



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**AUTOMATICKÉ VYHODNOCENÍ ZÁZNAMU CÍLOVÉ
KAMERY**

AUTOMATIC EVALUATION FOR PHOTO FINISH SYSTEM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

VOJTECH JAHODA

VEDOUCÍ PRÁCE
SUPERVISOR

MICHAL HRADIŠ, Ing., Ph.D.

BRNO 2021

Zadání bakalářské práce



24066

Student: **Jahoda Vojtěch**

Program: Informační technologie

Název: **Automatické vyhodnocení záznamu cílové kamery**
Automatic Finish Camera

Kategorie: Zpracování obrazu

Zadání:

1. Seznamte se s principem fungování cílové kamery a vytvořte si přehled o současných možnostech využití cílové kamery a platných pravidlech vyhodnocování jejich záznamů.
2. Seznamte se s přístupy počítačového vidění a neuronovými sítěmi pro zpracování obrazu.
3. Navrhněte metodu pro automatické vyhodnocování záznamu cílové kamery nebo metodu, která by řešila podstatnou část takového vyhodnocení.
4. Seznamte se s potřebnými knihovnami pro řešení a vytvořte si datovou sadu záznamů cílových kamer.
5. Implementujte vybranou metodu a vyhodnoťte její vlastnosti zvolenou datovou sadou.
6. Porovnejte dosažené výsledky a diskutujte možnosti budoucího vývoje.
7. Vytvořte stručné video, prezentující vaši práci, její cíle a výsledky.

Literatura:

- Cao et al.: OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. PAMI, 2019.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 4.

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

Vedoucí práce: **Hradiš Michal, Ing., Ph.D.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2020

Datum odevzdání: 12. května 2021

Datum schválení: 30. října 2020

Abstrakt

Tato práce řeší automatizaci vyhodnocování cílové fotografie v atletice. Detekce závodníků na cílové fotografii byla provedena pomocí knihovny OpenPose. Následně byla provedena segmentace pozadí závodníka pro odstranění šumu a ořez pro jednotlivé závodníky. Samotné vyhodnocení je řešeno pomocí regresní konvoluční neuronové sítě. Při detekci postav na fotografii bylo dosaženo přesnosti 70.94% a 89.28% vyhodnocených záznamů dosáhlo přesnosti do 10 ms. Hlavní přínos této práce je pro začínající cílové rozhodčí, neboť bude mít již předem k dispozici vyhodnocený cílový záznam, který mu ušetří čas. Dalším využitím je možnost pro trenéry a závodníky, kteří si budou mocí ověřit vyhodnocené záznamy.

Abstract

This thesis addresses the automation of photo finish evaluation in athletics. The detection of athletes in the target photo has been performed using the OpenPose library. Subsequently, athlete background segmentation has been performed to remove noise and cropping for individual athletes. The evaluation itself has been solved using a regression convolutional neural network. The accuracy of 77.60% has been achieved through detecting the figures in the photo and 89.28% of the evaluated records has reached the accuracy lesser than 10 ms. The main benefit of this thesis is for novice photo finish referees, since they will have the evaluated target record available to them beforehand. Another usage serves for coaches and competitors, since they will be able to easily validate the evaluated records by themselves.

Klíčová slova

vyhodnocování cílových fotografií, segmentace cílových fotografií, Přesný cílový systém, regresní konvoluční neuronové sítě, strojové učení, segmentace pozadí

Keywords

Photo finish evaluation, photo finish segmentation, Accurate photo finish system, Regression convolutional neural networks, Machine learning, Background segmentation

Citace

JAHODA, Vojtěch. *Automatické vyhodnocení záznamu cílové kamery*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Michal Hradiš, Ing., Ph.D.

Automatické vyhodnocení záznamu cílové kamery

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Michala Hradiše. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Vojtěch Jahoda
18. května 2021

Poděkování

Chtěl bych poděkovat panu Ing. Michalu Hradišovi, Ph.D. za odborné vedení mé práce, konzultace a rady, které mi pomohly tuto práci vytvořit.

Obsah

1	Úvod	2
2	Cílový systém	3
2.1	Základní pojmy pro atletiku	3
2.2	Funkcionalita cílové kamery	4
2.3	Cílová fotografie	6
2.4	Postavení a nastavení cílové kamery	8
2.5	Pravidla pro běžecké disciplíny	10
3	Shrnutí dosavadního stavu	13
3.1	Neuronové sítě	13
3.2	Konvoluční neuronové sítě	14
3.3	Segmentace pozadí	17
3.4	Předchozí studie automatizace cílového systému	18
4	Metoda pro automatizaci cílového systému	20
4.1	Zpracování cílového záznamu	21
4.2	Detekce postav	21
4.3	Vytvoření výřezů závodníků	23
4.4	Segmentace	24
4.5	Regresní konvoluční neuronová síť	26
5	Implementace a experimenty	27
5.1	Implementace	27
5.2	Datová sada	28
5.3	Vytváření datové sady	29
5.4	Experimenty	31
5.5	Diskuze	35
6	Závěr	36
	Literatura	37
	A Struktura odevzdaných souborů	38

Kapitola 1

Úvod

Cílové systémy s cílovou kamerou jsou dnes již samozřejmostí na každých oficiálních atletických závodech na světě. Zajišťují, aby byl každý běžecký závod změřen přesně, jasně a nevytvářel se prostor dohadům o zmáčknutí stopek. Jednotlivé cílové fotografie vyhodnocuje cílový rozhodčí, který je školen Českým atletickým svazem, *ale chybovat je lidské*. ale i ten může vyhodnotit fotografie nepřesně, se subjektivní chybou (rozdíly jsou ale minimální v porovnání s ručním měřením).

Tato práce se zabývá minimalizací prostoru pro tvorbu těchto chyb z nedbalosti či nevědomosti, a to pomocí programu, který vyhodnocuje na základě zkušeností jiného rozhodčího v podobě počítače.

V minulosti se veškeré závody měřily ručně pomocí stopek, kdy nastávaly problémy a spory o přesnosti měření. Navíc jakékoli rekordy nebyly zpětně dokazatelné. Momentální stav je řešení pomocí cílové kamery a ručního vyhodnocení jejich záznamů. Cílová kamera funguje na principu zachycení sloupových úzkých snímků cílové čáry. Jednotlivé snímky se poté poskládají za sebe a vytvoří cílovou fotografii, která je poskládána z okamžíků doběhů závodníků do cíle.

Plná automatizace systému cílové kamery je, dle mého názoru, dalším krokem v evoluci automatizovaných cílových systémů. Znamenalo by to, že u cílové kamery nemusí figurovat taklik lidí, závěry rozhodčího nejsou zatížené subjektivní chybou (a tím pádem se vyloučí napadení výsledku) a cílový rozhodčí se potencionálně bude moci věnovat i zadávání dat do systému zpracování výsledků, což povede k levnějšímu a efektivnějšímu provozu. K výběru tohoto tématu mě vedla vlastní zkušenost. Jsem cílový rozhodčí, vyškolený Českým atletickým svazem, a při vyhodnocování cílových záznamů jsem si mnohdy nebyl jist, jak záznam přesně vyhodnotit.

Cílem mojí bakalářské práce je vytvoření programu, který správně určí počet postav a polohu postav na fotografii a pro každého závodníka na fotografii určí správné místo pro vyhodnocení.

Následující kapitola pojednává o současném stavu metod a technologií, které souvisí s problematikou vyhodnocování cílové fotografie. v kapitole "Zhodnocení aktuálního stavu a návrh řešení" jsou pak současné způsoby fúze a detekce zhodnoceny a je navrhнуто řešení, které bude využito. v kapitole "Popis vlastní práce" se nachází popis způsobu řešení daného zadání a testování výsledného systému.

Kapitola 2

Cílový systém

Tato kapitola se věnuje seznámení čtenáře s pojmy, technologiemi a metodami, které je možné použít při návrhu a realizaci vlastního řešení práce. Protože samotné vyhodnocování a fungování cílového systému je důležitou částí této práce, bude kapitola pojednávat i o těchto záležitostech. Nejedná se o encyklopedický přehled všeho, čeho se práce týká.

2.1 Základní pojmy pro atletiku

Cílový systém

Cílový systém se běžně skládá ze tří hlavních komponent:

- Řídící HUB s počítačem, kteří společně tvoří „mozek“ celého cílového systému. Přijímají a odesílají řídící signály pro ostatní komponenty systému.
- Startovací čidlo, které spouští časomíru systému a okamžik jeho spuštění figuruje v cílovém systému jako časový bod 0.0 s.
- Cíl - skládá se z fotobuněk, které zaznamenávají proběhnutí závodníka cílem, a cílové kamery, která zaznamená přesný obrazový záznam průběhu závodu v oblasti cílové čáry. Fotobuňky bývají nastavené většinou na úroveň pasu závodníka, aby se nemohlo stát, že závodník paprsek fotobuněk překročí či podběhne (v případě, kdy bývá v předklonu).

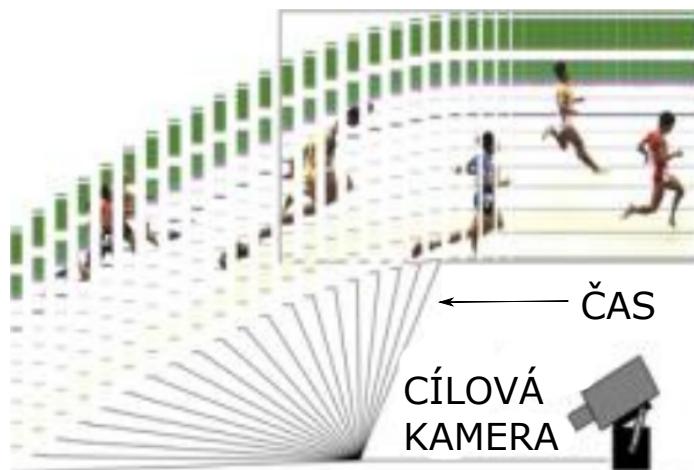
Běžecké povrchy

Na běžeckých oválech v České republice se používají nejčastěji tři druhy povrchů. Jejich barva může ovlivnit kvalitu pořízených cílových fotografií.

- **Tartan** obvykle oranžové až červené barvy, mírně se drolí (čím starší, tím se drolí více). na území ČR je jednoznačně nejrozšířenější.
- **Antuka** obvykle také oranžové barvy. Dříve se používala hojně na většině oválů, ale protože to není příjemný povrch, postupně se nahrazuje tartanem. v dnešní době je vidět spíše na tenisových kurtech než na atletických oválech.
- **Mondo** obvykle modré barvy. Údajně se jedná o nejrychlejší povrch na světě (světový rekord ve většině sprinterských disciplín byl zaběhnutý právě na něm). Jeho povrch má strukturu šestiúhelníkových „bubblek“.

2.2 Funkcionalita cílové kamery

Cílová kamera funguje na principu "strip photography" (do češtiny se volně překládá jako pásové nebo řádkové fotografování, ale tento anglický výraz je dohledatelný) a celá cílová kamera je v ČR spíše známá jako řádková kamera. Jedná se o technologii, která snímá 2D obraz jako sekvenci 1D proužků (*1D proužek je vertikální fotografie o šířce pouze jednoho pixelu a výšce celé snímané oblasti*) dané oblasti (zde je daná oblast určena jako cílová čára). Při pohybu na horizontální ose se na fotografii pohybuje i časová značka pořízení jednotlivých proužků kamery 2.1. Jednotlivé snímky jsou úzké proužky o velikosti 1px na šířku, které ale na výšku zaznamenávají celou výšku cílové čáry. Pro poskládání kompletní fotky se tedy jednotlivé proužky dají vedle sebe tak, že časově první snímek je na pravé straně a poslední na levé.



Obrázek 2.1: Obrázek znárožující sestavení cílové fotografie ze záznamů cílové kamery. Převzat z práce o cílových metodách[3]

Podobný princip se používá u vysokorychlostních kamer, kdy díky snížení snímaného prostoru lze zvýšit snímkovací frekvenci zařízení. Pro cílovou kameru se běžně používá snímkovací frekvence 1 000 snímků za sekundu (tj. přesnost 0.001 s), ale existují i cílové kamery, které operují při frekvenci 10.000 snímků za sekundu (tj. přesnost 0.0001 s). Tato metoda zaznamenávání obrazu je však zcela dokonalá pro potřeby atletiky. Protože zaznamenává pouze časové události proběhnutí prostorem cílové čáry, tak místo protnutí cílového prostoru závodníkem je i časem doběhu závodníka (časová značka na části těla závodníka, která je vyhodnocena).

