ZADÁNÍ BAKALÁŘSKÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Žďánský** Jméno: **Filip** Osobní číslo: **485270**

Fakulta/ústav: **Fakulta informačních technologií**

Zadávající katedra/ústav: **Katedra aplikované matematiky**

Studijní program: **Informatika**

Studijní obor: **Znalostní inženýrství**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Predikce věku na základě 3D scanu obličeje

Název bakalářské práce anglicky:

Age prediction based on 3D facial scans

Pokyny pro vypracování:

Cílem práce je navrhnout a experimentálně ověřit metodu pro klasifikaci věku jedince na základě prostorového scanu obličeje. Zaměřte se zejména na metody strojového učení a neuronové sítě.
1. Prozkoumejte metody strojového učení vhodné pro zpracování prostorových dat.
2. Po konzultaci s vedoucím práce navrhněte experimenty modelování a klasifikace věku jedince na základě 3D scanu obličeje.
- zvýšenou pozornost věnujte věkovým hranicím 15 let (trestní odpovědnost jedince) a 18 let
- je známo, že průběh stárnutí vykazuje rozdíly v závislosti na pohlaví, pokuste se v experimentech toto zohlednit (využitím rozdílných modelů), případně vyřešit (např. explicitní klasifikací pohlaví na základě scanu, či komplexním modelem predikce věku, který obsáhne obě pohlaví)
3. Alternativně prozkoumejte možnost rozšíření prostorových bodů o texturu obličeje
Předmětem práce není sběr dat. Data poskytla katedra antropologie a genetiky člověka, PřF, UK.

Seznam doporučené literatury:

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Zdeněk Buk, Ph.D. katedra teoretické informatiky FIT

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

Datum zadání bakalářské práce: **17.02.2022** Termín odevzdání bakalářské práce: **12.05.2022**

Platnost zadání bakalářské práce: _____

Ing. Zdeněk Buk, Ph.D.	Ing. Karel Klouda, Ph.D.	doc. RNDr. Ing. Marcel Jiřina, Ph.D.
podpis vedoucí(ho) práce	podpis vedoucí(ho) ústavu/katedry	podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

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

. _____ Datum převzetí zadání

_____ Podpis studenta

© ČVUT v Praze, Design: ČVUT v Praze, VIC

Bakalářská práce

PREDIKCE VĚKU NA ZÁKLADĚ 3D SCANU OBLIČEJE

Filip Žďánský

Fakulta informačních technologií
Katedra aplikované matematiky
Vedoucí: Ing. Zdeněk Buk, Ph.D.
12. května 2022

České vysoké učení technické v Praze
Fakulta informačních technologií
© 2022 Filip Žďánský. Všechna práva vyhrazena..
Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení, je nezbytný souhlas autora.

Odkaz na tuto práci: Žďánský Filip. *Predikce věku na základě 3D scanu obličeje.* Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2022.

Obsah

Poděkování vii

Prohlášení viii

Abstrakt ix

Úvod 1

Cíle práce 3

1 Odhad věku 5
1.1 Biologické faktory ovlivňující věk . . . 5
1.2 Strojový odhad věku . . . 5
1.2.1 Strojové učení . . . 6
1.2.2 Klasifikace věku . . . 6
1.2.3 Regrese věku . . . 9
1.2.4 Křížová validace . . . 9
Shrnutí . . . 10

2 Popis využitých dat a jejich zpracování 11
2.1 Věkové zastoupení jedinců v datové množině . . . 11
2.2 Podoba dat . . . 11
2.3 Kvalita dat . . . 12
2.4 Příprava dat . . . 12
2.4.1 Rozšíření a balancování datové množiny . . . 12
2.4.2 Předzpracování polygonových sítí . . . 13
2.4.3 TFRecord format . . . 14
2.5 Trénování a testování modelů . . . 14
2.5.1 Implementace křížové validace . . . 14
2.6 Vstupní data a předzpracování . . . 15
Shrnutí kapitoly . . . 15

3 Nástroje a metodiky pro zpracování prostorových dat 17
3.1 Průzkum přístupů . . . 17
3.1.1 Multi-view . . . 17
3.1.2 Volumetric-based . . . 17
3.1.3 Point-based . . . 17
3.2 Implementace modelů zpracovávající trojrozměrná data . . . 18
3.2.1 Tensorflow verze 1.x a 2.x . . . 18
3.2.2 Výběr, zprovoznění a úprava architektur . . . 18
Další kroky . . . 20

4 Prvotní experimenty **21**
4.1 Dopady rozšiřování dat . . . 21
4.1.1 Popis experimentu . . . 21
4.1.2 Výsledky rozšiřování dat . . . 22
4.1.3 Závěry rozšiřování dat . . . 24
4.2 Konfigurace modelů . . . 24
4.2.1 Velikosti mračen bodů a rozměry a počet pohledů . . . 24
4.2.2 Počet parametrů modelů . . . 24
4.2.3 Separátní konvoluční sítě . . . 24
Shrnutí . . . 25

5 Porovnání modelů odhadující věk **27**
5.1 Porovnání na malých datech . . . 27
5.1.1 Provedení experimentu porovnávající modely na malých datech . . . 27
5.1.2 Chybovost modelů na malých datech . . . 28
5.1.3 Porovnání doby trénovaní modelů . . . 31
5.1.4 Výsledky porovnání modelů na malých datech . . . 32
5.2 Porovnání na velkých datech . . . 32
5.2.1 Rozšíření a balancování datové množiny . . . 32
5.2.2 Výběr modelů pro trénování na velkých datch . . . 33
5.2.3 Provedení experimentu porovnávající modely na velkých datech . . . 34
5.2.4 Vyhodnocení výsledků na velkých datech . . . 37
Shrnutí dosažených výsledků . . . 39

6 Návrh modelu predikující věk na základě pohlaví **41**
6.1 Experiment na množině vybalancované podle věku i pohlaví . . . 41
6.1.1 Rozdělení datové množiny . . . 41
6.1.2 Odhad věku na vybalancované množině podle pohlaví a věku . . . 43
6.1.3 Výsledky jednotlivých modelů . . . 43
6.2 Experiment na množině vybalancované podle věku . . . 47
6.2.1 Návrh experimentu . . . 47
6.2.2 Výsledky jednotlivých modelů . . . 47
Shrnutí . . . 48

7 Návrh modelu na základě věkových kategorií **49**
7.1 Klasifikace jedinců . . . 49
7.1.1 Odhad věkové hranice 15 let . . . 49
7.1.2 Odhad věkových hranic 18 a 40 let . . . 50
7.2 Komplexní model zohledňující věkové kategorie . . . 51
7.2.1 Referenční model . . . 52
7.2.2 Model s explicitně přidanými věkovými hranicemi . . . 52
7.2.3 Komplexní model . . . 52
7.2.4 Výsledky modelů zohledňující věkové kategorie . . . 53
Shrnutí . . . 53

