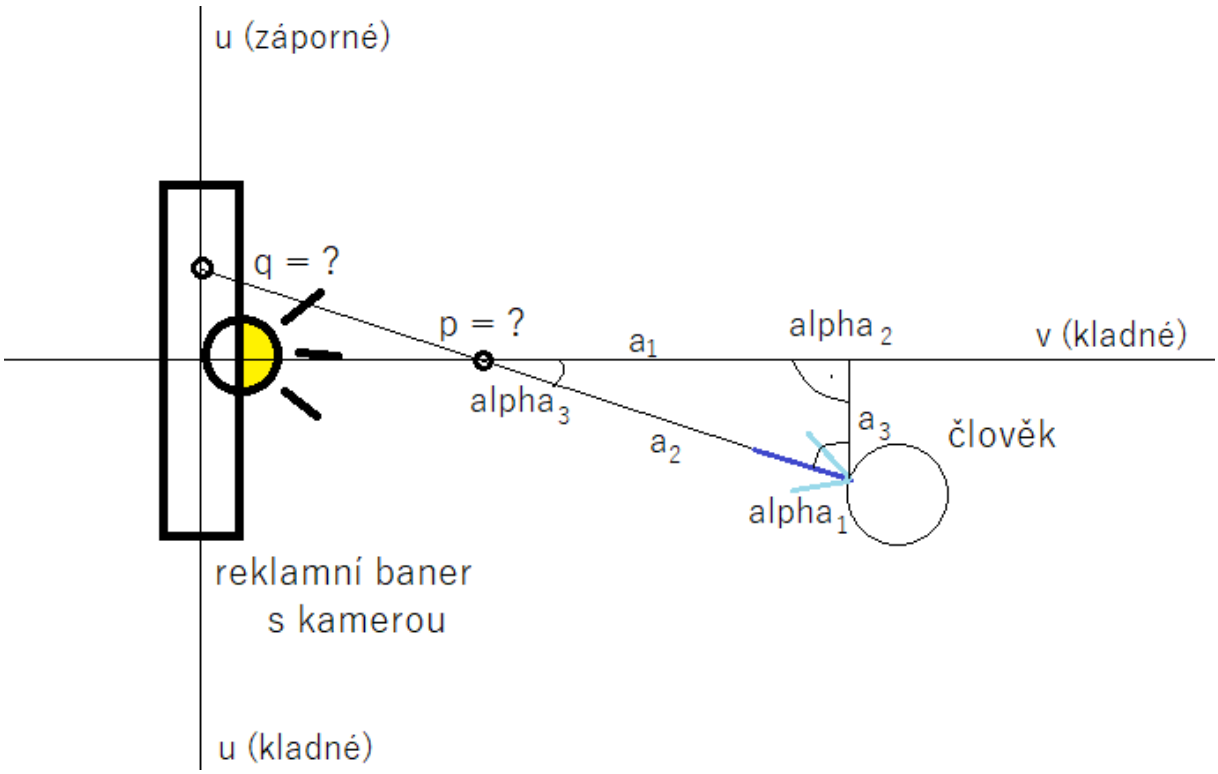
Dosazením jednoho z bodů p nebo bodu člověka vypočítáme konstantu c a dostaneme rovnici přímky:

$$-(v - n)x + uy + c = 0, \tag{5.13}$$

kde x, y představují souřadnicový systém. Nyní budeme hledat bod q se souřadnicemi $(0, k)$, který je průnik s osou u, takže platí $u = 0$:

$$k = -c/u \tag{5.14}$$

Poté se ptáme, zda bod q je v oblasti reklamního baneru. Obdobně to děláme pro zorné pole, kdy počítáme tento bod q s úhlem úklonu ±30 stupňů. Tímto dostaneme 2 body q, který nám tvoří interval označme si ho jako $[q_1, q_2]$ a ptáme se, zda se tento interval nepřekrývá s intervalem daného reklamního baneru. Pokud se stane, že jeden z úhlů pro zorné pole je > 90, tak nemůžeme vyhodnotit zda to člověk má v zorném poli.