Výhody zachycení cílovou kamerou

- Zaznamenávání časů je přesné na minimálně 1ns (přesnost může být i vyšší, záleží od cílové kamery).
- Vyhodnocené záznamy jsou zpětně dohledatelné (celý průběh cílem je uložen v cílovém záznamu a může být zpětně vyžádán trenéry nebo závodníky).
- Vyhodnocené záznamy lze vyexportovat pro pozdější zpracování cílovým systémem.

Nevýhody zachycení cílovou kamerou

- Pro vyhodnocování záznamů je zapotřebí školeného rozhodčího, který musí být do statečně zkušený, aby rozpoznal všechny problematické situace (např. vyhodnocení doběhu v předklonu, vyhodnocení závodníků, kteří jsou v zákrytu nebo závodníků, kteří cílem neproběhli, ale prošli).
- Cílový systém není levná záležitost.

Další metody vyhodnocování doběhu do cíle

Mezi další metody, které se používaly pro zachycení doběhu do cíle, byla klasická video-kamera, která natáčela celý průběh cílem na videokazetu či jiné digitální médium (někde se tato metoda ještě používá, ale většinou je to už záležitost historie). Kamera byla postavena kolmo k cílové čáře (stejně jako u cílové kamery), aby bylo možné zkoumat přesný průběh závodníka cílem. Následný záznam doběhu do cíle vyhodnocoval rozhočí tak, že si zpomaleně přehrál záznam a snažil se najít přesný čas, ve kterém daný závodník proběhl cílem. v dnešní době se tato metoda považuje za obdobu ručního měření a není možné pomocí ní zaběhnout oficiální čas.

Další možnost zaznamenání doběhu do cíle je použití tzv. čipové brány, která zaznamenává průběh účastníků prostorem cílové čáry. Nevýhodou této technologie je, že každý závodník musí mít nasazen čip, který brána bude detekovat.

2.3 Cílová fotografie

Díky principu popsanému výše 2.2 se nepohyblivé snímané předměty projeví jako statická jednobarevná čára na pozadí, ale pohybující se objekty jsou zachyceny z profilu s určitým zkreslením. Tento způsob zachycení tak vede mnohdy k deformacím postavy a ty vedou k problému při rozpoznávání na fotografií.

Typy deformací při doběhu do cíle

Zkreslení pro pohyblivé objekty (závodníky) se odvíjí od relativní rychlosti při doběhu vůči kameře. Čím rychleji závodník proběhne, tím kratší je možnost záznamu průběhu a jeho obrazové stopy na snímku. Naopak - čím pomaleji proběhne závodník cílem, tím roztáhlejší a zdeformovanější je jeho postava.



Obrázek 2.2: Na tomto obrázku je vidět příklad pomalé deformace (levý) vůči tomu jak vypadá normální „rychlý“ průběh cílem (pravý)

Nejčastější případy deformací závodníků:

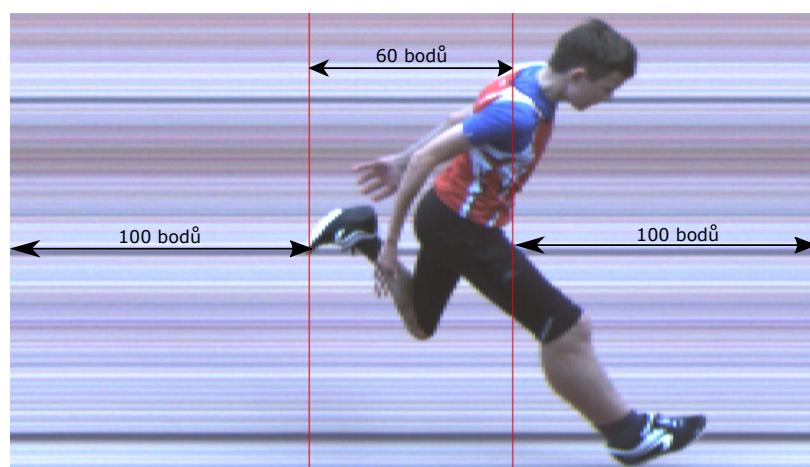
- Závodník došlápně na cílovou čáru v momentu průběhu cílem, a tím se snímek došlapu natáhne pod celou délkou závodníka.
- Závodník při průběhu cílem kmitne rukou, a tím se jeho ruka může roztáhnout či zkrátit (v závislosti na tom, jakým směrem kmitl).



Obrázek 2.3: Na tomto obrázku je vidět příklad deformace nohy (levý) a deformace ruky (pravý)

Sestavení výsledné fotografie

Výsledná fotografia se sestavuje podle dat z kamery a z fotobuněk cílového systému (více [2.1](#)). Podle časové značky začátku a konce sepnutí fotobuňky se určuje prostor na výsledné fotografii pro každého závodníka. Obvykle tyto údaje bývají pro sprinty 100 snímků před sepnutím a 100 snímků za ukončením sepnutí pro každého závodníka. Příklad: *Mějme závodníka, u kterého fotobuňky seply v čase 8,72 s a sepnutí skončilo v čase 8,78 s. Víme, že před i za sepnutím je 100 snímků, které při snímkovací frekvenci 1.000 snímků za sekundu znamenají 0,1 s. Celková doba záznamu pro tohoto závodníka je tedy 0,26 s, což při naší snímkovací frekvenci znamená 260 bodů (časových značek). Tyto snímky před a za sepnutí fotobuňky se přidávají kvůli přehlednosti a předcházejí ztrátě dat.* Viz. [2.4](#)



Obrázek 2.4: Na obrázku je viděl příklad celého obrazového záznamu pro jednotlivého závodníka. Červené čáry zde symbolizují místa sepnutí.

Formát výsledné fotografie

Pro svoji práci jsem se rozhodl využít cílové kamery od společnosti TimeTronics [\[8\]](#), která na svém výstupu poskytuje fotografiu ve formátu .MF4 (nejedná se o standardní MDF souborem, ale soubor s nějakou vlastní architekturou, která bohužel není veřejná). z tohoto důvodu jsem každou vyhodnocenou fotografiu musel vyexportovat do formátu .PNG pomocí programu MacFinish od firmy TimeTronics. Po vyexportování se každý původní pixel fotografie vyexportoval jako políčko o velikosti 3x3 pixely a tím se celková velikost fotografie zvětšila 9x.

2.4 Postavení a nastavení cílové kamery

V této sekci je popsáno správné nastavení kamery a její postavení vůči atletické dráze pro vyprodukování nejlepších možných výsledků.

Ideální podmínky pro záznam

Ideální podmínky pro měření cílovou kamerou by měly zaručit, že je fotografie zcela bílá, kromě černých čar drah a závodníků. Splnění těchto podmínek se dosáhne pomocí zcela bílé cílové čáry po celé šírce cíle, zcela bílým prknem nebo zábranou bílé barvy na protější straně cílové čáry ke kameře a černými čtverečky nalepenými na okraji cílové čáry v místech, kde končí čáry drah. Bohužel, tyto ideální podmínky se dají udržet pouze na velkých stadionech, kde se hřiště řádně spravuje a udržuje. na většině menších závodů se hřiště tolik neudržují, a proto cílová čára většinou není zcela bílá, ale prosvítá pod ní běžecký povrch [2.1](#).



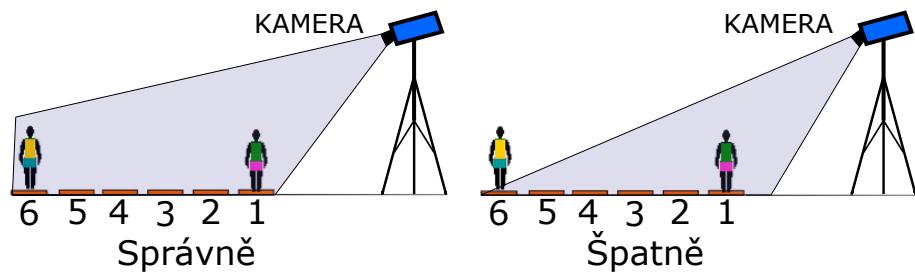
Obrázek 2.5: Na obrázku je vidět detail protější hranice cílové čáry oproti kameře, kde za dráhou je zábrana, která navazuje bílou barvou na cílovou čáru. a také malý černý čtverec na cílové čáře mezi 1 a 2 drahou.

Instalace kamery

Kamera musí být postavena tak, aby byla kolmo na cílovou čáru. Kamera vždy musí být umístěna tak, aby na cílovou dráhu nahlížela z výšky. Celková výška kamery a náklon na dráhu musí být takové, aby v záběru byl vidět jednak minimálně okraj nejbližší dráhy, jednak velikost jednoho člověka nad nejvzdálenější dráhou - viz. [2.6](#), a zároveň tak, aby závodníci, kteří jsou v zákrytu, byli od sebe rozeznat. Většina atletických hřišť má na kameru přímo určené místo. u stadionů, kde je tribuna vedle cílové kamery, bývá toto místo až u střechy tribuny.

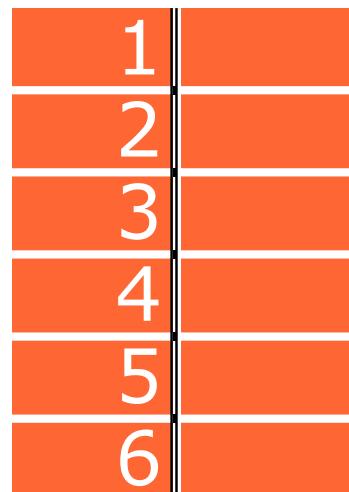
Snímání kamery

Při nastavování kamery před konkrétním závodem se vybírá oblast, ze které kamera bude skládat výslednou fotografiю (1D proužek [2.2](#)). Tato oblast musí být na cílové čáře tak, aby její levá hrana končila na levé hraně cílové čáry a to ve všech jejích místech [2.7](#)(toto místo



Obrázek 2.6: Na obrázku je vidět příklad umístění kamery, na levé straně je správné umístění, kdy je viděl celá 1. dráha i volné místo nad 6. drahou a na pravé straně je příklad špatného umístění, kdy běžec v 6. dráze není zaznamenán

vychází z pravidel 2.5). Aby se docílilo tohoto stavu, lze cílovou kameru různě naklánět a natáček tak, aby byla kamera ve správném postavení.



Obrázek 2.7: Na obrázku je viděl cílová čáry, u které je černými naznačená snímaná výseč.

2.5 Pravidla pro běžecké disciplíny

V této části jsou popsána některá pravidla, která se vztahují k běžeckým disciplínám a vyhodnocování cílových fotografií. Jsou to výňatky a vlastní interpretace z české verze atletických pravidel [7]. Celosvětově se o vydávání a spravování pravidel stará organizace IAAF (dříve "Mezinárodní asociace atletických federací", dnes přejmenovaná na "Světová atletika", zkratka zůstává stejná).

Běžecká trať

Běžecká trať začíná vnější stranou startovací čáry tj. blíže k závodníkovi (proto závodníci nesmí při startu mít ruce na čáře) a končí okrajem cílové čáry blíže ke startu (na tuto část se také nastavuje snímání prostoru cílové čáry, kde se také snímá cílový prostor cílovou kamerou). Do běžecké dráhy každého závodníka ještě navíc patří prostor mezi startovní a cílovou čarou a pravá (vnější) čára dráhy každého závodníka. Levá (vnitřní) čára dráhy do závodníkovy dráhy nepatří. Pokud závodník vyběhne z prostoru daného běžeckou dráhou, je diskvalifikován. Patří sem i vystoupení z dráhy, tj. přešlap levé (vnitřní) čáry, která již patří do závodní trati vedlejšího závodníka.

Cílový čas každého závodníka je určen jako moment, kdy překročil cílovou čáru. Pravidla tento moment definují jako „Okamžik, kdy kterákolič část těla atleta (tj. trup, nikoliv hlava, krk, paže, noha, ruka, chodidlo) dosáhne svislé roviny procházející okrajem cílové čáry bližším startu“¹. Nejčastěji vyhodnocené místa jsou ramena a hrudník.

Nulový test

K tomu, aby mohly být výsledky cílové kamery na každém závodu brány jako oficiální, je zapotřebí vždy před začátkem závodu provést takzvaný „nulový test“ (častěji se používá jeho anglický název „zero test“). Při nulovém testu se ověřuje, jestli zpoždění mezi výstřelem ze startovní pistole (spuštěním snímání) a sepnutím fotobuněk je maximálně opožděno o 0.001 s. Za provedení nulového testu zodpovídá vedoucí rozhodčí cílové kamery a jeho provedení je požadováno při jakémkoliv prokazování rekordu. Nulový test se provádí tak, že si rozhodčí stoupne na cílovou čáru mezi fotobuňky (musí být sepnuté). Následně vystřelí ze startovní pistole tak, aby mířila přímo na cílovou čáru v prostoru, kde to snímá cílová kamera. Následně na cílové kamere je vidět moment, ve kterém byl výstřel proveden. Tento moment výstřelu se vyhodnotí a jeho „cílový čas“ je zpoždění mezi výstřelem a sepnutím fotobuněk 2.8.