Závěr **55**

Obsah přiloženého média **59**

Seznam obrázků

2.1 Věková pyramida dostupných jedinců v datové množině. 12
2.2 Ukázka skenu lidské tváře. 13

4.1 Průběh trénován na rozšířených datech. 23
4.2 Příklad pohledů na sken. 25

5.1 Průběh trénování Pointconv a Xception - malá množina. 29
5.2 Průběh trénování PointNet 1024, 4096 - malá množina. 29
5.3 Průběh trénování MVCNN - malá množina. 30
5.4 Průběh trénování PointNet++ SSG - malá množina. 30
5.5 Průběh trénování PointNet++ MSG - malá množina. 31
5.6 Diagram vybalancované a rozšířené trénovací množiny - velká množina. 33
5.7 Průběh trénování PointNet a SSG - velká množina. 35
5.8 Průběh trénování MSG a Pointconv - velká data 35
5.9 Průběh trénování MVCNN - velká data 36
5.10 Zobrazení velikosti chyby odhadu podle věkové kategorie - MVCNN 37
5.11 Zobrazení velikosti chyby odhadu podle věkové kategorie, krabicový graf - MVCNN 38
5.12 Zobrazení velikosti chyby odhadu podle věkové kategorie, krabicový graf s body - MVCNN 39

6.1 Vybalancovaná datová množina podle pohlaví a věku. 42
6.2 Vybalancovaná trénovací množina podle pohlaví a věku. 42
6.3 Porovnání odhadu ŽM a MM na Ž a M. 44
6.4 Porovnání odhadu ŽM a RM na Ž a M. 45
6.5 Porovnání EXT a RM na Ž a M. 46
6.6 Odhad MVCNN s pohlavím. 48

7.1 Záměna klasifikace věkových hranic 18 a 40 let. 51

Seznam tabulek

2.1 Metadata jednotlivých skenů - příklad. 14
2.2 Tvar tensorů pro různá data. 15

4.1 Dopady chyby odhadu pro rozšířená data. 22

5.1 Porovnání modelů - malá množina 28
5.2 Délka epochy - malá množina 32

5.3 Porovnání modelů - velká množina 34

6.1 Porovnání modelů zohledňující pohlaví. 43
6.2 Výsledné modely porovnávající pohlaví. 47

7.1 Míra přesnosti klasifikace - 4 foldy 50
7.2 Míra přesnosti klasifikace - 6 foldů 50
7.3 Klasifikace - precision, recall, f_1 51
7.4 Explicitní zohlednění modelu. 52
7.5 Porovnání modelů zohledňující věkové kategorie. 53

Chtěl bych poděkovat vedoucímu práce Ing. Zdeňku Bukovi, Ph.D. za vedení práce a kolegům z Katedry antropologie a genetiky člověka UK, především pak Laboratoři 3D zobrazovacích a analytických metod za poskytnutí dat.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací. Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 citovaného zákona.

V Praze dne 12. května 2022 .....

Abstrakt

Bakalářská práce se zabývá odhadem věku lidského obličeje zachyceném na trojrozměrném skenu tváře. Odhad je prováděn pomocí metod strojového učení zejména umělých neuronových sítí. Práce posuzuje chybovost odhadu věku pomocí různých umělých neuronových sítí. Zohledňuje různou reprezentaci dat, které lze převést z trojrozměrného skenu a analyzovat, a jejich dopad na výslednou predikci věku. Dále zkoumá dopad klasifikace v souvislosti s pohlavím osoby zachycené na skenu. V neposlední řadě posuzuje přesnost rozpoznání věkových skupin 15, 18 a 40 let.

Klíčová slova umělé neuronové sítě, odhad věku, regrese, klasifikace, strojové učení, 3D sken, tensorflow, lidská tvář

Abstract

The bachelor thesis deals with the estimation of the age of the human face captured in a three-dimensional facial scan. The estimation is performed using machine learning methods especially artificial neural networks. The thesis assesses the error rate of age estimation using different artificial neural networks. It considers the different representations of the data that can be converted from the three-dimensional scan and analyzed, and their impact on the resulting age prediction. It also examines the impact of classification in relation to the gender of the person captured in the scan. Finally, it assesses the accuracy of the recognition of the age groups 15, 18 and 40 years.

Keywords artificial neural network, age estimation, classification, regression, machine learning, 3D scan, tensorflow, human face

Úvod

S procesem stárnutí se setkávají všechny organismy na naší planetě. Změny způsobené tímto procesem se mohou projevit například změnou tvaru obličeje nebo zvlněním tvaru pokožky, vráskami. Popisem těchto změn se zabývá věda fyzické antropologie člověka, která zachycuje, a popisuje, vývoj lidského druhu, různých odchylek tělesných znaků, rozdílných tělesných znaků v závislosti na pohlaví aj. Díky těmto odlišnostem můžeme, jako lidský druh, rozlišovat věk jedinců a přisuzovat jim společenské postavení příslušící dané kultuře a přizpůsobovat tomu naše vzorce chování.

S neustále se rozvíjejícími novými technologiemi se lidská společnost snaží začleňovat větší množství elektronických zařízení do každodenní činnosti, a to ať z důvodů zjednodušení práce, zvládání mezilidských vztahů, zábavy nebo snižování zdrojů. Tím zde vzniká potřeba automaticky zpracovávat stále větší množství různorodých úloh, které je potřeba analyzovat a navrhovat odpovídající řešení.

Jednou z takových úloh je i rozpoznávání věku jedince. Využití se může nacházet v systémech doporučující produkty či služby, forenzních vědách, ale i v bezpečnostních systémech kontrolujících věk mladistvých, kvůli produktům nevhodným nebo dokonce zakázaným pro určité věkové kategorie jako je prodej tabákových výrobků a alkoholu osobám mladších 18 let.

Takovéto systémy mohou fungovat na bázi zadání věku zkoumaného jednotlivcem samotným. V tomto případě, ale systém naráží na problém zadání nepravdivých informací, které jedinec může zadat a nemožnost ověření jejich správnosti. V tomto případě zde přichází ověření zadaných hodnot pomocí pověřené osoby.

Nicméně, provedení kontroly externí osobou je v procesu automatizace neakceptovatelná v budoucích informačních systémech, a proto zde přichází přístupy strojového učení, které dokážou na základě vizuálního podnětu, či jiného, analyzovat tělesné znaky jako je lidský obličej a určit věk automaticky. A právě lidský obličej je jednou z oblastí zkoumání, která dokáže ve značné míře reflektovat věk člověka.

Jednoduchým přístupem je vytvoření modelu, který zpracovává fotografii lidského obličeje např. z frontálního pohledu, kdy na základě tohoto vstupu model provede strojovou analýzu a následně odhadne věk. Avšak fotografie zachycuje pouze dvojrozměrnou informaci, lidský obličej je trojrozměrný objekt, tudíž se při tomto způsobu zachycení skutečnosti mohou ztratit důležité informace, které jsou k přesnému odhadu věku potřebné. Proto se zde můžeme ptát, zdali není lepší místo dvojrozměrné fotografie analyzovat trojrozměrný sken, zachycující lidský obličej z vícera úhlů, čímž poskytneme více dat k analýze a zlepšíme tak schopnost odhadu predikujícího modelu...