Obrázek 5.10: Nákres pohledu ze shora

5.7 Vizualizace

Napsali jsme si funkce, které nám vše vizualizuje pro lepší orientování se v obrázcích. Na všech obrázcích vizualizuje odhad pozice hlavy, číslo tracku, pohlaví a věk, což je vyobrazeno na obrázku 5.11. Pokud se člověk dívá na reklamu, je obklopen obdélníkem.

Obrázek 5.11: Vizualizace osoby, která se zrovna nedívá na reklamu

5.8 Experiment a výsledky

Pro dataset dívá se/nedívá se, jsme se snažili anotovat hlavně lidi, kteří se dívají na reklamu. Vybrali 1700 náhodných snímků, pouze na 80 snímcích jsou pozorovatelé reklamy. To znamená, že stejná osoba na více snímcích je tedy započítána pokaždé na každém snímku. Na vybraných snímcích se nacházeli lidi, kteří byli vyhodnoceni, že se dívají na reklamu. Pro představu celkový počet osob byl přibližně 1500. Kvůli časové náročnosti celé práce jsme neanotovali další snímky. Počet neidentifikovaných pozorovatelů je vysoká, protože na těchto snímcích selhal odhad pozice hlavy. Výsledky na tomto datasetu jsou vyznačeny v tabulce:

Počet pozorovatelů (GT)	80
Počet správně predikovaných pozorovatelů (TP)	52
Počet špatně predikovaných pozorovatelů (FP)	10
Počet neidentifikovaných pozorovatelů (FN)	28
Počet správně klasifikovaných negativních případů (TN)	1330

Tabulka 5.2: Výsledky přesnosti dívá se/nedívá se na anotovaným datasetu bez sledování

Z těchto hodnot můžeme vypočítat statistiky, které jsou uvedeny v tabulce 5.3. V datasetu byl velký nepoměr dívajících se a nedívajících se lidí, proto tyto mohou být nepřesné. Senzitivita je 0.65 a udává, jak dobře model detekuje TP. FPR je 0.0074 udává, jak často model dává FP výsledky. S 0.84 náš model udává, jak často jsou výsledky modelu prohlášeny za TP. Přesnost klasifikace je 0.967, a říká nám, jak dobře model celkově klasifikuje případy.

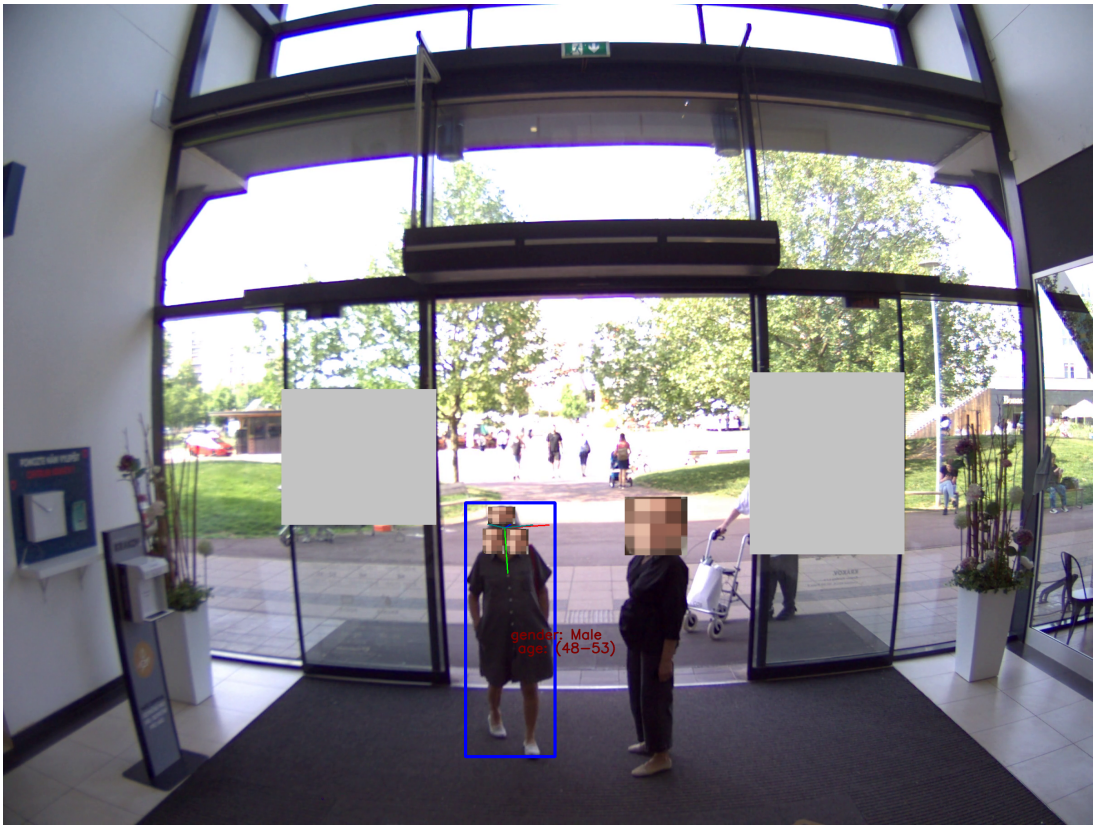
Senzitivita (TPR)	0.65
Chybně pozitivní míra (FPR)	0.0074
Přesnost (angl. precision)	0.84
Přesnost klasifikace (angl. accurancy)	0.967