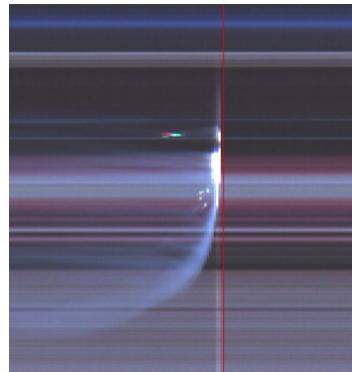
Nejčastější případy vyhodnocení doběhů

Při vyhodnocování cílových záznamů se vždy ignoruje pozice hlavy a rozhodčí se zaměřuje hlavně na ramena a hrudník 2.9, které bývají nejčastějšími případy částí těla, které se vyhodnocují. U ramen mohou nastat dvě situace a vyhodnocování je znatelně zrádnější.

První situace nastává, když první v cíli je rameno blíže ke kameře (pravé rameno) a závodník rameno při doběhu „vystrčil před sebe“. Pak se vyhodnocuje střed ramena. V případě, že závodník doběhl s ramenem blíže ke kameře, ale nemá jej „vystrčené před sebe“, vyhodnocuje se vršek ramena (část ramena, která prošla cílem jako první) 2.10.

Druhá situace nastává v moměntě, kdy první část závodníka, která projde cílem, je rameno vzdálenější od kamery (levé rameno). Nezkušenému oku se může zdát, že se vy-

¹pravidla atletiky [7] pravidlo 19.2 (strana 129)



Obrázek 2.8: Na obrázku je viděl ukázka toho jak vypadá nulový test.



Obrázek 2.9: Na obrázku je viděl ukázka toho jak vypadá vyhodnocený doběh pro muže a ženu v oblasti hrudníku.

hodnocuje hlava jako část, která doběhla první, ale to by bylo proti pravidlům. Pokud je tedy rameno v zákrytu těla, tak se hodnocuje jeho první část (nebo předpokládaná část, pokud to není zcela viděl přes hlavu), která prošla cílem. Když se stane, že rameno, které je dál od kamery, je „vystrčeno“ před závodníka, postupuje se stejně jako u pravého ramene (viz výše) a určí se jeho střed [2.11](#).



Obrázek 2.10: Na obrázku je viděl ukázka doběhu závodníků, u kterých se vyhodnocuje bližší rameno, jako první část těla co přoběhla cílem.



Obrázek 2.11: Na obrázku je viděl ukázka doběhu závodníků, u kterých se vyhodnocuje vzdálenější rameno, jako první část těla co přoběhla cílem.

Kapitola 3

Shrnutí dosavadního stavu

3.1 Neuronové sítě

Lineární vrstvy

Lineární vrstva je charakteristická tím, že plně propojuje všechny uzly předchozí vrstvy se všemi uzly následující vrstvy. Tato spojení slouží pro distribuci všech informací předchozí vrstvy. Pro každou lineární vrstvu je třeba předem zadat rozměr vstupní matice a rozměr výstupní matice. Samotný výpočet pro tuto vrstvu vypadá jako zobrazení vstupu na výstup. Nevýhodou této vrstvy je velký počet parametrů, který je pro vstupní vektor o velikosti (délce) In a výstupní vektor o velikosti (délce) Out roven $Out(In + 1)$.

Učení sítí

Učení neuronových sítí je inspirováno učením člověka (konkrétněji spíš malého dítěte), který zkouší pojmenovávat věci kolem sebe a okolí mu říká, jestli je to správně nebo jak moc je to špatně. Tuto úlohu v neuronové síti řeší ztrátová funkce (loss funkce), která řeší, jak moc se výsledná predikce z neuronové sítě vzdálila od správné anotace daného vstupu. Fáze učení neuronové sítě probíhá v mnoha iteracích, během kterých se snaží naučit approximovat daný problém.

Pokud se ale neuronová síť nechá projít velkým počtem iterací, tak vzniká nebezpeční přeúčení. Jediný spolehlivý způsob detekce přeúčení neuronové sítě je sledování trénovací a testovací chyby na datech (v ideálním případě by měly obě tyto veličiny konvergovat k nule). V reálných podmínkách ovšem často nastává moment, ve kterém chybovost na trénovací sadě jde stále blíže k 0, ale chybovost na testovací sadě se začíná opět zvyšovat. V této situaci se můžeme bavit o přetrénovanosti neuronové sítě.

Augmentace dat

Augmentace dat se provádí pro zvýšení počtu dat v datasetu, protože obecně platí, že čím více dat je v Datasetu, tak tím lépe se neuronová síť zvládne naučit daný úkol. Mezi základní typy augmentace dat patří například:

- **Rotace** - různě natočí obrázky v datasetu, simuluje různé natočení, pod kterým může být obrázek potencionálně vyfocen.
- **Výřez** - udělá náhodný výřez z obrázku, simuluje situace, kdy se nepovede přesně zachytit objekt, který měl být zachycen.

- **Převrácení** - zrcadlově převrátí fotky, simuluje zachycení objektu na obrázku z druhé strany, než bylo provedeno.
- **Barevná augmentace** - náhodně upraví jas, kontrast, sytost a odstín u obrázků, simuluje zachycení obrázků v různých světelných podmírkách.
- **Přidání šumu** - do obrázku se přidá „Gaussian“ šum, simuluje zachycení obrázku v horší kvalitě.
- **Posun** - pokud se jako dataset využívá ořez původní fotografie, tak se tento ořez může posunout o několik bodů do různých směrů, simuluje nepřesnost vyhodnocení výřezů.

3.2 Konvoluční neuronové sítě

Konvoluční neuronové sítě spadají pod neuronové sítě a vyznačují se použitím konvolučních a pool vrstev. Díky této vrstvám jsou schopny lépe zvládat vstupy velkých rozměrů, jako jsou obrázky, za použití mnohem menšího výpočetního výkonu, protože používají menší počet parametrů. Díky kombinaci pool a konvolučních vrstev se poté síť snáze učí.

Konvoluční vrstva

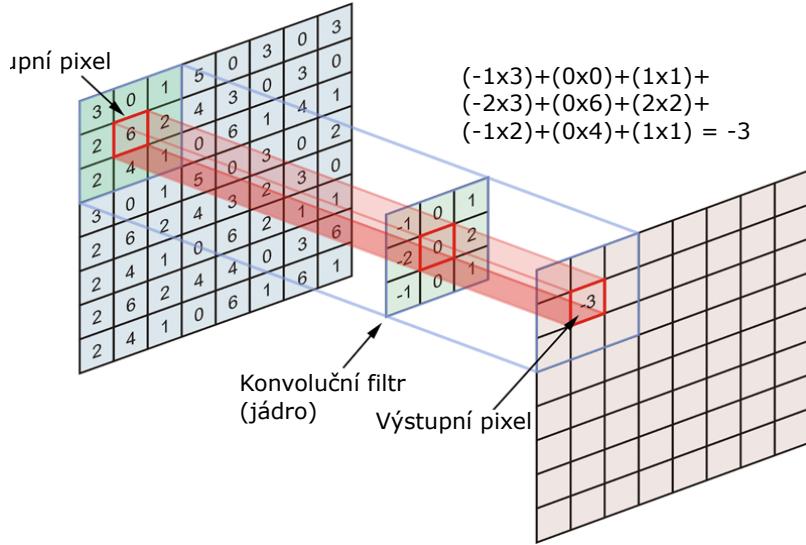
Tato vrstva vykonává operaci konvoluce. Každá taková operace potřebuje konvoluční jádro, které bývá nejčastěji matice o rozměrech 3×3 . Pro celkovou aplikaci jsou potřeba ještě informace o paddingu a kroku. Krok je vždy posun aplikace konvolučního jádra na vstupu (většinou bývá jednotkový, jinak by nastávalo zároveň i zmenšení výsledné matice). Padding znamená přidané ohraničení kolem vstupního obrázku nulami a slouží k tomu, aby se výsledný obraz po aplikaci nezmenšil (konvoluce lze aplikovat pouze na pixely, které kolem sebe mají další pixely, takže pro hraniční ji není možné bez paddingu provést). Tedy pro vstup o velikosti $i \times j$, jádře o rozměrech 3×3 s jednotkovým krokem a nulovým paddingem můžeme vyjádřit konvoluci v konvoluční neuronové síti jako 3.1.

$$\mathbf{Y}[i, j] = \sum_{k=1}^3 \sum_{l=1}^3 \mathbf{X}[k, l] \mathbf{F}[i - k, j - l] \quad (3.1)$$

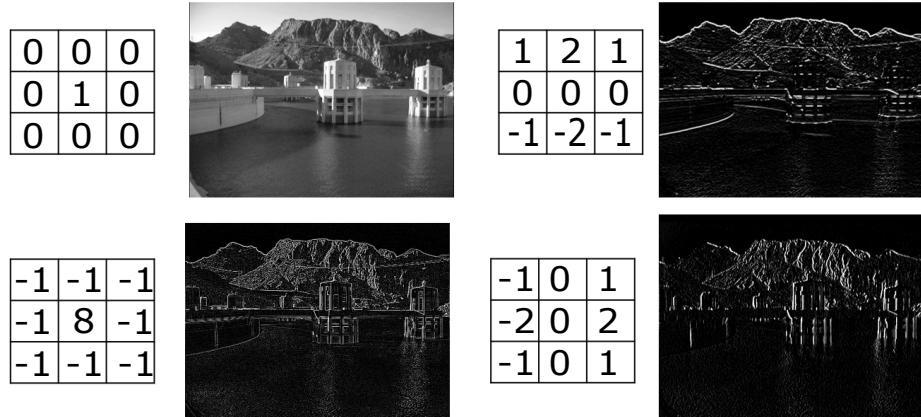
Kde \mathbf{Y} je výstupní matice, \mathbf{X} je vstupní matice a \mathbf{F} je konvoluční jádro. Konvoluční jádro se v konvolučních vrstvách obecně označuje jako filtr. Obecně tedy konvoluční filtr aplikujeme tak, že jej postupně přikládáme na jednotlivé vstupní pixely, ze kterých vzniká výstupní matice. Výhodou konvoluční sítě je to, že podle toho, jaký filtr se aplikuje, tak vzniká několik hladin (replik) obrázku (tzv. feature maps). Poté co se aplikují filtry, tak se ještě aplikuje nelinearity (aktivace funkce viz. 3.1) Výsledné hladiny obrázku v sobě nesou informaci o různých příznacích. v prvních konvolučních vrstvách se může jednat např. o detekci hran 3.2. Při následné kombinaci více konvolučních sítí daných za sebe je možné získat kombinaci filtrů, které dokážou detektovat i složitější věci než jsou hrany.

Pool vrstva

Pool vrstva provede podvzorkování původního obrázku na nějaký menší v určitém poměru. Toto podvzorkování slouží nepřímo k lepší distribuci informací mezi vrstvami a snižuje



Obrázek 3.1: Znázornění 2D konvoluce spolu s reálným příkladem.



Obrázek 3.2: Ukázka aplikace konvolučních filtrů a vniklých hladin obrazu.

výpočetní nároky sítě. Každá vrstva má opět nějaké jádro a jeho definovanou velikost v podobě matice, která určuje, o kolik se zmenší daný prostor. Většinou tato matice bývá o velikosti 2×2 (matice v této podobě znamená, že se vstupní obraz zmenší na polovinu ve vertikální o horizontální ose). Dále je potřeba vždy ještě definovat horizontální a vertikální krok, které většinou bývají o velikosti rozměrů dané matice. Obecně po aplikaci pool vrstvy s jádrem o velikosti $i \times j$, vertikálním krokem v a horizontálním krokem h dostaneme výsledný obrázek o rozměrech $\frac{i}{v} \times \frac{j}{h}$.

Pool vrstev existuje několik různých druhů, které můžou brát různé matematické operace jako výsledek jádra. Já ve své práci používám převážně tzv. *Maxpool* vrstvy, které po aplikaci jádra vezmou jeho maximální hodnotu ze vstupního obrázku a dají ji na výstup 3.3.