V práci se budeme zabývat návrhem modelu strojového učení, který bude schopen rozpoznávat věk na základě trojrozměrného skenu tváře. Lidská tvář je komplexní struktura a k jeho analýze budeme využívat různé architektury umělých neuronových sítí, které porovnáme a na základě jejich úspěšnosti navrhneme vhodný model pro odhad věku.

Práci dělíme na 7 kapitol. V kapitole 1 čtenáři představíme ovlivnění vzhledu věkem a aspekty spojující se s odhadem věku společně s úvodem do strojového učení a definicí pojmů s tím souvisejících. V kapitole 2 popíšeme podobu dat na což naváže kapitola 3, kde představíme nástroje strojového učení, které použijeme na zpracování těchto dat.

V kapitola 4 se zabýváme prvními experimenty provedenými pro nastavení parametrů sítí a další konfigurace experimentů. Samotné experimenty, které jsou předmětem této práce, dělíme do tří posledních kapitol 5, 6 a 7.

Cíle práce

Cílem práce je otestovat nástroje strojového učení, které jsou schopné zpracovávat a klasifikovat trojrozměrná data a na základě těchto nástrojů navrhnout experimenty, které klasifikují věk člověka z 3D skenu tváře.

Cílem je navrhnout experimenty zabývající se klasifikací věku z modelů určených na klasifikaci trojrozměrných objektů a porovnat je na reálných datech. Cílem bude navrhnout metody zohledňující pohlaví zachycených jedinců a posoudit dopad této informace na výsledný odhad. Dále se zaměříme na odhad věkových hranicí 15 a 18 let a prozkoumáme přesnost jejich klasifikace. V práci také zohledníme texturu trojrozměrných dat a výsledky porovnáme s přístupem zohledňující pouze trojrozměrnou strukturu.

Výsledky budou mít přínos pro katedru antropologie a genetiky člověka Přírodovědecké fakulty UK, která výsledky zohlední v jejich výzkumu.

Kapitola 1
Odhad věku

V kapitole budou popsány základní biologické dopady stáří, které ovlivňují vzhled a pomocí, kterých lze stáří odhadnout.

Dále se kapitola zabývá teoretickou stránkou strojového učení. Představuje použité mechanismy strojového učení, které budou v práci využity.

1.1 Biologické faktory ovlivňující věk

Při odhadu věku je potřeba znát faktory, které ovlivňují vzhled a které se s věkem spojují. Kalendářní věk se odlišuje od biologického věku a ten je závislý na spoustě klasifikačních faktorech. Studie [1] dělí tyto klasifikační faktory na intrinsické a extrinsické. Intrinsické faktory mohou být zdraví nebo životní styl, kdežto extrinsické např. počasí, podnebí nebo okolí, ve kterém člověk žije.

Jednou z oblastí, na které se stáří jedince projevuje nejvíce je lidská tvář. Různé deformace tváře mohou být zapříčiněny právě stářím a jsou individuální pro každého jednotlivce [2]. Dále se může jednat o množství vrásek, kterými osoba disponuje, zároveň mohou hrát roli i emoční výrazy závislé na každém jednotlivci [2] nebo množství vlasů, resp. vousů a jejich úprava.

Vrásky hrají důležitý prvek při posuzování věku, nicméně věk nelze odhadnout pouze z tohoto příznaku. Ukazuje se, že jejich množství může být ovlivněno faktory jako je sport, kouření nebo používání různých kosmetických doplňků, navíc se podle [2] vrásky u žen objevují dříve a častěji než u mužů.

Je důležité zmínit, že se tvar tváře mění s časem a to nerovnoměrně [3]. V pubertě se vzhled mění rychleji a výrazněji, než je tomu v dospělosti [4], kde se mění pomalu.

Obecně lze konstatovat, že odhad lidského věku je komplexní problém, který záleží na mnoha faktorech. Musí se zohlednit fyzická, geologická a biologická podstata člověka. Podle [5] se mohou doktoři a kriminalisté při odhadu věku mýlit až o 10 let.

1.2 Strojový odhad věku

Strojový odhad věku je prováděn pomocí různých metod *strojového učení* a různých reprezentací dat. Mezi nejčastější reprezentace patří obrázek osoby nebo tváře osoby jako je tomu např. v článku [6, 7].

Metody použité k odhadu věku z obrázku mohou být založené na deskriptoru vytvořeného z tohoto obrázku jako je *Local Binary Pattern* (LBP), který ohodnocuje každý pixel obrázku a vytváří z tohoto ohodnocení histogram, a s použitím dalších metod jako je metoda *k-nebližších*

sousedů [6], nicméně se ukazuje, že lepších výsledků dosahují metody hlubokého učení pomocí umělých neuronových sítí [7, 8].

Další reprezentace dat mohou být míry člověka jako je výška, šířka, hmotnost... nebo záznamy radiologických dat [9].

Odhad věku je nejčastěji prováděn pomocí klasifikační nebo regresní úlohy, kterou provádí natrénovaný *model* z metod strojového učení. Modelu jsou předložena trénovací data, pomocí kterých nalézá vzory příslušného věku, které se učí. Po natrénování provádí příslušný odhad jedince.

Existují studie, které se zabývají analýzou trojrozměrného skenu tváře [10, 11], a podle kterých lze věk na základě těchto dat rozlišovat tj. identifikovat rozdíly v tváři způsobené věkem a použít je k strojovému odhadu.

1.2.1 Strojové učení

Strojové učení je jedna z podkategorií umělé inteligence [12]. V naší práci budeme využívat tzv. *supervizované učení*. V principu se navrhne model, kterému se poskytnou data, příklady, na kterých se model natrénuje, tj. v příkladech vyhledá statistické struktury - vzory, a následně provádí automatizovanou činnost, například odhad hodnoty z poskytnutých dat. Model můžeme učit na základě tři věcí [12]:

1. *Vstupní datové body* – Obrázky tváří.
2. *Očekávaný výstup datového bodu* – Informace o skutečném věku jedince zachyceném na obrázku.
3. *Metriku úspěšnosti* – Velikost chyby odhadu věku.

Podskupinou strojového učení je *hluboké učení*. Modely uplatňující tento přístup jsou založené na vrstevnatých strukturách, kde se vrstva učí vzory z naučených vzorů předchozích vrstev [12]. Tento přístup lze realizovat pomocí *vícevrstvých umělých neuronových sítí* (MLP).

Model je trénován pomocí *ztrátové funkce* (loss), která měří míru úspěchu řešené úlohy, vytváří *ztrátové skóre* (loss score). Pomocí tohoto skóre se pak snažíme minimalizovat hodnotu ztrátové funkce. Tímto způsobem model učíme resp. se model učí lépe řešit zadaný problém.