Tabulka 5.3: Výsledky přesnosti dívá se/nedívá se na anotovaným datasetu bez sledování

Zde jsme to pustili již na sekvenci, kde bylo možné spustit sledovač. Spustili jsme tento algoritmus na téměř 4000 snímcích z videa, kde bylo detekováno 431 identit lidí, při sledování jsem identifikovali jednotlivé chodce na více snímcích a každému člověku jsme přiřadili identitu. Na obrázku 5.12 můžeme vidět vizualizaci odhadu pozice hlavy člověka, který se kouká na reklamu. Průměrný čas každého diváka je 8 snímků, což je méně než 1 sekunda, to je způsobeno tím, že algoritmus často detekuje člověka, který se dívá pouze na jednom snímku. Nejdelší čas sledování diváka je 67 snímků, což je více než 2 sekundy. Odhadnout věk, nebo i věkovou skupinu je i pro člověka složité, ale z vizuálních výsledků tyto dvě věkové skupiny $(25-32), (48-53)$ by byli nejvíce pravděpodobné. Dále odhadnout pohlaví i pro člověka může být těžké, také z vizuálních výsledků můžeme říci, že muži i ženy se tam objevovali v podobném počtu.

Počet identit diváků	99
Počet lidí s reklamou v zorném poli	neznámé
Průměrný čas sledování identity diváka	8 snímků
Nejdelší čas sledování identity diváka	67 snímků
Nejčastější věkové skupiny	$(25-32), (48-53)$ let
Pohlaví	počet žen $\approx$ počet mužů

Tabulka 5.4: Výsledky se sledováním



Obrázek 5.12: Vizualizace člověka, který se dívá

5.9 Diskuze

Na začátku jsme získali a vytvořili vlastní dataset a další potřebná data pro dotrénování detektoru YOLOv7. Na predikci pohlaví jsme využili veřejný dataset PETA. Nejprve jsme vyzkoušeli různé detektory chodců, včetně Pedestron a YOLOv7, a zjistili jsme, že pro naše účely je nejvhodnější použít detektor YOLOv7. YOLOv7 je velmi rychlý a přesný. Jeho váhy jsme kvantizovali z 32 bitů na 16 bitů, což snížilo vytížení paměti.

Výsledky dotrénování detektoru YOLOv7 na predikci pohlaví nebyly tak dobré, jak jsme očekávali. To bylo způsobeno nevhodným datasetem a nedostatečnou kapacitou naší grafické karty. Tento fakt nás donutil hledat další způsoby, jak zlepšit výkon modelu. Proto jsme se rozhodli použít přetrénovanou neuronovou síť [10], která umí detekovat nejen pohlaví, ale i věk.

Další krok, který jsme provedli, byla kalibrace kamery, jejíž parametry jsme použili k predikci toho, zda se člověk dívá na danou reklamu nebo ne. Pro tuto predikci jsme vytvořili vlastní funkci, která využívá odhadu pozice hlavy a souřadnic v reálném světě.