Dropout vrstvy

Ve své práci používám také tzv. dropout vrstvy (volně přeloženo jako výpadkové vrstvy). Tyto vrstvy bojují a pomáhají síti proti tzv. přeúčení 3.1. v průběhu učení mají tyto vrstvy

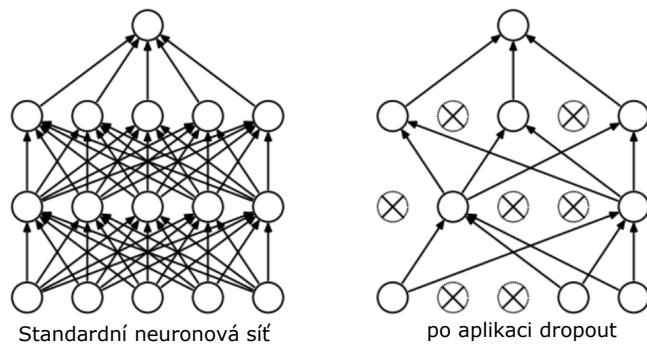
12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

Max-pool
2 x 2

20	30
112	37

Obrázek 3.3: Ukázka MaxPool vrstvy s vnitřním jádrem 2×2 , kde po aplikaci z matice 4×4 vznikne matice 2×2 s maximálními hodnotami.

za úkol způsobovat náhodné „výpadky“ některých neuronů viz.[3.4](#). Sice se tím mírně zpomaluje učení, ale také se zároveň průběžně mění váhy, a to přispívá ke ztrátové funkci. Tento proces by měl zajistit rovnoměrné učení a tím omezit přetrénování sítě.



Obrázek 3.4: Obrázek převrat z [\[4\]](#) a vizualizuje výpadky v neuronových sítích při dropoutu.

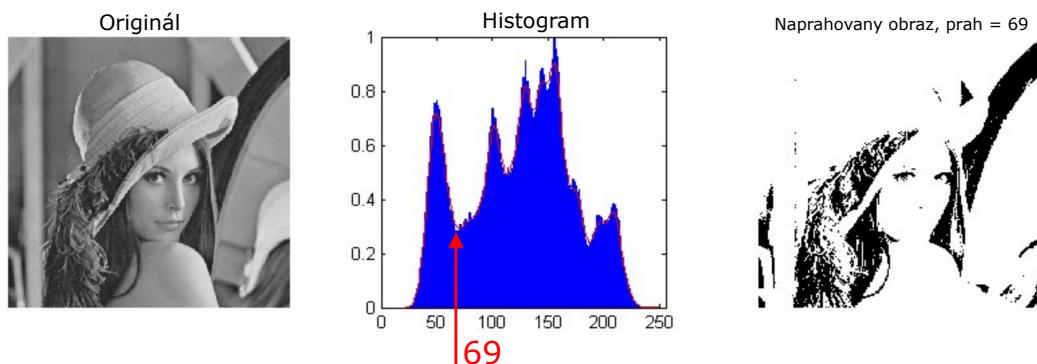
3.3 Segmentace pozadí

Cílem segmentace obrazu je rozlišit, jaká část patří do popředí fotografie a jaká část je nezajímavé pozadí. Po provedení segmentace se každému pixelu vždy přiřadí index, který určuje, zda daný bod patří do popředí nebo pozadí. Tyto informace jsou převážně čerpány z [2].

Segmentace pomocí prahování

Segmentace pomocí prahování funguje tak, že se na obraz aplikuje nějaký prah, který určuje hranici, kdy se pixel v masce označí 1 (popředí) nebo 0 (pozadí). Tato metoda se používá i jako způsob přetvoření barevného obrazu na černobílý. Když se při prahování použije více hodnot prahu, tak lze touto metodou převádět obraz na stupně šedi. u prahování bývá největší problém s výběrem prahu.

Prah se dá vybrat několika způsoby. Jedním z nich je experimentálně tj. provedení několika experimentů na základě kterých se může rozhodnout. Dalším způsobem je z histogramu, při tomto způsobu se sestrojí histogram, na kterém je zobrazena četnost výskytu jednotlivých jasových úrovní a podle něj se určí lokální minimum, které se určí jako prah 3.5. Další možností zjištění a určení prahu je podle statistik daného obrázku, například může být použit průměr nebo median. Případně lze využít nějaké globální znalosti pro obraz, například zelená barva při klíšování plátna.



Obrázek 3.5: Ukázka určení prahu pomocí histogramu.

Segmentace pomocí obrazu z hran

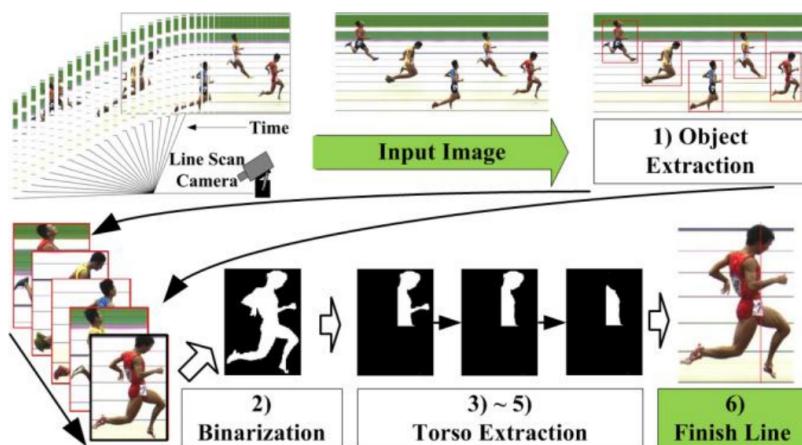
Segmentace pomocí obrazu z hran se využívá za pomocí nějakého už předzpracovaného obrazu, který v sobě nese informaci o hranicích objektů v popředí záběru. Tuto informaci můžeme získat pomocí aplikace různých filtrov, jako se používá u konvolučních vrstev neuronových sítí viz. 3.2 nebo nějaká apriorní informace o tvaru tělesa (např detekce čtverce a podob).

Podmínky pro toto využití jsou však relativně striktní. Je zapotřebí mít v předzpracovaných datech minimální počet chyb, mít nejvyšší možnou přesnost těchto obrysů objektů na popředí (tj. rozdíl mezi skutečnou hranou a nalezenou aby byl minimální), a aby hrany objektů byly jednoznačné.

3.4 Předchozí studie automatizace cílového systému

Rešerší literatury jsem na tuto práci našel pouze jednu studii, která se zabývala stejným tématem *A High Accuracy Automatic Timing Method for Photo Finish Systems* [3] (volně přeloženo jako „Vysoce přesná metoda pro automatické vyhodnocení v cílových systémech“). Tato práce je z roku 2014 a dosahuje přesnosti 82.07% s odchylkou 2ms.

V této práci prvně lokalizovali závodníka na záznamu a vytvořili z něj výřez celého bězce. z těchto výřezů závodníků jsou následně „odřezány“ nepotřebné části těla pro vyhodnocení. Po odstranění těchto částí těla zbyde pouze torso těla, ze kterého se následně určí výsledný čas (viz. 3.6).



Obrázek 3.6: Obrázek znárožňující zpracování cílového záznamu ve studii *A High Accuracy Automatic Timing Method for Photo Finish Systems* [3] (obrázek byl převzat z této studie)

Lokalizace závodníka se provedla pomocí přesné segmentace pozadí, kdy se získaly přesné obrysy závodníku pomocí metody „Canny“ [5]. Tato metoda se snaží hledat hrany na snímku pomocí rozdílu v intenzitě barev. Okolo obrysů se následně udělala konvexní obálka, pomocí které se získaly obdélníkové výřezy celých závodníků. Ve své metodě spolehlám na detekci závodníků knihovnou OpenPose [1]. Díky této knihovně není zapotřebí řešit takové množství falešně pozitivních (anglicky „false positive“) případů. Po lokalizaci běžců výše zmíněné metody se musí řešit případy falešně pozitivně označených „běžců“. Tyto případy se odstraňují dvěma způsoby. Budě podle velikosti (pokud jsou některé označené oblasti značně menší než oblasti běžců) anebo pokud je tato oblast podobně velká jako ostatní, provede se kontrola pomocí spočítání nenulových pixelů (pixelu popředí) a podle počtu se určí, zda se jedná o validní objekt či nikoliv.

Odstranění jednotlivých částí těla závodníka probíhá v několika krocích. První dva kroky jsou jednoduché, kdy pouze rozpůlí běžce po vertikální i horizontální ose a dále se pracuje pouze s pravou horní čtvrtinou běžce. z této vybrané čtvrtiny se následně separují končetiny (ruka a hlava).

Separace ruky je provedena na základě její tloušťky, kdy ruka je značně užší než torso. Tato detekce tloušťky probíhá na základě porovnávání částí záznamu s maticí 7×4 nebo 4×7 . Tyto rozměry jsou dány tloušťkou ruky na zanalyzovaných záznamech, kdy jeden rozměr je pro části směřující směrem vzadu a druhý pro části směřující ve směru dohledu závodníka.

Pro separaci hlavy je zapotřebí prvně najít její střed. Střed hlavy se hledá pomocí provedení transformace vzdálenosti záznamu. Tato transformace spočívá v tom, že okraje záznamu jsou zakresleny světlejší barvou a to se postupně stává tmavší ke středu záznamu. To tedy znamená, že střed záznamu může být určen pomocí maximální hodnoty záznamu. Následně se provede separace, kdy se z nalezeného středu hlavy vyjmé kruh o průměru R , který je určen jako vzdálenost odhadované vrchní části hlavy a odhadnutého středu krku.

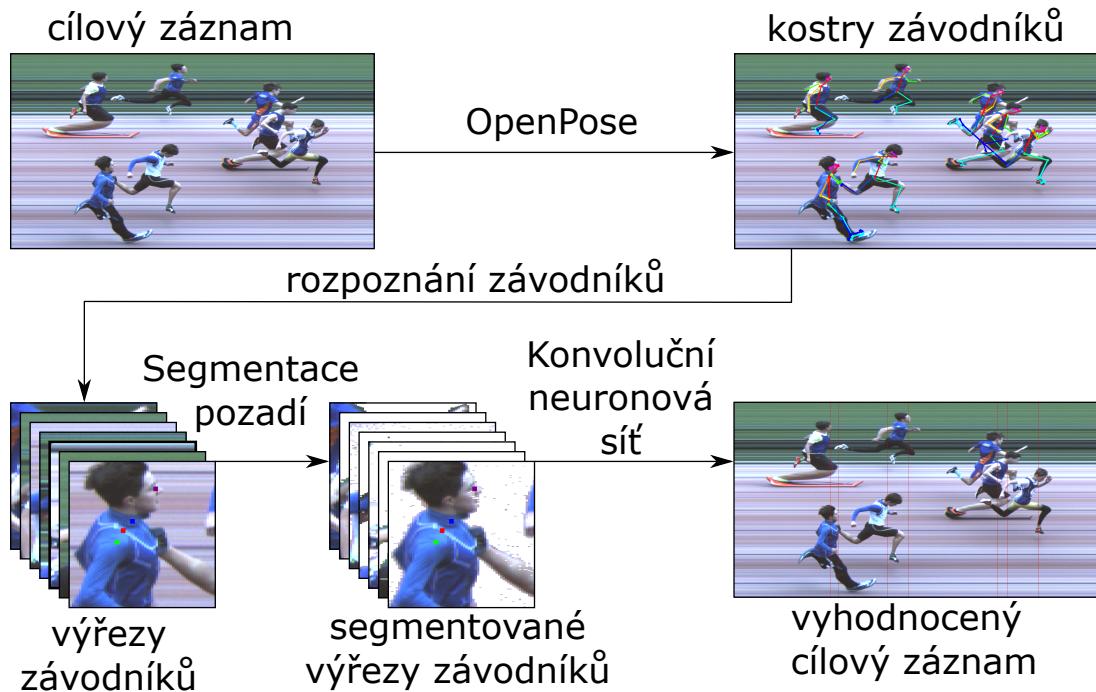
Po separaci paží a hlavy od trupu se následně určí cílový bod jako bod trupu, který je nejbližše pravé straně záznamu (straně doběhu). Ve své metodě neřeším problémy, které vznikají při separování končetin, a místo toho celé záznamy vyhodnocuji pomocí neuronové sítě. Tato metoda je dle mého mínění výhodnější. Nemá problémy s detekcí více závodníků najednou (ve zmíněné studii jsem nenašel žádné záznamy na toto téma, z textu vyplývá, že se zabývali pouze detekcí doběhu jednotlivců). Samotné zpracování je dle mého názoru výpočetně méně náročné (tuto svoji teorii bohužel nemohu porovnat).

Ve zmiňované studii měli v datové sadě 390 cílových záznamů, které všechny byly z nejvyšších atletických závodů, pořádaných v Číně. Lze proto hovořit o tom, že měli „ideální podmínky“ (viz 2.4) pro automatické vyhodnocování záznamů. Proto je možné, že jejich metoda vyhodnocení nebude fungovat na cílové záznamy z menších soutěží, kde je v záznamech větší šum v pozadí.