Úspěšnost modelu lze sledovat pomocí zvolené *metriky*, která může být shodná se ztrátovou funkcí, anebo se může jednat o jinou funkci.

Vstupní bod bude sken jedince. Očekávaný výstup bude záviset na typu prováděné úlohy, bude se například jednat o hodnotu věku jedince nebo zda je mladší nebo starší dané věkové kategorii např. 18 let.

V této práci budou využívané modely strojového učení založené na vícevrstvých umělých neuronových sítí, které budou provádět odhad a to pomocí *klasifikace* a *regrese*, které popíšeme v následujících dvou sekcích.

1.2.2 Klasifikace věku

Klasifikace je úloha provádějící rozdělení datových bodů do diskrétních kategorií. V našem případě se jedná o klasifikaci věku, která bude dělit jedince do věkových kategorií. Za kategorii budeme považovat věkový interval shlukující více věků nebo věk samotný.

V klasifikační úloze budeme používat binární klasifikaci pomocí které budeme určovat, zda je jedinec mladší, nebo starší určitého věku.

1.2.2.1 Ztrátová funkce - kategorická křížová entropie

Při trénování klasifikační úlohy budeme využívat ztrátovou funkci - *kategorickou křížovou entropii*. Ztrátová funkce definovaná pro množinu o N datových bodech je vidět na 1.1.

$$-\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} 1_{y_i \in C_c} \log_2 \widehat{p}_{i,c} \tag{1.1}$$

kde:

- N: počet datových bodů v datové množině
- C: počet kategorií
- $1_{y_i \in C_c}$: pokud datový bod y_i leží v kategorii C_c pak se rovná 1 jinak se rovná 0
- y_i: i-tý datový bod
- C_c: c-tá kategorie
- $\widehat{p}_{i,c}$: odhadovaná pravděpodobnost i-tého prvku, že náleží c-té kategorii

1.2.2.2 Využité metriky pro klasifikaci

Úspěšnost modelu úlohy budeme posuzovat pomocí čtyř metrik: *míry přesnosti*, *precision*[1], *recall* a
F_1 skóre.

Míra přesnosti je poměr správně odhadnutých jedinců vůči všem odhadnutým jedincům, výraz 1.2.

$$\frac{S}{N} \tag{1.2}$$

kde:

- S: počet správně odhadnutých bodů
- N: počet všech odhadnutých bodů

V mnoha případech je míra přesnosti nedostačující a její výsledky jsou zavádějící, proto při posuzování využijeme následující metriky.

Precision a Recall metriky vypočítávají poměr tzv. *správně pozitivních* resp. *chybně pozitivních* a *správně negativních* resp. *chybně negativních* výsledků.

Pokud máme např. klasifikační úlohu, ve které odhadujeme, zda je jedinec mladší 18 let. Pak *pozitivní* výsledek je ten jedinec, který je mladší 18 let, v opačném případě bude výsledek *negativní*.

Konečně, pokud jedinci bude méně než 18 let a bude klasifikován jako mladší 18 let, pak se bude jednat o *správně pozitivní výsledek - true positive* (TP) v opačném případě o *chybně negativní výsledek - false negative* (FN), pokud bude jedinec starší 18 let a bude klasifikován jako starší 18 let, pak se jedná o *správně negativní výsledek - true negative* (TN) v opačném případě o *chybně positivní výsledek - false positive* (FP).

[1] Z důvodu možné záměny ponecháváme některé termíny v původním znění bez překladu.

Precision definovaná na 1.3 dává do poměru datové body klasifikované jako TP reps. FP.

$$\frac{TP}{TP + FP} \tag{1.3}$$

kde:

- TP: všechny správně pozitivní výsledky
- FP: všechny špatně pozitivní výsledky

Recall zobrazený na 1.4 pracuje s poměrem TP a FN klasifikovaných bodů.

$$\frac{TP}{TP + FN} \tag{1.4}$$

kde:

- TP: všechny správně pozitivní výsledky
- FN: všechny špatně negativní výsledky

Metrika agregující výsledky obou předchozích výrazů se označuje jako F_1 skóre[2] a je vidět na 1.5.

$$\frac{TP}{TP + \frac{FN+FP}{2}} \tag{1.5}$$

kde:

- TP: všechny správně pozitivní výsledky
- FP: všechny špatně pozitivní výsledky
- FN: všechny špatně negativní výsledky

Jedná se o harmonickou střední hodnotu funkcí precision a recall, která je vyšší, čím více jsou hodnoty metrik precision a recall podobné.

[2]Pro zjednodušení bude dále v práci označována jako F_1 nebo f_1.

1.2.3 Regrese věku

Dalším typem úlohy je regrese. Tato úloha odhaduje spojité číslo, v našem případě se jedná o věk v rámci spojitého věkového intervalu např. $(1; 100)$ let.

1.2.3.1 Ztrátová funkce - střední kvadratická chyba

Při trénování modelu v regresní úloze budeme využívat ztrátovou funkci - *střední kvadratickou chybu* (MSE), definovanou pro N datových bodů výrazem 1.6.

$$\frac{1}{N}\sum_{i=1}^{N}(Y_i - \widehat{Y}_i)^2 \tag{1.6}$$

kde:

- N: počet všech odhadovaných bodů
- Y_i: očekávaná hodnota výstupu datového bodu
- $\widehat{Y}_i$: odhadovaná hodnota datového bodu

Funkce posuzuje kvadrát rozdílu mezi skutečným věkem a odhadnutým věkem.

1.2.3.2 Využité metriky pro regresi

Pozorovací metriku budeme využívat - *střední absolutní chybu* (MAE), definovanou výrazem 1.7.

$$\frac{1}{N}\sum_{i=1}^{N}|Y_i - \widehat{Y}_i| \tag{1.7}$$

kde:

- N: počet všech odhadovaných bodů
- Y_i: očekávaná hodnota výstupu datového bodu
- $\widehat{Y}_i$: odhadovaná hodnota datového bodu

Hodnota funkce představuje *de facto* průměrnou chybu odhadu věku.

1.2.4 Křížová validace

Při testování úspěšnosti modelu budeme využívat *křížovou validaci*. Z originální datové množiny metoda vytvoří t stejných datových množin tzv. *foldů*. V každé nově vytvořené množině oddělí $\frac{1}{t}$ datových bodů, ze kterých vytvoří validační množinu a ze zbylých datových bodů vytvoří množinu trénovací.

Validační množina každého foldu je vytvořena tak, aby neobsahovala žádný prvek, který se vyskytuje ve validační množině jiného foldu.

Takto vznikne t různých páru trénovací a validační množiny, které pokrývají celou datovou množinu. Nad každým foldem je natrénován a validován samostatný model.