Pro převod souřadnic z 2D do 3D jsme vyzkoušeli 2 různé přístupy, projekční matici a perspektivní transformaci, ale bohužel ani jeden z těchto přístupů nebyl dostatečně přesný. Jedním z problémů bylo to, že kamera neviděla přímo pod sebe a svoje nejbližší okolí (cca 2.5 metru). Pro dosažení přesnějších výsledků bychom potřebovali kameru ze shora nebo bychom mohli vyzkoušet nové algoritmy, jako jsou hloubkové mapy. Tyto metody, by mohli mít lepší přesnost.

Pro odhad pozice hlavy jsme použili neuronovou síť img2pose. I když se jedná o dobrou síť, nedosahovala optimálních výsledků. Proto by bylo vhodné se dále zabývat tímto problémem a hledat nové

přístupy, které by mohly přinést lepší výsledky. Na závěr jsme vytvořili vizualizaci výsledků naší pipeline. Algoritmus jsme vyzkoušeli na anotovaných datech, kde jsme získali důležité statistiky. S 0.84 náš model udává, jak často jsou výsledky modelu prohlášeny za TP. Přesnost klasifikace je 0.967, a říká nám, jak dobře model celkově klasifikuje případy. Bohužel, v tomto anotovaným datasetu byl velký nepoměr dívajících a nedívajících se lidí na reklamu. V případech FN neproběhla klasifikace dívá se/nedívá se z důvodu, že img2pose nepredikovala odhad pozice hlavy. Dále jsme spustili algoritmus na části videa o 4000 snímcích. Celkový počet diváků podle našeho algoritmu bylo 99 lidí. Nejpočetnější věkové skupiny byly (25-32) a (48-53) let. Počet mužů a žen bylo stejné množství a nejdelší čas sledování diváka byly 2 sekundy.

Závěr

Po prostudování problematiky na téma detekce a sledování lidského těla, predikce pohlaví a odhad pozice hlavy pomocí metod hlubokého učení, následovalo hledání vhodného detektoru chodců pro naše účely a zdokumentování dostupných datasetů pro predikci pohlaví z torza/celé postavy chodce. Dále následovalo prostudování v oblasti kalibrace kamery. Dále jsme vytvořili pipelinu, která má na vstupu obrázek a na výstupu textový soubor i vizualizaci, zda se člověk dívá, nebo nedívá na danou reklamu, pohlaví a věk.

Před tím než jsme zhodnotili vybraný detektor chodců, jsme vytvořili vlastní dataset pohlaví o 100 snímcích z daného prostředí, který jsme použili v dalších krocích. Vybraným detektorem byl jednofázový detektor YOLOv7, který je natrénovaný na COCO datasetu a umí klasifikovat až 80 tříd objektů. Tento detektor jsme dotrénovali, aby uměl predikovat pohlaví člověka z celé postavy. K tomu jsme využili veřejný dataset PETA a vlastní anotovaný dataset. Na detektoru jsme vyhodnocovali přesnost. Přesnost této dotrénované sítě na testovací sadě o 706 snímcích je pro muže 0.36 a pro ženy 0.15, tato detekce trvala 4 sekundy pro snímek. Jelikož jsme měli omezenou výpočetní kapacitu a málo trénovacích dat, výsledky nebyly uspokojivé. Proto jsme se rozhodli využít už natrénovanou síť ONNX, která umí predikovat pohlaví i věk, která je součástí celé pipeliny.

Po té jsme YOLOv7 využívali pouze jako detektor pro chodce. Tuto síť jsme optimalizovali, přesněji jsme kvantizovali její parametry. Pro kvantizaci jsme využili knihovnu ONNX RUNTIME, která nám převedla váhy z 32 bitů na 16 bitů, což umožní menší zátěž paměti. Pro tyto detekce jsme spustili sledování pomocí IoU tracker.

Dále jsme vymysleli funkci, která predikovala zda se daný člověk dívá na reklamu. Jako vstup byl brán odhad pozice hlavy a odhad souřadnic člověka v reálném světě a to vše z obrázku.