Kapitola 4

Metoda pro automatizaci cílového systému

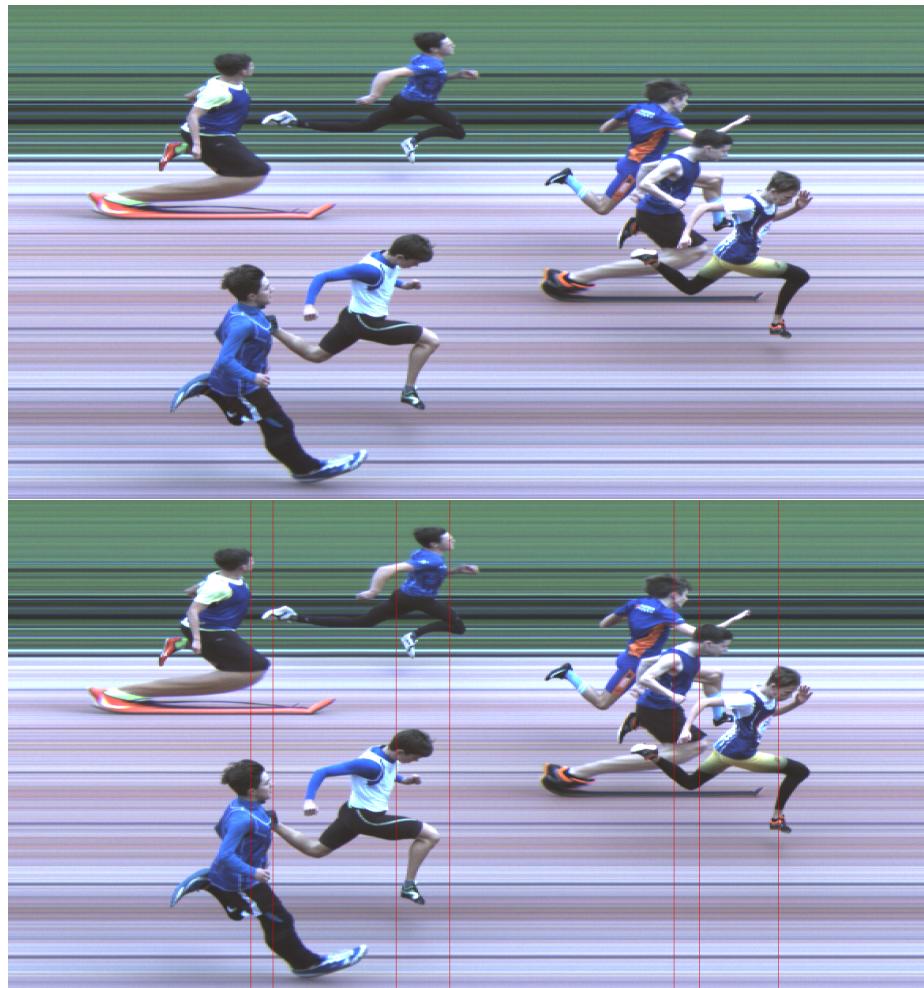
V této kapitole se věnuji popisu navrhnuté metody. Způsob zpracování mojí metody začíná u knihovny OpenPose [1]. Tato knihovna se zabývá detekcí postav na obrázku. Po detekci postav přichází vytvoření jednotlivých výřezů a segmentace pozadí pro jednotlivé výřezy. Jednotlivé výřezy se následně vyhodnotí pomocí regresní konvoluční neuronové sítě.



Obrázek 4.1: Blokové schéma využívané metody.

4.1 Zpracování cílového záznamu

Jednotlivé cílové záznamy jsou 2D snímky z řádkové kamery (viz. 2.2), které je zapotřebí správně vyhodnotit. na obrázcích níže 4.2 lze vidět příklad jednoho z cílových záznamů. Horní obrázek je původní „vstupní“ záznam, který přijde rozhodčímu do počítače na vyhodnocení. Spodní obrázek je již vyhodnocený „výstupní“ záznam, ze kterého se určují doběhnuté časy.

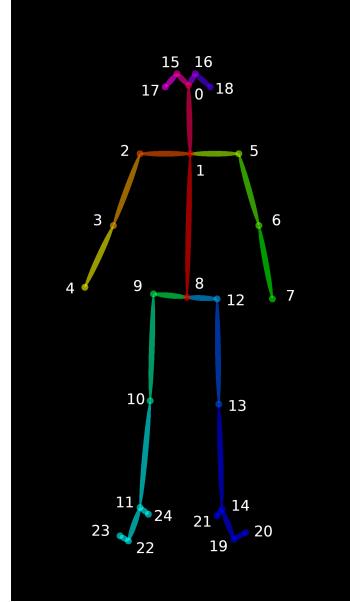


Obrázek 4.2: Příklad záznamu cílové kamery. Horní obrázek je původní výstup z cílové kamery a spodní je již vyhodnocený obrázek rozhodčím (červené čáry), který je referenčním řešením pro naše ukázky.

4.2 Detekce postav

Detekci postav provádím pomocí knihovny OpenPose [1], která zajišťuje detekci lidských postav z obrázku. OpenPose je schopná detektovat více než 135 klíčových bodů. Jelikož však OpenPose využívám pouze jako prostředek pro detekci postav a několika základních částí, tak potřebuji a generuji pouze 25 bodů podle modelu „MODEL_25“ (viz. 4.3) pro zmenšení nároků na výpočetní výkon. na tomto modelu mě zajímají následující body: 0 (hlava),

1 (trup), 2 (levé rameno) a 5 (pravé rameno). Podle těchto bodů lze totiž určit náklon a natočení těla v oblasti důležité na vyhodnocení a mohou pomoci budoucí konvoluční neuronové síti ve vyhodnocování.



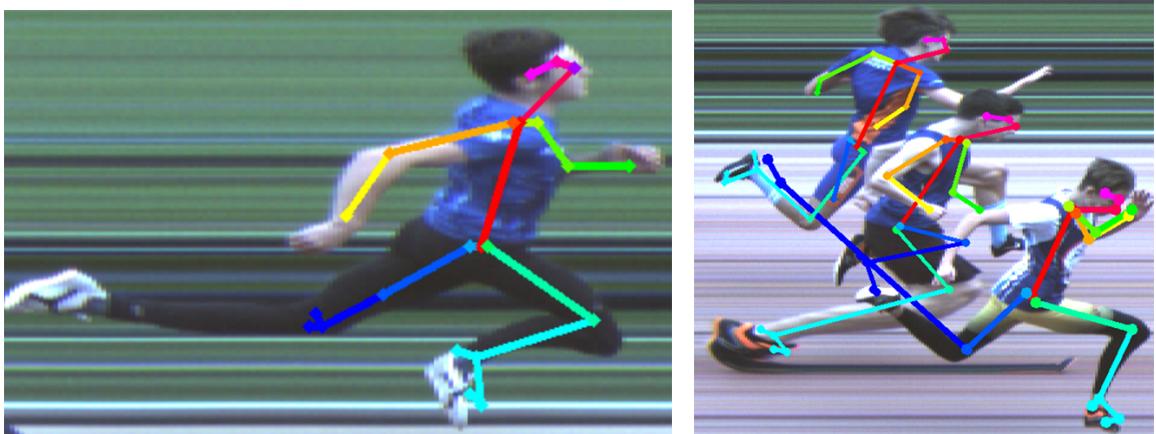
Obrázek 4.3: Ukázka modelu „BODY_25“ z knihovny OpenPose [1]

Po zpracování touto knihovnou tedy dostáváme seznam bodů pro jednotlivé závodníky. Vizualizace těchto vygenerovaných bodů na mém příkladu může vypadat následovně 4.4. na této vizualizaci jsou vidět barevné kostry dle modelu „BODY_25“.



Obrázek 4.4: Ukázka vizualizace výstupu z knihovny OpenPose [1] za využití modelu „BODY_25“

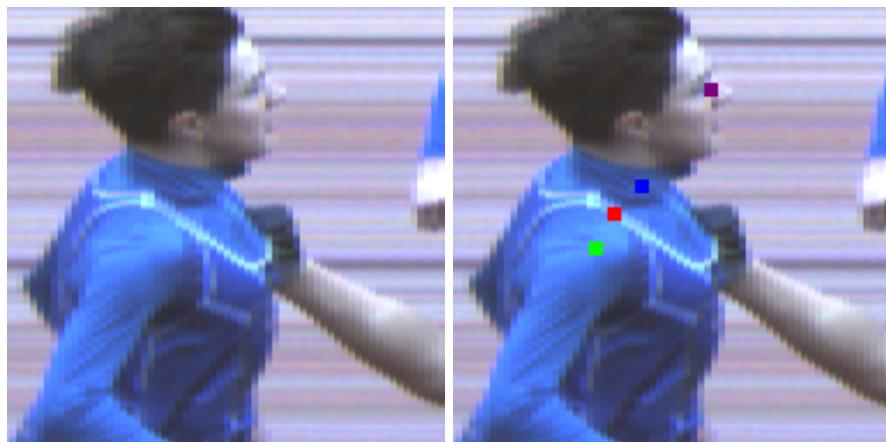
Tato již předtrénovaná neuronová síť je místy zmatená ze zkreslení, které nastává při pořizování cílového záznamu (viz 2.3). Tato zkreslení se promítají převážně na dolní končetiny (viz 4.5) a ojediněle na horní, ale v takových případech ramena zůstávají, kde mají, takže tyto deformace kostry jsou v mém případě zanedbatelné.



Obrázek 4.5: Ukázka deformací v kostře pro můj příklad vstupu. na levém příkladu je vidět, že si knihovna neporadila s mírnou neformací nohy, která se nepřizoreně natáhla. a na pravém výřezu je vidět opak, kdy pro závodníka vpravo dole byla pravá noha vyhodnocena i v délce sousedících závodníků.

4.3 Vytvoření výřezů závodníků

Dalším krokem mého řešení je vytvoření výřezů kolem závodníků, kde jako centrální bod každého závodníka beru jeho hrudník, který je většinou detekován s největší jistotou. Pro každý bod hrudníku tedy určím jeden výřez, do kterého dále zanesu informace o bodech: hrudníku, hlavy a ramen. Tuto informaci do jednotlivých výřezů zanesu pomocí barevných čtverečků, kdy hrudník je červený, hlava je fialová, levé rameno modré a pravé je zelené 4.6. Tuto informaci zanáším přímo do výřezů, aby napomohla neuronové síti v učení vyhodnocování jednotlivých výřezů. Tyto body jsou vždy zakresleny pouze pro aktuálně vybraného běžce ve výřezu, takže by mely i pomoci při vyhodnocování běžců, kteří jsou v zákrytu pro identifikaci hledaného běžce.



Obrázek 4.6: Vizualizace zobrazení jednotlivých bodů, z detekce těla.

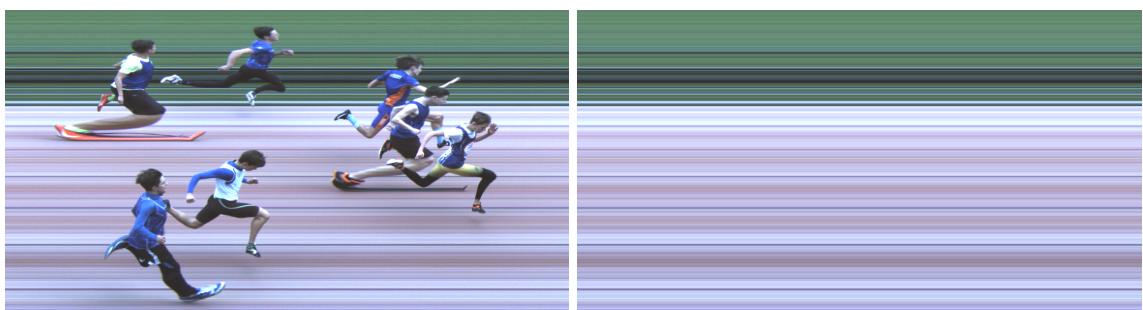
Při vytváření výřezů je potřeba také zohlednit směr běhu závodníků. Zpravidla bývají cílové záznamy pořizovány z vnější strany oválu a tudíž všichni závodníci směřují zleva doprava (stejným směrem jako je na ukázkách). Občas se ale může stát, že podmínky

stadionu nedovolují zaznamenávání z vnější strany oválu, a musí se tedy záznamy pořizovat ze strany vnitřní. v takovém případě závodníci směřují zprava doleva a je zapotřebí je otočit¹ kolem vertikální osy a provézt tzv. „zrcadlení“ (obrácení horizontální osy).

4.4 Segmentace

Pro segmentaci pozadí jsem využil vlastnosti rádkové kamery, která říká, že nehybné předměty se projeví jako statická barva (více 2.2). a jelikož nehybné předměty na cílovém záznamu jsou brány jako pozadí, využil jsem této vlastnosti pro svou metodu segmentace.