Shrnutí

V kapitole jsme popsali základní problematiku odhadu stáří. Řekli jsme si, že odhad věku je složitý problém, protože závisí na vzhledu člověka, který je ovlivněn mnoha proměnnými a přímo nereflektuje kalendářní věk.

Také jsme definovali pojmy strojového učení jako je klasifikace, regrese, ztrátová funkce nebo metrika, které budeme dále v práci využívat.

V následující kapitole představíme podobu dat, ve kterých jsou uloženy skeny lidských tváří a popíšeme předzpracování těchto dat, pomocí kterého data upravíme tak, abychom s nimi mohli dále pracovat.

Kapitola 2

Popis využitých dat a jejich zpracování

V práci využíváme trojrozměrné skeny lidských tváří poskytnuté katedrou antropologie a genetiky člověka Přírodovědecké fakulty UK. V datové množině jsou zastoupeni pouze občané České a Slovenské republiky. Práce se tedy orientuje jen na konkrétní část lidské populace a dosažené výsledky nemusí zcela reflektovat lidskou populaci.

2.1 Věkové zastoupení jedinců v datové množině

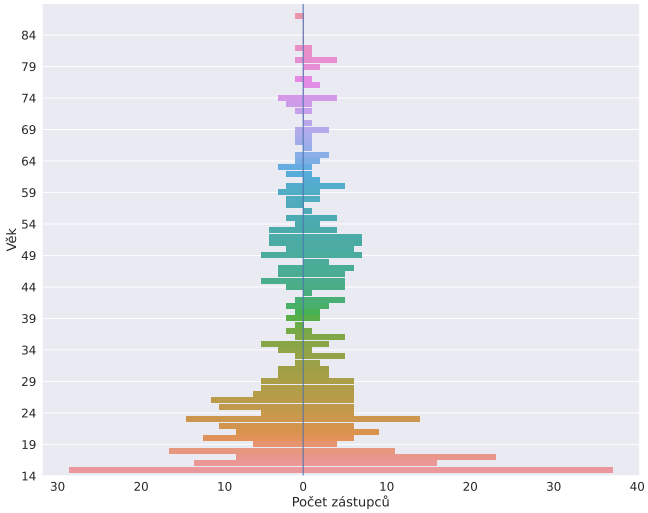
V použité datové množině se nachází jedinci v rozmezí 15–87 let, celkově se jedná o 550 zástupců. Jedinci nejsou zastoupeny rovnoměrně podle věku. Více je mladších jedinců mezi lety 15–30, méně je pak zástupců starších 55 let. V daném rozmezí 15-87 let není pravidlem, že má každý věk zástupce, např. věk 75 nebo 78 jsou zcela vynechány, také neplatí, že v daném věku jsou zástupci vždy z obou pohlaví, příkladem je věk 38 (pouze muž) nebo 48 (pouze žena). Distribuce jednotlivých let dostupné datové množiny je zachycena na grafu 2.1.

2.2 Podoba dat

Jednotlivé skeny jsou v podobě *polygonových sítí*. Skeny jsou oříznuté a k dispozici je pouze jejich přední část obsahující obličej. Každý sken je uložen v souboru s příponou *.obj* obsahující celou polygonovou síť. Společně se sítí je k dispozici i soubor *.mtl* ve kterém je uložen materiál, který pokrývá příslušné plochy polygonové sítě, příklad skenu je vidět na 2.2.

V práci budeme používat převážně modely, které budou na vstupu dostávat mračna bodů. Tato mračna se vyrobí ze zmíněných polygonových sítí pomocí rovnoměrného vzorkování povrchu sítě.

Kromě mračna bodů budeme data využívat i ve formě dvojrozměrných obrázků představující pohledy na daný sken tváře. Pohledy jsou vytvořeny natočením polygonové sítě příslušného skenu pod zadaným úhlem.



Obrázek 2.1 Distribuce zástupců použité datové množiny podle věku a pohlaví. Nalevo od dělící svislé čáry procházející bodem 0 je zastoupena distribuce mužů, napravo distribuce žen. Osa y zastupuje věk, osa x počet zástupců v daném věku.

2.3 Kvalita dat

Ukázalo se, že některé skeny jsou dodány ve špatné kvalitě, tj. počet bodů polygonové sítě je kolem 3000 bodů, zbylé skeny mají kolem 40 000-110 000 bodů, bylo rozhodnuto, že se tyto skeny z datové množiny odeberou, protože v některých případech pracujeme s mračny o velikosti 4096 bodů. Jedná se o 8 zástupců. Celkově v práci využíváme 542 skenů.

Dále, k některým skenům nebyla dodána textura, proto v případě posuzování skenu a textury tyto jedince vynecháme, abychom do modelů nezanášeli chybu. Jedná se o 4 zástupce. V případě, že pracujeme s modelem zohledňující texturu využíváme pouze 538 zástupců.

Některé skeny v určitých místech disponují černými skvrnami vzniklých špatným renderováním. Skvrny nejsou natolik výrazné, proto jsme se rozhodli tyto jedince v datové množině ponechat. Celkově tedy pracujeme s datovou množinou o 542 zástupcích v případě posuzování mračen bodů a s 538 zástupci v případě posuzování pohledů.

2.4 Příprava dat

Předtím, než bude data možné použít v modelech strojového učení, je potřeba je zorganizovat a upravit pro jejich snazší použití.

2.4.1 Rozšíření a balancování datové množiny

Nerovnoměrná distribuce jednotlivců může mít negativní vliv na celkovou korektnost výsledků prováděných experimentů. Tento faktor částečně eliminujeme pomocí dorovnání jednotlivých let pomocí již zastoupených jedinců v množině.

Zároveň, z důvodu malého počtu datových záznamů (550 jedinců), je celá datová množina uměle rozšířena o kopie již existujících jedinců. Na tyto záznamy je aplikována datová augmentace, která do množiny zavádí šum [12]. Použité metody datové augmentace, budou popsány u příslušných experimentů.

(a) Ukázka skenu uloženém v souboru s příponou .obj obohaceného o materiál souboru .mtl.

(b) Ukázka skenu uloženém v souboru s příponou .obj.

Obrázek 2.2 Ukázka skenu lidské tváře.

Dorovnání a rozšíření dat datové množiny je implementované v modulu *data_reader*[1] ve třídě *DatasetBalancer*. Třída využívá 3 klíčové parametry podle kterých řídí balancování a rozšiřování:

1. *category* - určuje, zda se data vybalancují podle věku, pohlaví nebo věku i pohlaví
2. *max_* - určuje, kolikrát se může v množině jedinec, který je duplikován kvůli balancování, maximálně objevit
3. *leasts_samps* - určuje, kolikrát se jedinec duplikuje kvůli rozšíření (násobí parametr *max_one*)

Algoritmus napočítá počet výskytů každého zástupce v množině podle vzorce:

$$\min\left(|a-b|, maxone \cdot a\right) \tag{2.1}$$

kde:

- a: počet jedinců stejného věku
- c: počet jedinců věku s nejvíce zástupci
- $maxone$: maximální počet jedinců

poté přenásobí tento výsledek u každého vzorku hodnotou *least_samps*.