Pro odhad pozice hlavy jsme využili img2pose algoritmus. Provedli jsme kalibraci kamery a to za pomocí knihovny OpenCV. S touto kalibrací kamery jsme vyzkoušeli odhadnout souřadnice z obrázku v reálném světě. Vyzkoušeli jsme 2 přístupy - pomocí projekční matice a pomocí perspektivní transformace. Perspektivní transformace se zdála být přesnější než projekční matice, ale jsou k ní třeba znát informace o prostředí - minimálně 4 body v reálném světě a projekce těchto 4 bodů do obrázku. Po detekci a sledovaní lidského těla, predikci pohlaví a odhadu pozice hlavy byl algoritmus doplněn o funkce, které vizualizovaly získané výsledky.

Celou pipelinu jsme spustili na snímcích z videa. 1700 snímků jsme anotovali na klasifikaci člověka - dívá se/nedívá se. Senzitivita je 0.65, což nám říká, jak dobře model detekuje správné případy a přesnost je 0.84. V případech, kdy nastalo FN, selhal odhad pozice hlavy ve většině případů. To způsobilo, že obrázek ani nemohl být klasifikován na dívá se/nedívá se. Výsledky mohou být zavádějící, protože poměr dívajících lidí a nedívajících lidí je 80:1620, což je velký nepoměr.

Problém, který se řeší, je velmi komplexní a vyžaduje mnoho úspěšných dílčích kroků k dosažení fungujícího řešení. Byla vyvinuta struktura algoritmu, který umožňuje jednoduché nahrazování a zlepšování jednotlivých modulů. Díky tomuto postupu je možné postupně vylepšovat celý algoritmus a vý-

sledky. Navzdory nepřesvědčivým výsledkům je výsledná práce dobrým odrazovým můstkem pro dotažení celé aplikace do podoby, která je použitelná pro reálné aplikace.

Pro budoucí vývoj aplikace bychom měli anotovat více obrázků v našem prostředí a využít i další veřejné datasety k dotrénování predikce pohlaví z celé postavy. Pro zlepšení pipeliny můžeme nahradit robustnější algoritmus pro odhad pozice hlavy, který by mohly přinést lepší výsledky a odstranil by problém počet neidentifikovaných pozorovatelů (FN). Pro přesnější predikci souřadnic z 2D do 3D bychom potřebovali kameru pozorující prostor ze shora nebo bychom mohli vyzkoušet nové algoritmy, jako jsou hloubkové mapy. Tyto metody, by mohli vést k lepší přesnosti, protože jedním z problémů bylo to, že kamera neviděla přímo pod sebe a svoje nejbližší okolí (cca 2.5 metru).

Literatura

[1] Activation functions in neural networks. https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6. Accessed: 2022-8-8.

[2] Camera calibration. https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html. Accessed: 2022-06-23.

[3] The complete guide to object tracking [+v7 tutorial]. https://www.v7labs.com/blog/object-tracking-guide#h1. Accessed: 2022-3-6.

[4] A comprehensive guide to convolutional neural networks — the eli5 way. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. Accessed: 2022-8-13.

[5] Digital communication - quantization. https://www.tutorialspoint.com/digital_communication/digital_communication_quantization.htm. Accessed: 2023-4-6.

[6] Evaluation metrics for object detection algorithms. https://medium.com/@vijayshankerdubey550/evaluation-metrics-for-object-detection-algorithms-b0d6489879f3 Accessed: 2023-2-27.

[7] File:visual perception human fov.png. https://commons.wikimedia.org/wiki/File:Visual_Perception_Human_FOV.png. Accessed: 2023-4-3.

[8] Homografie a epipolární geometrie. http://trilobit.fai.utb.cz/homografie-a-epipolarni-geometrie. Accessed: 2022-6-21.

[9] Homografie a epipolární geometrie. http://trilobit.fai.utb.cz/homografie-a-epipolarni-geometrie. Accessed: 2022-8-8.

[10] https://github.com/onnx/models/tree/main/vision/body_analysis/age_gender. AgeandGenderClassificationusingConvolutionalNeuralNetworks. Accessed: 2023-4-13.

[11] https://medium.com/geekculture/all-about-yolo-v7-optimization-using-model-scaling-to-trade-off-accuracy-and-computation-e80adfff9d62. AllAboutYOLOV7Optimization: UsingModelScalingtoTradeOffAccuracyandComputation. Accessed: 2023-4-13.