Základem mé metody je segmentace pomocí prahování (viz 3.3), kdy pomocí znalosti statistické (v mém případě medianu) určím prah pro jednotlivé rádky kamery a tím vytvořím „průměrné“ pozadí cílového záznamu. na obrázcích níže 4.7 je vidět porovnání průměrného pozadí (vytvořeného pomocí mediánu daného obrázku) vůči původnímu obrázku.



Obrázek 4.7: Vizualizace vytvoření „průměrného“ pozadí.

Jelikož však na reálném pozadí figuruje šum ze stínů závodníků a šum, který vzniká v oblasti zeleně za první dráhou (v tomto případě je první dráha na vnitřní straně oválu a záznam je pořízen z vnější). Informace o šumu může poskytnout jednoduché odečtení obrázků od sebe 4.8.



Obrázek 4.8: Vizualizace odečtu pozadí vytvořeného pomocí medianu od originálního obrázku. Bílá barva zde vyjadřuje příslušnost k popředí a černá k pozadí obrázku.

Tento problém lze odstranit jednoduchou aplikací nějakého rozptylu kolem prahu. Tímto rozptylem dělám z pevného prahu (který určí, zda se jedná o pozadí či nikoliv) množinu o modulárním rozsahu pro každý řádek. Do výsledného rozsahu zakomponovávám i poměr

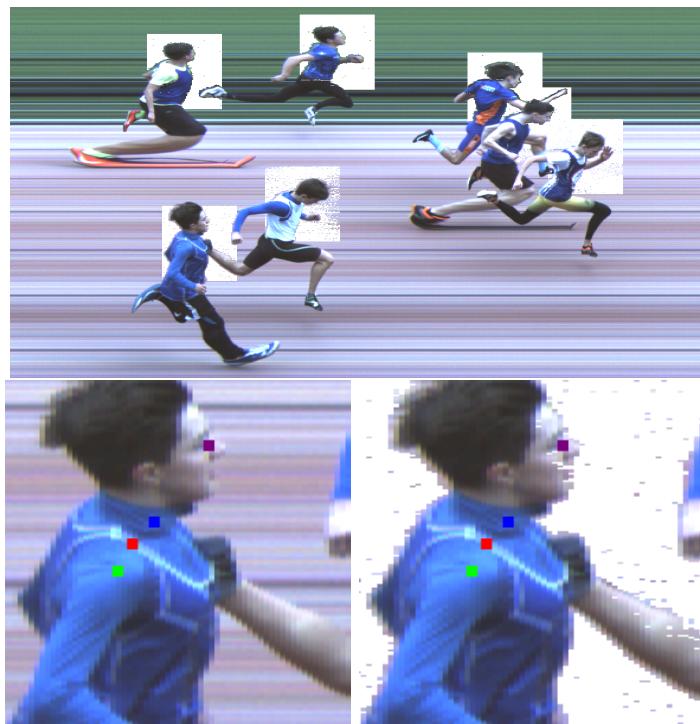
¹Jelikož je tato proměnná stejná po celý chod závodu, tak jsem se rozhodl neimplementovat metodu na rozpoznávání směru běhu a otočení je směr běhu je brán jako vstupní parametr programu.

mezi jednotlivými barevnými složkami tak, aby byl rozptyl flexibilnější ve své hlavní barevné složce, ale méně flexibilní ve vedlejších. Příklad: *Mějme vypočítaný median pro řádek s hodnotami [25, 125, 50] (zelenou), tím pádem jeho poměr barevných složek je 1:5:2. To znamená, že rozptyl pro červenou složku bude $\frac{1}{8}$ z původního rozptylu, pro zelenou $\frac{5}{8}$ z původního rozptylu a pro modrou $\frac{1}{4}$ z původního rozptylu.* viz. 4.9.



Obrázek 4.9: Vizualizace aplikace metody pro segmentaci pozadí.

Pro lepší optimalizaci samotné segmentace pozadí jsem se rozhodl neprovádět segmentaci pro celý obrázek, ale pouze pro oblasti, ze kterých se provádí výřezy viz 4.10.



Obrázek 4.10: Vizualizace kompletní metody segmentace pozadí, která je ve vrchní části aplikována na celý obrázek a ve spodní je porovnání aplikace na výřez před a po segmentaci.

4.5 Regresní konvoluční neuronová síť

Protože se při používání konvoluční neuronové sítě využívají tzv. pool vrstvy, které zmenšují vstupní obraz (více 3.2), a mé výřezy z datasetu byly o velikosti 190×190 (které se špatně půlí tak, aby zůstalo celé číslo), každý výřez jsem před zařazením do datasetu zmenšil na velikost 160×160 .

Moje výsledná architektura neuronové sítě, která byla vyhodnocena jako nejpřesnější, obsahuje konvoluční vrstvy, aktivacní funkci ReLu, MaxPool vrstvy, dropout vrstvy a lineární vrstvy na konci.

Moje neuronová síť neměla k dispozici na učení moc velký dataset, následkem čeho tak často docházelo k přeúčení (tj. neuronová síť se až moc dobře naučila vyhodnocovat trénovací data). Proto jsem aplikoval dropout vrstvy s pravděpodobností 25% mezi konvoluční vrstvy a první lineární vrstvu a mezi lineární jsem vložil dropout vrstvu s pravděpodobností 50%. Tyto pravděpodobnosti se ukázaly být jako nejlepší z různých možností dosazení dropout vrstev na základě testování. Využití dropout by celkově mělo zajistit, aby na sobě nebyly jednotlivé vrstvy tolík závislé.

Pro učení jsem používal trénovací sadu o velikosti 5314 výřezů jednotlivých závodníků (viz. 5.2). Tato trénovací sada určitě nebyla dostatečně veliká, ale s ohledem na aktuální situaci byla to nejlepší, co se mi podařilo nashromáždit. K učení jsem používal „learning rate“ (volně přeloženo jako učící koeficient) o velikosti 0.001. Jako ztrátovou funkci jsem využil „mean square error“, která určuje přesnost odhadů jako střední hodnoty (průměr) druhých mocnin rozdílů mezi odhadem a skutečností (anotací 4.1).

$$MSE = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x}_i)^2 \quad (4.1)$$

Kde výsledná odchylka je **MSE**, **n** je počet prvků, x_i je predikce neuronové sítě a \bar{x}_i je anotace daného data.

Kapitola 5

Implementace a experimenty

V této kapitole se zabývám způsobem a metodami implementace výše zmíněné metody, tvorbou datové sady pro trénování neuronové sítě a provedenými experimenty.

Pro práci s obrázky využívám knihovny opencv (cv2 pro python) a pro zpracování obrázků v podobě matic poté knihovnu numpy. Pro následné vytváření datasetu a zpracování neuronových sítí jsem využil knihovny pytorch a torchvision¹.

5.1 Implementace

OpenPose

Program OpenPose využívám v podobě knihovny přeložené pro Python. Při pracování s touto knihovnou je zapotřebí zohlednit, zda systém, na kterém má knihovna fungovat, je Windows nebo Linux. Pro Linux je zapotřebí pouze přidat cestu ke složce *build/python* do systémových cest, ale u Windows je navíc zapotřebí přidat cesty do složek *release* a *bin* do systémové proměnné PATH. Dále je práce s touto knihovnou z velké části inspirována manuálem OpenPose [1].

Data z OpenPose poté ukládám ve formátu JSON, který lze jednoduše dále parsovovat a uložit. Jednotlivé body jsou zde uloženy ve formátu $[x, y, P]$, kde x a y jsou souřadnice v obrázku a P je jistota vyhodnocení, že daný bod skutečně je na daném místě a uvádí se v rozsahu $<0, 1>$.

Vytváření výřezů a segmentace

Pro vytváření jednotlivých výřezů je zapotřebí cílová fotografie (stejná, co se dává jako vstup zpracování OpenPose) a výše zmíněný soubor JSON. Pro každého závodníka je následně vytvořen výřez o velikosti 190×190 , který je určen podle bodu hrudníku každého závodníka. Bod hrudníku je následně posunut o 25 pixelů doprava a 5 pixelů dolů tak, aby v oblasti výřezu byla nejdůležitější část doběhu závodníka. Pokud je pořízený záznam doběhu obrácený (závodník směřuje zleva doprava), tak je bod hrudníku posunut na horizontální obráceně tj. o 25 pixelu doleva. Cílový záznam je zde načten pomocí knihovny opencv, ze které se načte obraz jako pole numpy pro lepší následnou práci pomocí této knihovny. Jelikož knihovna opencv načítá obrazové záznamy pomocí barevného modelu BGR, je zapotřebí jej přetrasformovat do barevného modelu RGB pro lepší práci.

¹obě tyto knihovny jsou využívány ve verzi kompatibilní s CUDA

Pro každý výrez je následně provedena segmentace pozadí, kdy se prvně určí vektor o velikosti výšky záznamu, kde v každém bodu je median z barevných hodnot, který se určí pomocí knihovny numpy. Následná samotná segmentace se provádí pomocí postupného procházení a porovnávání jednotlivých pixelů s pixelem v tomto vektoru, který má stejnou vertikální souřadnici (viz 4.4).

Konvoluční neuronová síť

Moje výsledná architektura neuronové sítě, která byla vyhodnocena jako nejpřesnější, obsahuje konvoluční vrstvy, aktivační funkci ReLu, MaxPool vrstvy, dropout vrstvy a linearní vrstvy na konci. Konvoluční vrstvy jsou čtyři a dosahuje se v nich maximální velikosti 64 kanálů (neboli „features“), každá z těchto vrstev má velikost jádra 3×3 , padding jedna a posun také o jedna. Po každé konvoluční vrstvě se použije aktivační funkce ReLu a pool vrstva o velikosti 2×2 a posunem také 2×2 . Následně se aplikují dvě plně propojené vrstvy, které se z „obrázku“ o rozměrech $10 \times 10 \times 64$ (neboli „tensoru“ s počtem prvků 6400) pokusí predikovat výsledek v rozsahu $<0,1>$, který znamená posun na horizontální ose vůči výrezu.

Samotné učení neuronové sítě bere relativně veliké množství výpočetního výkonu tak, aby bylo trénování provedeno v rozumném čase. Proto jsem pro trénování své neuronové sítě využil služby Google Colab, která bezplatně poskytuje obecný výpočetní výkon na grafických kartách pro Python notebooky, kterého jsem doma nemohl dosáhnout.

Na ověření přesnosti různých architektur neuronových sítí jsem využíval funkci, která počítala odchylku od správné hodnoty jako absolutní velikost rozdílu anotace a predikce daného výrezu. Tuto odchylku počítám z požadované přesnosti v ms tak, že požadovanou přesnost v ms vynásobím skutečnou velikostí jednoho bodu na obrázku (viz 2.3) pro získání počtu zasáhnutých obrazových bodů na cílovém záznamu a celé podélím velikostí výrezu pro získání hodnoty v rozsahu $<0,1>$.

$$Odchylka = \frac{Pozadovana\ presnost \times velikost\ pixelu}{velikost\ výrezu} \quad (5.1)$$

5.2 Datová sada

Pro řešenou problematiku neexistuje žádný veřejný dataset a proto jsem si ho musel vytvořit sám. Při sbírání datasetu bohužel nastaly problémy. Původní plán byl získat fotky z nejvyšších atletických soutěží, které se v České republice pořádají, ale vzhledem k problémům v souvislosti s SARS-CoV-2 bylo jejich získání problematické a nakonec se nezdářilo. Byl problém se spojit s firmou OnlineSystem, která tyto záležitosti v ČR realizuje. Takže veškeré záznamy mého datasetu byly nakonec z brněnského oddílu ČibaSport Athletics (pořízené v letech 2014 - 2020).

Zpracování

Veškeré záznamy do datasetu byly potřeba prvně vyexportovat z formátu kamery TimeTronics, kde jsou data ukládána do formátu .mf4. Tento formát je ale upravený a nemá stejné kódování jako klasické .mf4 formáty, které patří do „rodiny“ MDF souborů, a po kontaktování společnosti se mi bohužel nedostalo žádné odpovědi. Takže jsem za využití programu MacFinish vyexportoval danou fotografii do formátu .png a s ním nadále pra-

coval. Tímto exportem jsem bohužel ztratil informaci o časových značkách pro jednotlivé sloupce záznamu.

Každá fotografie byla poté vyexportována ve dvou možnostech, jedna normální pro zpracování a jedna již vyhodnocená, ze které se poté získávaly anotace. z vyexportovaného cílového záznamu jsem poté získal body hrudníku, hlavy a ramen, podle kterých se vytvářely následné výřezy.