Dorovnání a rozšíření bude prováděno pro každý experiment individuálně podle potřeby a před každým experimentem bude způsob uveden.

Informace o pohlaví a věku jedince zobrazeném na skenu jsou uložené v tabulce v *csv* formátu v podobě metadat. Záznam jedince je označen unikátním identifikátorem - *name*, který se shoduje s názvem souboru se skenem (bez přípony). Příklad tabulky a záznamu je vidět na 2.1.

2.4.2 Předzpracování polygonových sítí

Skeny uložené v souborech *.obj* je potřeba po jednom načíst a ovzorkovat, abychom ze skenů získali mračno bodů. V případě, že chceme získat jednotlivé pohledy, musíme načíst každý sken a pod určeným úhlem jej natočit a uložit pohled.

[1] Jednotlivé moduly napsané v jazyce Python jsou dostupné na přiloženém médiu.

Tabulka 2.1 Příklad tabulky uchovávající metadata všech záznamů, pozn. M - muž, F - žena.

name	sex	age
...	...	...
GB034	M	18
GB035	F	17
DOM2402	F	50
...	...	...

Pro práci se skeny je využita open source knihovna *trimesh*[2]. Knihovna disponuje všemi potřebnými funkcemi pro práci, zobrazení a ovzorkování skenu.

Funkce jsou využity v modulu *data_reader*, který byl vytvořen pro potřeby této práce. Model zajišťuje načítání a zpracování dat, jejich úpravu, balancování aj. Dále byl vytvořen modul *view_maker*, který zajišťuje tvorbu pohledů na sken.

2.4.3 TFRecord format

Kvůli redukci času a zjednodušení práce s daty při načítání a předzpracování dat při trénování jsou skeny ovzorkované, případně jsou předpřipravené pohledy, předem uložené v binárním formátu TFRecord[3]. V příslušném binárním souboru jsou uložena mračna bodů, pohlaví, věk, jmenné identifikátory případně cesty k obrázkům reprezentující pohledy na sken.

Funkcionality napsané pro ukládání dat do těchto záznamů se nachází v modulu *tf_recutils*, funkce určené k načítání dat z TFRecord formátu se nachází v modulu *to_tfrecord*.

Script, který z dat na předané cestě vytvoří požadovaný binární soubor, je v modulu *to_tfrecord*. Script podporuje balancování i rozšiřování podle zadaných sloupců. Disponuje 2 módy:

- *simple*, do binárního souboru ukládá pouze mračna bodů a věk odpovídajících jedinců
- *all_cols*, do binárního souboru ukládá mračna bodů, věk jedinců, pohlaví a identifikátor jedince

2.5 Trénování a testování modelů

Modely strojového učení se trénují pomocí trénovací množiny, která se oddělí od datové množiny. Data jsou během trénování a testování předkládána pomocí *dávek* určité velikosti, protože se celá množina dat nevejde do paměti zařízení. Ze zbylých dat se vytvoří validační a testovací množiny. Z důvodu nedostatku dat v experimentech nevytváříme testovací množinu, ale pouze validační.

Kompenzaci testovací množiny zajišťujeme pomocí *křížové validace*. Výjimku tvoří některé experimenty z kapitoly 4, které pouze prozkoumávají prvotní nastavení experimentů a není kladen důraz na přesnost výsledků.

Zároveň trénováním vícera modelů na všech určených foldech můžeme vyhodnotit všechna dostupná data v datové množině a nejen validační, resp. testovací.

2.5.1 Implementace křížové validace

Křížová validace byla implementována tak, aby se data, která se duplikovala během balancování a rozšiřování, nenacházela zároveň v trénovací a validační množině datových foldů.

[2]Dostupná z: https://github.com/mikedh/trimesh, použitá verze 3.10.2
[3]`https://www.tensorflow.org/tutorials/load_data/tfrecord`

Jednotlivé foldy jsou nejdříve vytvořeny na nevybalancovaných a nerozšířených datech a rozděleny na trénovací a validační množiny. Trénovací množina příslušného foldu je následně vybalancovaná, rozšířena a uložena do souboru v TFRecord formátu. Validační množina není nijak upravována, pouze se uloží do samostatného souboru stejného formátu jako trénovací množina. Algoritmus je implementován v modulu *prepare_folds*.

Nad takto vytvořenými foldy je spuštěná samotná křížová validace, která načítá data ze souborů příslušící danému foldu a trénuje a validuje model. Algoritmus je implementován v modulu *tf_cross_validation_from_files*.

Některé experimenty nevyžadují balancování a rozšiřování, proto je implementovaná i jednodušší verze, kde křížová validace pracuje s daty v paměti. Tato verze je implementovaná v modulu *tf_cross_validation*.

2.6 Vstupní data a předzpracování

Data vstupující do modelu se načítají z vytvořených TFRecord souborů v podobě tensorů, které jsou implementované v jedné z knihoven platformy *tensoflow*[4]. V tabulce 2.2 je uveden tvar tensoru pro využívaná data.

Tabulka 2.2 Tvar tensorů reprezentující data vstupující do modelů umělých neuronových sítí. Zobrazuje typ vstupních dat (typ dat), tvar tensoru (tvar) a ukázku reálných dat (příklad).

typ dat	tvar	příklad
mračno bodů	(počet bodů, souřadnice)	(1024, 3)
obrázek pohledu	(šířka, výška, kanál)	(120, 120, 3)
věk	(hodnota)	(1,)
pohlaví	(hodnota)	(1,)

Nad tensory se aplikuje předzpracování v podobě filtrování, datové augmentace, zakódování nominálních příznaků, promíchání atd. pomocí datového API[5].

V případě, že pracuje s pohlavím, kódujeme jej pomocí *one-hot* kódování, v tomto případě má tvar tensoru s pohlavím podobu *(2,)*.

Totéž provádíme v případě zakódování kategorie při klasifikaci, kdy nejdříve převedeme hodnoty na příslušné kategorie a následně uplatníme one-hot kódování. V práci se omezujeme pouze na binární klasifikaci, proto má tvar tensoru podobu *(2,)*.

Během trénování a validace je na data uplatněno promíchání, aby se zajistila nezávislost vzorků během trénování a nezanášely se tak do modely nevhodné vzory, které reflektují uspořádanost dat v množině.

V neposlední řadě, je na data při trénování aplikována datová augmentace, při které se do vzorků zavádí šum. Tohoto je využito v případě experimentů, ve kterých operují namnožená data, aby se zamezilo přítomnosti identických kopií v trénovací množině. U mračen bodů se jedná o náhodné posunutí všech bodů o hodnotu z intervalu *(-0.003, 0.003)*. V případě pohledů jsou obrázky rotovány o úhel *0-30°*. Šum je zároveň zanesen i do hodnoty skutečného věku v případě regrese v podobě přičtení hodnoty v rozmezí *(0-0.5)*.