[12] Introduction to artificial neutral networks | set 1. https://www.geeksforgeeks.org/introduction-to-artificial-neutral-networks/. Accessed: 2022-8-13.

[13] Machine learning | an introduction. https://towardsdatascience.com/machine-learning-an-introduction-23b84d51e6d0. Accessed: 2022-8-13.

[14] Neural network quantization: What is it and how does it relate to tinyml? https://www.allaboutcircuits.com/technical-articles/neural-network-quantization-what-is-it-and-how-does-it-relate-to-tiny-machine-learning/. Accessed: 2023-2-18.

[15] Neural network quantization: What is it and how does it relate to tinyml? QuantizeONNXModels. Accessed: 2023-4-13.

[16] Object tracking state of the art 2022. https://medium.com/@pedroazevedo6/object-tracking-state-of-the-art-2022-fe9457b77382. Accessed: 2022-8-13.

[17] One-stage object detection. https://machinethink.net/blog/object-detection/. Accessed: 2022-4-8.

[18] Open images dataset v6 + extensions. https://storage.googleapis.com/openimages/web/index.html. Accessed: 2023-4-8.

[19] Personalized and non-personalized ads. https://support.google.com/admanager/answer/9005435?hl=en. Accessed: 2023-4-13.

[20] Pinhole camera. https://kornia.readthedocs.io/en/0.5.7/geometry.camera.pinhole.html. Accessed: 2023-4-6.

[21] Quantize onnx models. https://onnxruntime.ai/docs/performance/model-optimizations/quantization.html#onnx-quantization-representation-format. Accessed: 2023-4-6.

[22] Understanding lens distortion. https://learnopencv.com/understanding-lens-distortion/. Accessed: 2023-4-6.

[23] Various optimization algorithms for training neural network. https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6. Accessed: 2022-8-13.

[24] Yolo: Real-time object detection explained. https://www.v7labs.com/blog/yolo-object-detection. Accessed: 2023-1-12.

[25] Vitor Albiero, Xingyu Chen, Xi Yin, Guan Pang, and Tal Hassner. img2pose: Face alignment and detection via 6dof, face pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7617–7627, 2021.

[26] Erik Bochinski, Volker Eiselein, and Thomas Sikora. High-speed tracking-by-detection without using image information. In *2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS)*, pages 1–6. IEEE, 2017.

[27] Antonio Brunetti, Domenico Buongiorno, Gianpaolo Francesco Trotta, and Vitoantonio Bevilacqua. Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. *Neurocomputing*, 300:17–33, 2018.

[28] Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Zeroq: A novel zero shot quantization framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13169–13178, 2020.

[29] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: high quality object detection and instance segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 43(5):1483–1498, 2019.

[30] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[31] Sruti Das Choudhury, Srikanth Maturu, Ashok Samal, Vincent Stoerger, and Tala Awada. Leveraging image analysis to compute 3d plant phenotypes based on voxel-grid plant reconstruction. *Frontiers in Plant Science*, 11:521431, 2020.

[32] Yubin Deng, Ping Luo, Chen Change Loy, and Xiaoou Tang. Pedestrian attribute recognition at far distance. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 789–792, 2014.

[33] K Divya, K Damodhar Rao, and Prasanta Kumar Sahoo. Pedestrian gender prediction using machine learning.

[34] Yunhao Du, Yang Song, Bo Yang, and Yanyun Zhao. Strongsort: Make deepsort great again. *arXiv preprint arXiv:2202.13514*, 2022.

[35] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *The Journal of Machine Learning Research*, 20(1):1997–2017, 2019.

[36] Angela Fan, Pierre Stock, Benjamin Graham, Edouard Grave, Rémi Gribonval, Herve Jegou, and Armand Joulin. Training with quantization noise for extreme model compression. *arXiv preprint arXiv:2004.07320*, 2020.

[37] Olivier Faugeras and Olivier Autor Faugeras. *Three-dimensional computer vision: a geometric viewpoint*. MIT press, 1993.

[38] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. Ieee, 2008.

[39] Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. *Advances in neural information processing systems*, 28, 2015.

[40] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. *arXiv preprint arXiv:2103.13630*, 2021.