Technické parametry datasetu

Můj dataset obsahuje záznamy z 48 závodů, které dohromady obsahují 1649 fotek, z kterých je vytvořeno celkově 5314 výřezů. Do testovací sady jsem dal 1082 výřezů ze 334 celých fotek, které jsou ze 12 různých závodů. v trénovací sadě je zbytek, tj. 1315 fotek ze 36 různých závodů, která dávají dohromady 4232 unikátních výřezů².

Každý jeden výřez jsem poté z původní velikosti 190×190 zmenšil na velikost 160×160 pro lepší zpracování v neuronové síti. Výřezy jsou ve formátu PNG s bitovou hloubkou 24. Anotace pro jednotlivé výřezy jsou poté uvedeny v rozmezí $<0,1>$.

5.3 Vytváření datové sady

K tvorbě datové sady byly využity cílové záznamy atletického oddílu ČibaSport Brno. Bohužel, konečný počet výřezů v datasetu není moc velký a i to může být důvod přetrénování neuronových sítí a menší přesnosti, než jsem na počátku očekával.

Anotace pro jednotlivé záznamy byly získány pomocí již vyhodnoceného záznamu. na tomto záznamu byly v místech vyhodnocení vždy červené čáry (vždy s RGB hodnotou [220,10,10]) přes celou výšku záznamu, takže pro ně bylo jednoduché určit pozici na horizontální ose. Tyto výřezy byly poté vloženy do seznamu „možných výsledků“. Při vytváření výřezů jsem vždy zjistil, kolik pozic ze seznamu možných výsledků se nachází v oblasti daného výřezu a z jejich pozic poté udělal seznam možných výsledků (anotací) pro daný výřez. Pro jednotlivé výřezy byly anotace uloženy v JSON souboru vždy se stejným jménem.

Pro každý záznam byly postupně vytvořeny výřezy (viz 4.3), u kterých jsem navíc dělal ruční augmentaci (umělé navýšení počtu dat v datasetu) posunu a výřezu. u každého výřezu bylo vždy náhodně určeno, jestli se provede augmentace posunu nebo výřezu, u augmentace posunu byl náhodně určen jeden z osmi různých směrů a u výřezu bylo náhodně určeno, zda se provede výřez větší a následně se zmenší na určenou velikost 190×190 , nebo naopak. Jako zdroje k porozumění augmentace a její aplikace mi posloužila stránka [6].

Jelikož některé cílové záznamy byly „foceny“ z vnějšího a některé z vnitřního okraje oválu, tak nejsou všechny otočené stejným směrem. Jelikož však moje datová sada nebyla dostatečně velká, rozhodl jsem se, že do ní nebudu zakomponovávat data z obou směrů doběhu. Ručně jsem stanovil směr pro jednotlivé závody a podle toho otáčel příslušné záznamy a k nim příslušící anotace. Anotace nebylo možné otáčet stejným způsobem jako výřezy, ale bylo nutné je zrcadlit podle středu každého výřezu.

Při vytváření výřezů určených pro dataset byly výřezy rozděleny do 3 skupin:

1. „Samostatné“ - tato skupina výřezů obsahovala pouze výřezy, u kterých byla anotace jednoznačná. na každém výřezu byl pouze jeden běžec s jednou anotací.

²veškeré tyto údaje jsou uvedeny bez augmentace

2. „Několikanásobné“ - tyto výřezy bylo potřeba ručně analyzovat a rozhodnout, která z možných anotací patří danému bězci. na těchto výřezech bylo více možných anotací, protože doběhlo více závodníku zároveň.
3. „Augmetance“ - do této skupiny patřily všechny augmentace výřezů, které byly provedeny. Tato skupina sloužila pouze pro usnadnění ruční kontroly výřezů.

Skupiny *samostatné* a *několikanásobné* se poté dále dělily na skupiny „ke kontrole“ a „správné“. Do těchto podskupin se dělily na základě „jistoty“ knihovny OpenPose [1], že v dané oblasti skutečně je nějaký závodník či nikoliv.

Po vytvoření všech výřezů do datasetu jsem ručně prošel všechny výřezy a „pročistil“ je. Výsledné výřezy jsem rozdělil do trénovací a testovací sady, kde testovací obsahuje 20.36% ze všech výřezů.

Jednotlivé sady jsem poté připravil pro použití v neuronové síti. Každý výřez byl zmenšen na velikost 160×160 a pomocí knihovny PIL načten v podobě obrázku. Tato knihovna je dobře propojena s knihovnou PyTorch, pomocí které jsem pracoval s neuronovými sítěmi. Každý jednotlivý údaj byl poté uložen v podobě tuple (obrázek výřezu, anotace) do seznamu. z tohoto seznamu jsem poté vytvořil samotné datasety, které jsem uložil pomocí knihoven pickle a gzip do formátu *.pkl.gz*.

Pro následné načtení datové sady do podoby zpracovatelné knihovny PyTorch jsem vytvořil třídu „AthleticDataset“, které při inicializaci načte pomocí knihovny pandas výše zmíněná data a vytvoří se svůj *DataFrame*. Tuto knihovnu jsem zde zvolil pro pohodlnou a efektivní práci s tabulkovými daty. Následně v metodě *getitem* (vrácení položky na daném indexu) vracím tuple (tensor obrázku, tensor anotace) tak, že na obrázek nejprve aplikuju transformace. Transformace, které provádím, jsou náhodná barevná augmentace a převod na tensor. Pro neuronovou síť poté použiji třídu *DataLoader* z knihovny PyTorch, aby byl možný přístup neuronové sítě k datasetu.

5.4 Experimenty

V této části se chci zaměřit na provedené experimenty mojí metody nad mnou poskládaným datasetem. Experimenty se skládají z několika hlavních částí:

- Přesnost detekce závodníků na cílové fotografií.
- testování různých metod určení prahu při segmentaci pozadí.
- Přesnost vyhodnocování knovoluční neuronové sítě.

Přesnost detekce závodníků

Přesnost detekce závodníků jsem měřil na základě porovnání výsledků knihovny OpenPose vůči vyhodnocené fotografii. z vyhodnocené fotografie jsem vždy získal počet vyhodnocených závodníků, pomocí detekce vertikálních čar určujících čas v záznamu. Toto číslo jsem následně porovnával s počtem osob ze souboru JSON, který je na výstupu knihovny OpenPose. Počet závodníků v tomto souboru jsem určil podle počtu úspěšně detekovaných bodů hrudníku. Postupně jsem také snižoval prah jistoty u bodu hrudníku pro započítání platného závodníka.

práh jistoty	pravděpodobnost
0 %	77.70 %
10 %	74.81 %
20 %	74.06 %
30 %	73.11 %
40 %	71.80 %
50 %	70.33 %
60 %	68.30 %
70 %	65.29 %
80 %	56.06 %
90 %	22.63 %
100 %	0.00 %

Moje výsledná jistota pro detekci závodníků je tedy podobná prahu 50 %, ale dosažena profiltrovanou datovou sadou výřezů k skutečnému počtu závodníků. Skutečný počet závodníků v mé datové sadě bylo 7490 a záznamů v datové sadě 5314, což dává přesnost 70.94 %.

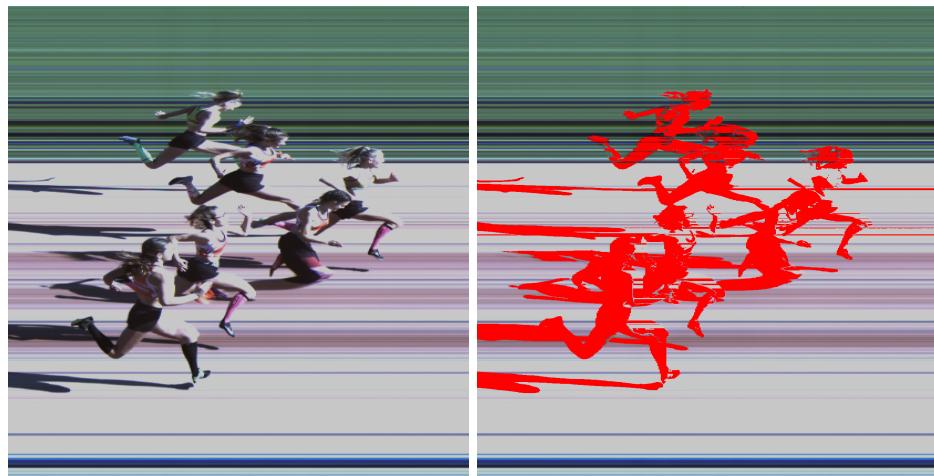
Segmentace pozadí

Pro segmentaci pozadí jsem se držel převážně původní myšlenky (tj. segmentace pomocí prahování), kdy jsem zkoušel prvně prahovat pomocí průměrného pixelu na řádek. v tomto případě však nastával problém v momentech, kdy se barva na dráze převážně stavěla z 2 hlavních barevných složek. Tento stav mohl nastat v případech, kdy při doběhu do cíle svítilo slunce, a tím pádem na doběhu byl převážně „přepálený“ závodník a jeho stín [5.1](#).



Obrázek 5.1: Vizualizace metody segmentace na základě **průměru** barvy na řádku. Popředí je zde zvýrazněno červenou barvou pro lepší přehled.

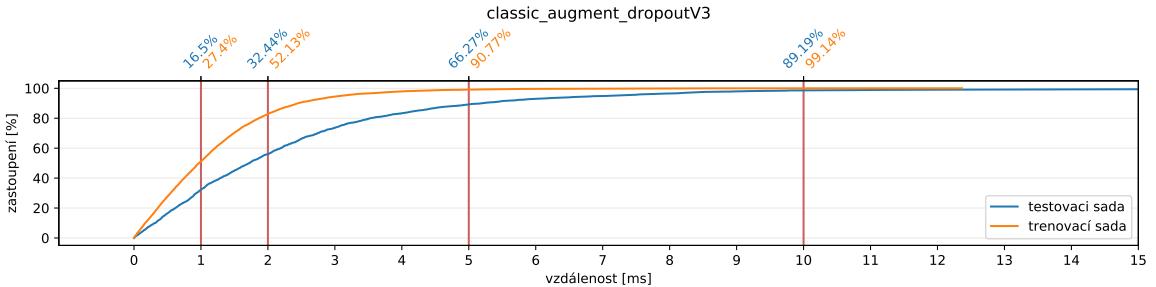
Dalším krokem v této metodě bylo nahrazení průměru za median. Tento krok vedl k znatelnému zlepšení z těchto extrémních světelných podmínek, ale zároveň rpo klasické fotografie nechal výsledky stejné 5.2.



Obrázek 5.2: Vizualizace metody segmentace na základě **medianu** barvy na řádku. Popředí je zde zvýrazněno červenou barvou pro lepší přehled.

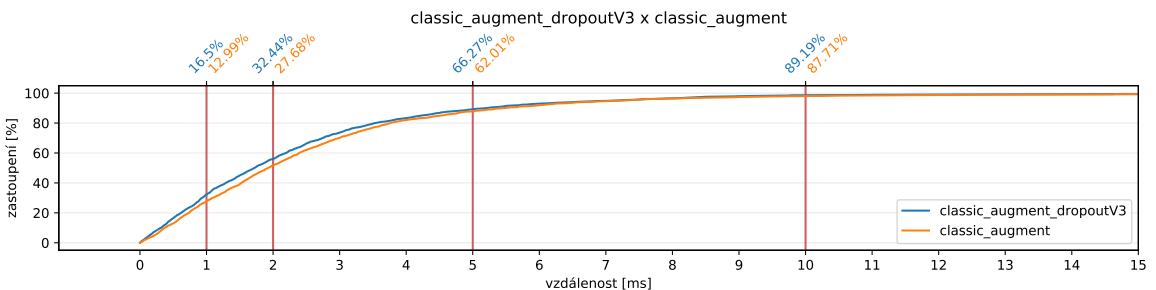
Přesnost konvoluční neuronové sítě

Přesnost vyhodnocení konvoluční neuronové sítě měřím za pomocí tzv. CDF grafů (Cumulative Distribution Function, tj. kumulativní distribuční funkce), kdy na ose X je vzdálenost od anotace v ms a na ose Y je procentuální zastoupení záznamů, které se vyhodnotily do dané přesnosti. V grafu jsou také zvýražněny 4 hranice (1ms, 2ms, 5ms a 10ms), na kterých jsem porovnával přesnost jednotlivých druhů neuronových sítí. Tyto hranice jsou v grafu zvýrazněny červenými čarami. Příklad vykresleného grafu pro "vítěznou" neuronovou síť, která se v mých testech objevuje pod označením **classic_augment_dropoutV3** 5.3.



Obrázek 5.3: Graf přesnosti finální architektury konvoluční neuronové sítě.