Shrnutí kapitoly

V kapitole jsme popsali podobu datové množiny a jednotlivých datových záznamů, které budeme v práci využívat. Popsali jsme si způsob balancování a rozšíření této množiny společně s

[4] `https://www.tensorflow.org/api_docs` verze 2.8.0
[5] `https://www.tensorflow.org/guide/data`

předzpracováním dat a způsobem trénování a testování modelů.

V následující kapitole se zaměříme na jednotlivé metody strojového učení, které v práci použijeme.

Kapitola 3

Nástroje a metodiky pro zpracování prostorových dat

Abychom mohli prostorová data využít v modelech strojového učení potřebujeme je mít v podobě, ve které je budou moci modely vhodně zpracovat, analyzovat a vyhodnocovat, 3D data jsou často reprezentovaná pomocí objemových mřížek, polygonových sítí nebo mračna bodů. Dále popsané metody zabývající se zpracováváním trojrozměrných dat jsou detailněji vysvětleny zde [13]. My si zde popíšeme tři hlavní přístupy ze zmíněné práce a to Multi-view, Volumetric-based a Point-based.

3.1 Průzkum přístupů

3.1.1 Multi-view

Multi-view přístup je založen na projekci, která z různých úhlů nasnímá dvojrozměrný pohled na sken, následně snímky posuzuje a agreguje do obecného deskriptoru, který je klasifikován. Takovýto přístup implementuje například síť MVCNN [14]. Síť tedy zpracovává dvojrozměrné reprezentace pomocí dobře známých a osvědčených metod, které pomocí sofistikované agregace spojí do obecného deskriptoru a na základě deskriptoru provede odhad.

3.1.2 Volumetric-based

Jiný přístup, Volumetric-based, využívá trojrozměrnou mřížku, kterou vytvoří ze skenu, na rozdíl od dvojrozměrné matice v prvním přístupu, zpracovává celý objem. Na tuto matici aplikuje trojrozměrnou konvoluci pomocí 3D konvolučních sítí. Nevýhodou přístupu je škálování, protože časová a paměťová složitost roste kubicky s objemem dat.

3.1.3 Point-based

Posledním zmíněným přístupem jsou Point-based metody, které mají několik podkategorií. My si zde uvedeme pouze dvě podkategorie, a to Pointwise Multi-Layer Perceptrons (MLPs) a Convolution-based. Tyto metody jsou založené na posuzování každého bodu. Vstupními daty těchto sítí je mračno bodů. Jedná se o množinu obsahující trojrozměrné souřadnice bodů daného objektu, které jsou vytvořeny z povrchu trojrozměrného objektu.

3.1.3.1 Pointwise MLPs

Metoda posuzuje každý bod samostatně a vytváří obecný deskriptor pomocí agregační funkce. Počáteční prací v tomto přístupu je PointNet [15]. Tato síť zpracovává informace o bodech pomocí vícevrstvých sítí (MLP), které agreguje pomocí vrstvy max-pooling. Spousta metod založených na Pointwise přístupu rozšiřuje PointNet a přidává nové prvky, příkladem může být síť PointNet++ [16], která lépe zachycuje škálované lokální struktury nebo síť PointASNL, která je postavena na PointNet++ přístupu [17].

3.1.3.2 Convolution-based

Point cloud je, na rozdíl od dvojrozměrných mřížkových struktur jako jsou obrázky, nepravidelná struktura, proto není tak lehké vytvořit konvoluční jádra pro jejich zpracování. Z tohoto důvodu se konvoluce dělí na dvě skupiny, a to spojitou a diskrétní.

Spojitá konvoluce je založena na centrálním bodu, vzhledem ke kterému se na základě prostorové distribuce jednotlivé body váží. Zástupcem této metody je PointConv neuronová síť [18].

Diskrétní konvoluce pracuje s pravidelnými mřížkami, které váží jednotlivé body podle offsetu vzdálenosti od centrálního bodu.

3.2 Implementace modelů zpracovávající trojrozměrná data

Studie [13] zkoumá různé přístupy a architektury umělých neuronových sítí, které zpracovávají prostorová data, a porovnává míru přesnosti. Z této práce jsou vybráni zástupci, které využíváme a porovnáváme při odhadu věku.

Jednotlivé architektury bylo potřeba stáhnout a upravit tak, aby je bylo možné použít pro odhad věku. V souvislosti s tím se zde objevilo několik problémů, které bylo potřeba vyřešit. Jednalo se o nekompatibilitu verze Tensoflow, ve které byla architektura napsána, a ve které je v práci využívána.

Dále některé architektury disponují utilitami, které byly napsány pomocí jazyka C++, tyto knihovny bylo potřeba zkompilovat na lokálním stroji, na kterém byly architektury použity.

Modely, využité v práci, jsou implementované pomocí platformy Tensorflow ve verzi 2.x [19] v jazyce Python 3 [20].

3.2.1 Tensorflow verze 1.x a 2.x

Knihovna Tensorflow s verzí 2.x změnila API, zjednodušila jej a zahrnula do něj Kerasovou knihovnu[1].

Mezi verzemi 1.x a 2.x existuje vzájemná nekompatibilita, která se řeší pomocí automatizovaného scriptu, který převede verzi 1.x do podoby kompatibilní s verzí 2.x. Nicméně, i poté je potřeba řešit individuální záležitosti, které script automaticky nevyřeší. Existuje oficiální postup, jak tuto nekompatibilitu vyřešit[2].

Takto převedený kód využívá stále API verze 1.x, pouze jej lze využít v nainstalovaných knihovnách verze 2.x.

3.2.2 Výběr, zprovoznění a úprava architektur

Architektury se vybíraly podle [13]. Většina oficiálních implementací vybraných architektur pro zpracování prostorových dat byla napsána ve verzi 1.x, protože se ukázalo, že pracovat s verzí

[1] `https://www.tensorflow.org/guide/migrate/tf1_vs_tf2`
[2] `https://www.tensorflow.org/guide/migrate`

1.x je bez podrobnějších znalostí velmi obtížné, byly vybrány architektury, které měly napsanou implementaci ve verzi 2.x a byly dostupné z neoficiálních repositářů.

Toto velmi omezilo počet architektur, ze kterých jsme mohly vybírat, např. architektura PointASNL [17], která v porovnání ve zmíněné studii, byla jedna z nejlepších, nebyla testována, protože k ní nebyla nalezena implementace v potřebné verzi.

Architektury, které jsme vybraly a se kterými operujeme v této práci jsou:

- PointNet [15]
- PointNet++ [16]
- PointConv [18]
- MVCNN [14]
- Xception [21]

V následující sekci uvedeme postup zprovoznění každé sítě, dostupnost implementace, kterou jsme použili, míru přesnosti, kterou dosáhly ve zmíněné studii na datové množině ModelNet 40 [22] způsobem označeným jako OA a kategorii sítě popsané v 3.1.