[41] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[42] Gene H Gloub and Charles F Van Loan. Matrix computations. *Johns Hopkins Universtiy Press, 3rd edtion*, 1996.

[43] Michael Haenlein and Andreas Kaplan. A brief history of artificial intelligence: On the past, present, and future of artificial intelligence. *California management review*, 61(4):5–14, 2019.

[44] Richard I Hartley. Self-calibration from multiple views with a rotating camera. In *European Conference on Computer Vision*, pages 471–478. Springer, 1994.

[45] Irtiza Hasan, Shengcai Liao, Jinpeng Li, Saad Ullah Akram, and Ling Shao. Generalizable pedestrian detection: The elephant in the room. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11328–11337, June 2021.

[46] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[47] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[48] Alan D Jones. Manual of photogrammetry, eds cc slama, c. theurer and sw hendrikson, american society of photogrammetry, falls church, va., 1980, 180× 260mm, xvi and 1056 pages (with index), 72 tables, 866 figures. isbn 0 937294 01 2. *Cartography*, 12(4):258–258, 1982.

[49] Khalil Khan, Rehan Ullah Khan, Riccardo Leonardi, Pierangelo Migliorati, and Sergio Benini. Head pose estimation: A survey of the last ten years. *Signal Processing: Image Communication*, 99:116479, 2021.

[50] Jinsu Lee, Sanghoon Kang, Jinmook Lee, Dongjoo Shin, Donghyeon Han, and Hoi-Jun Yoo. The hardware and algorithm co-design for energy-efficient dnn processor on edge/mobile devices. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 67(10):3458–3470, 2020.

[51] Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, et al. Yolov6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*, 2022.

[52] Dangwei Li, Zhang Zhang, Xiaotang Chen, and Kaiqi Huang. A richly annotated pedestrian dataset for person retrieval in real surveillance scenarios. *IEEE transactions on image processing*, 28(4):1575–1590, 2018.

[53] David Liebowitz and Andrew Zisserman. Metric rectification for perspective images of planes. In *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No. 98CB36231)*, pages 482–488. IEEE, 1998.

[54] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[55] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[56] Yutian Lin, Liang Zheng, Zhedong Zheng, Yu Wu, Zhilan Hu, Chenggang Yan, and Yi Yang. Improving person re-identification by attribute and identity learning. *Pattern Recognition*, 2019.

[57] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[58] J More. ªthe levenberg-marquardt algorithm, implementation, and theory, º numerical analysis, ga watson, ed, 1977.

[59] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *18th international conference on pattern recognition (ICPR'06)*, volume 3, pages 850–855. IEEE, 2006.

[60] Rajeev Ranjan, Vishal M Patel, and Rama Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):121–135, 2017.

[61] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[62] Fabio Remondino and Clive Fraser. Digital camera calibration methods: considerations and comparisons. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(5):266–272, 2006.

[63] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 32–36. IEEE, 2004.

[64] Han Shen, Lichao Huang, Chang Huang, and Wei Xu. Tracklet association tracker: An end-to-end learning-based association approach for multi-object tracking. *arXiv preprint arXiv:1808.01562*, 2018.

[65] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[66] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[67] Sasha Targ, Diogo Almeida, and Kevin Lyman. Resnet in resnet: Generalizing residual architectures. *arXiv preprint arXiv:1603.08029*, 2016.

[68] Paul Viola, Michael Jones, et al. Robust real-time object detection. *International journal of computer vision*, 4(34-47):4, 2001.

[69] Jianhua Wang, Fanhuai Shi, Jing Zhang, and Yuncai Liu. A new calibration model and method of camera lens distortion. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5713–5718. IEEE, 2006.

[70] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020.

[71] Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995.

[72] Jiaxiang Wu, Cong Leng, Yuhang Wang, Qinghao Hu, and Jian Cheng. Quantized convolutional neural networks for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4820–4828, 2016.

[73] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[74] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *International Journal of Computer Vision*, 129(11):3069–3087, 2021.

[75] Zhaoxiang Zhang, Min Li, Kaigi Huang, and Tieniu Tan. Practical camera auto-calibration based on object appearance and motion for traffic scene visual surveillance. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[76] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000.