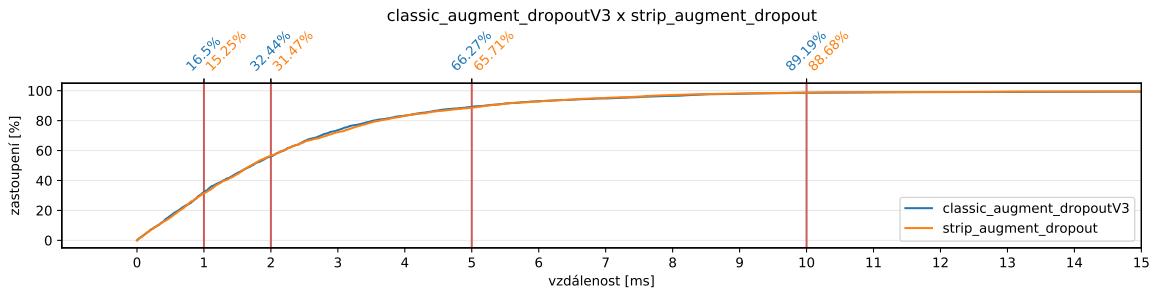
Základní architektura mé neuronové sítě bez dropout vrstev měla celkově horší výsledky, než verze s dropout vrstvami. Porovnání „původní“ vůči již upravené je vidět zde 5.4:



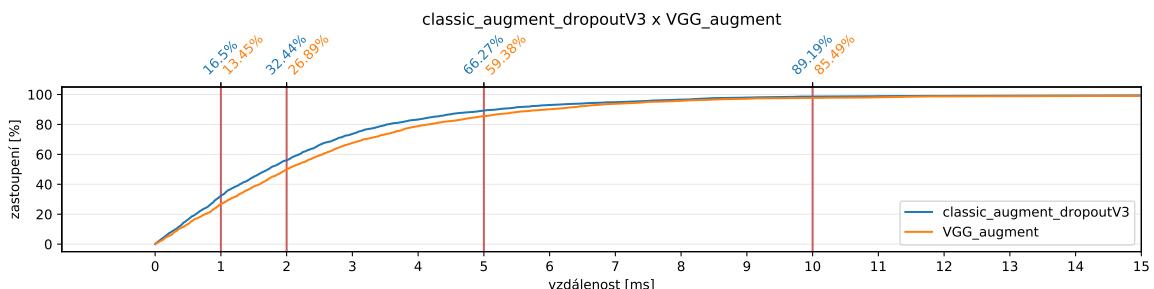
Obrázek 5.4: Porovnání přesnosti sítí „classic_augment“ (základní tvar sítě) a vybrané architektury „classic_augment_dropoutV3“

Mezi další pokusy při využití různých architektur patří pokusy s verzí výše zmíněné neuronové sítě (viz 5.1), ale za použití převážně vertikálního poolingu, kdy jsem po aplikaci konvolučních vrstev dostal výstup o rozměrech $60 \times 2 \times 64$. Myšlenka stojící za touto verzí neuronové sítě se týkala omezení ztráty informací na horizontální ose o „poze“ závodníka při doběhu. Tato síť při testování vyhodnocuje s menší přesností. Podle mě má tato síť menší přesnost, protože při větším zmenšování na vertikální ose se ztratí hranice mezi končetinami (ruce, nohy a hlava) a trupem, kterou by se měla neuronová síť naučit detektovat. Rozdíl ve variantě této verze s vybranou není veliký. Jejich porovnání je vidět zde 5.5:

Další testovanou architekturou neuronové sítě je předtrénovaná síť VGGNetSeq, ze které jsem využil prvních 10 vrstev. Za tyto vrstvy jsem následně vložil další své 2 konvoluční a mezi ně aplikoval dodatečný pooling. Tato varianta dosahovala podobných výsledků jako první zmíněná, ale kvůli větší vnitřní velikosti je značně pomalejší a proto pro mě nevyužitelná. Příklad porovnání s vybranou architekturou je vidět zde 5.6:

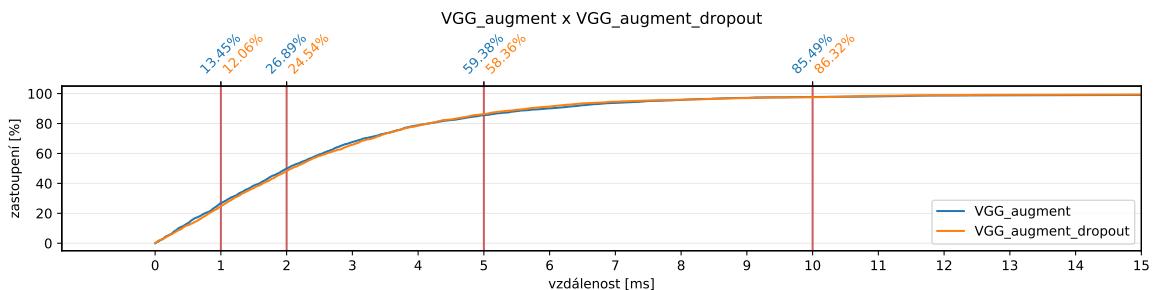


Obrázek 5.5: Porovnání přesnosti sítí „strip_augment_dropout“ (vertikální pooling) a vybrané architektury „classic_augment_dropoutV3“



Obrázek 5.6: Porovnání přesnosti sítí „VGG_augment“ (založeno na VGGNetSeg bez dropout vrstev) a vybrané architektury „classic_augment_dropoutV3“

Pro síť založenou na architektuře VGGNetSeq jsem se pokoušel udělat i variantu s dropout vrstvami. Při této modifikaci však nastalo zhoršení vyhodnocené přesnosti. Jejich porovnání lze vidět zde 5.7.



Obrázek 5.7: Porovnání přesnosti sítí „VGG_augment“ (založeno na VGGNetSeg bez dropout vrstev) a „VGG_augment_dropout“ (založeno na CGGNetSeg bez dropout vrstev)

Experimentoval jsem i s různými „hloubkami“ konvolučních vrstev v neuronových sítích, a to konkrétně s postupným zvýšením až na 128 nebo 256 úrovní. Tyto pokusy však vedly pouze k rychlejšímu přeúčení neuronové sítě na datech. Dále jsem experimentoval s dosazením dropout vrstev do různých částí neuronové sítě. v prvních verzích jsem zkoušel dosadit dropout vrstvy pouze mezi lineární vrstvy na konci neuronové sítě. v dalších experimentech jsem se pokoušel postupně dosazovat dropout vrstvy na co nejvíce pozic. z těchto experimentů mi vyšla nejlépe výše zmíněná architektura.

5.5 Diskuze

Dalším krokem rozšíření této práce, by mohlo být plné propojení s cílovým systémem. v případě tohoto propojení by systém zvládl vyprodukovavat celé výsledky včetně časových značek. Toto rozšíření bohužel záleží na spolupráci s firmou.

Další možnost rozšíření může být také lepší architektura neuronové sítě. Mnou vybraná architektura je relativně jednoduchá, ale dle mých experimentů vycházela nejlépe. Zde by možná stálo za pokus lépe nastavit moje experimenty s neuronovou sítí, která provádí převážně vertikální pooling, protože se tato síť přiblížila nejvíce mnou vybrané architektuře.

Domnívám se také, že pokud by se rozšířila datová sada o více záznámů a celkové záznamy by byly z vyšších soutěží, tak by se výkonnost celé metody také zvýšila.

Jedním z dalších rozšíření systému by mohla být detekce drah závodníků. Podle mě by se tohoto výsledku dalo dosáhnout pomocí detekce nohy závodníka mezi cílovými čarami.

Kapitola 6

Závěr

Cílem této práce bylo vytvořit a aplikovat metodu na vyhodnocování cílových záznamů v atletice. Tento cíl byl splněn a s jistou přesností lze pomocí této metody vyhodnocovat cílové záznamy.

Znalost principu fungování cílové kamery a pravidel atletiky pramení z mých vlastních zkušeností, které jsem získal jako cílový rozhodčí pro Český atletický svaz. Seznámení s přístupy počítačového vidění a neuronovými sítěmi jsem demonstroval při vytváření vlastní architektury neuronové sítě a metody pro segmentaci pozadí záznamů. Moje navrhnutá metoda řeší velkou část plně autonomního systému pro detekci závodníků na cílovém záznamu, a to samotné vyhodnocení. Při práci jsem vytvořil i datovou sadu pro trénování neuronových sítí a provedl následnou implementaci pomocí knihovny PyTorch.

Moje metoda zvládá detektovat 70.94% závodníků na cílovém záznamu. Tito závodníci jsou detekováni s přesností na 1ms v 16.5% případů, na 2ms v 32.44% případů, na 5ms v 66.27% případů a na 10ms v 89.19% případů. v průměrné rychlosti celkového vyhodnocení do 2 minut.

Při vytváření této práce jsem se naučil lépe pracovat s neuronovými sítěmi. Naučil jsem se zpracovávat obrázky a vyzkoušel si aplikovat různé metody segmentace pozadí. Navíc jsem se naučil pracovat s knihovnou OpenPose v alespoň základním rozsahu.

V práci bych chtěl pokračovat tak, že se domluvím s firmou TimeTronics a budu čist cílovou fotografií spolu s časovými značkami, takže ve výsledku bude možné z mého programu přímo generovat výsledné časy doběhu závodníků pro výsledkový systém. Také bych rád vyzkoušel další architektury neuronových sítí a systém na detekci drah závodníků.

V práci by ještě někdo mohl pokračovat tak, že by tuto práci napojil na cílový systém a tím „nahradil“ program na vyhodnocování cílových fotografií, včetně ovládání veškerých periferií cílového systému.

Literatura

- [1] CAO, Z., MARTINEZ, G. H., SIMON, T., WEI, S. a SHEIKH, Y. A. *OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields* [online]. V1.7.0. 2019 [cit. 2021-17-05]. Dostupné z:
<https://github.com/CMU-Perceptual-Computing-Lab/openpose>.
- [2] JANÁKOVÁ, I. *Segmentace* [online]. 2020 [cit. 2021-17-05]. Dostupné z:
http://vision.uamt.feec.vutbr.cz/POV/lectures/05_Segmentace.pdf.
- [3] LI, J., NIE, Y., ZHAO, M., SHUANG, F., ZHU, B. et al. A high accuracy automatic timing method for photo finish systems. In: China. *2014 IEEE International Conference on Progress in Informatics and Computing*. 2014, s. 195–199 [cit. 2021-17-05]. DOI: 10.1109/PIC.2014.6972323. ISBN 978-1-4799-2030-3.
- [4] MAKLIN, C. *Dropout Neural Network Layer In Keras Explained* [online]. 2019 [cit. 2021-17-05]. Dostupné z: <https://towardsdatascience.com/machine-learning-part-20-dropout-keras-layers-explained-8c9f6dc4c9ab>.
- [5] OPENCV. *Canny Edge Detection* [online]. 2013 [cit. 2021-17-05]. Dostupné z:
https://opencv-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_canny/py_canny.html.
- [6] PYTORCH. *Pytorch Image Augmentation using Transforms*. [online]. 2020. 2020-09-02 [cit. 2021-17-05]. Dostupné z:
<https://androidkt.com/pytorch-image-augmentation-using-transforms/>.
- [7] SVAZ Český atletický. *Soutěžní a technická pravidla 2020* [online]. Leden 2020 [cit. 2021-17-05]. Dostupné z:
<https://www.atletika.cz/clenska-sekce/rozhodci/pravidla1/>.
- [8] TIMETRONICS. *Photo Finish Manual - MacFinish 2D 200/300/400* [online]. 1. vyd. 2016 [cit. 2021-17-05]. Dostupné z: https://edbe4506-2408-49bb-8a63-250446456f4f.filesusr.com/ugd/a8e2ac_2e4c99e110454b5cbf49bf53df409a6c.pdf.

Příloha A

Struktura odevzdaných souborů

Na přiložených DVD lze najít tyto složky a soubory:

- **OpenPose_lib** - Přiložená knihovna OpenPose [1] pro python v „módu“ CPU.
- **dataset** - Náležitosti potřebné pro vytvoření datasetu.
- **CNN_uceni** - Složka obsahujici scripty pro trénování neuronové sítě.
- **BP-complet.py** - Hlavní soubor pro spuštění.
- **OpenPose_process.py** - Soubor pro zpracování obrázků knihovnou OpenPose.
- **cutout.py** - Soubor pro vytvoření výřezů závodníků.
- **CNN.py** - Soubor obsahující konvoluční neurunovou síť a její vyhodnocení.
- **README.md** - soubor obsahující informace pro zprovoznění a ovládání programu.
- **model.pth.tar** - předučený model konvoluční neuronové sítě.