3.2.2.1 PointNet

Tato síť byla vybrána především pro její jednoduchost její přesnost dosahoval *89.2%*.

Síť nebylo potřeba nijak upravovat. Implementace je dostupná z repozitáře[3]. Síť spadá do kategorie PointWise MLPs.

3.2.2.2 PointNet++

Síť vychází z předešlé architektury a podle zmíněného porovnání má větší přesnost kolem *90,7%*.

Tato síť využívá utility napsané v jazyce C++, aby bylo možno síť použít, je potřeba tyto funkce zkompilovat. Proces instalace a kód samotný je dostupný v repozitáři[4].

Architektura disponuje 2 verzemi. První, pojmenovaná SSG, je odlehčená verze, která nevyužívá algoritmus nevzdálenějšího bodu (FPS), tento algoritmus lépe vzorkuje povrch polygonových sítí, ale je časově náročný. Druhá verze MSG tento algoritmus využívá. V práci používáme obě sítě.

Kategorie této sítě je Pointwise MLPs.

3.2.2.3 PointConv

Přesnost uváděná ve studii je *92.5%* a to ji činí nejlepší architekturou, kterou v práci prozkoumáme.

Podobně jako v předešlém případě využívá síť funkcionality napsané v jazyce C++, které je potřeba před využitím sítě zkompilovat. Proces instalace a kód samotný je dostupný z repozitáře[5].

Síť vychází ze sítě PointNet++, ale ve zmíněné studii je zařazená do kategorie Convolution-based.

[3] https://github.com/keras-team/keras-io/blob/master/examples/vision/pointnet.py
[4] https://github.com/dgriffiths3/pointnet2-tensorflow2
[5] https://github.com/dgriffiths3/pointconv-tensorflow2

3.2.2.4 Xception

Přesnost sítě na trojrozměrných datech nelze ověřit, protože sít není primárně určená na zpracování prostorových reprezentací, ale primárně ke zpracování dvourozměrných reprezentací.

Nicméně je v práci využita i na zpracování trojrozměrných dat. Slouží zde především pro porovnání sítí, které byly vytvořeny přímo na práci s prostorovými daty s architekturou, která k tomuto účelu nebyla navržena.

Z tohoto důvodu byla síť upravena do podoby, ve které bude moci zpracovávat mračna bodů. Úprava spočívá ve výměně 2D konvolučních sítí a agregačních vrstev typu MaxPooling za jejich 1D alternativy.

V následujícím modelu je architektura Xception využita i v její původní formě pro dvojrozměrná data. V tomto případě síť nebyla nijak upravena.

Původní implementace sítě je dostupná z repozitáře[6].

Tato síť nebyla vybrána ze zmíněné studie, ale protože zpracovává přímo mračna bodů, spadá do kategorie Point-based.

3.2.2.5 MVCNN

Ve studii nebyla architektura testovaná, proto její přesnost nelze porovnat stejným způsobem jako u předešlých sítí, ale v původní studii této architektury byla měřena na datové množině ModelNet 40 a dosahovala přesnosti *90,1%*, což je o *0,6%* méně než PointNet++.

Původní implementace modelu, interně používala síť ResNet, ta se ale při prvotním otestování na datové množině ModelNet 40, která byla k dispozici společně s kódem, ukázala jako nedostatečná (míra přesnosti dosahovala 5%), proto byla ResNet architektura vyměněna za arhitekturu Xception.

Původní implementace je dostupná v repositáři[7].

Síť zpracovává pohledy skenu, a proto spadá do kategorie Multi-view.

Další kroky

Představili jsme si knihovny, které budeme v práci používat společně s architekturami umělých neuronových sítí a dvěma různými přístupy zpracování prostorových dat, a to pomocí pohledů a mračen bodů.

Zmíněné architektury budou v práci porovnány na odhad věku pomocí regrese a ze získaných poznatků se vyberou ty, které budou následně použity pro další experimenty.

V rámci kapitoly 4 se modely ještě částečně upraví. Bude se jednat o úpravy, které jsou provedeny z důvodů malé datové množiny např. snížení počtu parametrů u jednotlivých sítí.

Dále se v následující kapitole prověří způsob rozšíření dat, pomocí kterého zvětšíme dostupnou datovou množinu, abychom docílí spolehlivějších odhadů.

[6] `https://github.com/keras-team/keras-applications/blob/master/keras_applications/xception.py`
[7] https://github.com/Mannix1994/MVCNN-Keras

Kapitola 4
Prvotní experimenty

Polygonové sítě reprezentující skeny se skládají z ploch, jejichž vrcholy jsou reprezentované prostorovými body. Počet bodů v polygonové síti se pohybuje mezi 40000-110000 body, tomu odpovídá počet ploch.

Strategie vzorkovacího algoritmu vybírá náhodně body z rovnoměrného rozložení. Kvůli tomu výsledná mračna bodů jednoho skenu nejsou stejná. Takže v praxi můžeme z jednoho skenu vytvořit několik různých mračen bodů a tím rozšířit množinu příkladů, kterými disponujeme.

V souvislosti s výběrem počtu bodů reprezentující sken, je potřeba určit počet bodů, které budou vstupovat do modelu neuronové sítě.

Dále, protože disponujeme malým množstvím dat, je potřeba zkontrolovat počet parametrů každé neuronové sítě, zda se nepřeučují v důsledku velkého počtu parametrů. Originální architektury byly trénované na řádově tisícovkách příkladů...

V této kapitole budou pomocí jednoduchých experimentů prozkoumány výše popsané problémy a na základě výsledků se tomu přizpůsobí následné experimenty v dalších kapitolách.

4.1 Dopady rozšiřování dat

V této části práce sledujeme dopady množení mračna bodů na celkový odhad věku. V experimentu množíme jedince pomocí množícího faktoru *sf* pro různé hodnoty, hodnota sf udává, kolikrát se bude jedinec v množině vyskytovat. Množení je provedeno pomocí vzorkovacího algoritmu, který volí náhodné body z polygonové sítě podle rovnoměrného rozdělení. Vzorkování je aplikováno nad stejným jedincem.

Experiment se provádí na množině 157 zástupců[1]. Věkové rozpětí jedinců se pohybuje mezi 17-80 lety.

4.1.1 Popis experimentu

V experimentu je použita architektura PointNet. Velikost mračna bodů je zkoušená pro hodnoty 1024 a 4096.

Odhad věku je prováděn pomocí regrese, jako ztrátová funkce se využívá MSE, metrika pozorování je využita MAE. Počet epoch je nastaven na 300, přičemž se sleduje změna ztrátové funkce, pokud nedojde ke změně během 40 epoch, trénování se zastaví. Velikost jedné dávky je zvolena na 32. Porovnáváme množící faktor *sf* pro hodnoty 1 (žádné množení), 3, 5, 14, 21, 29 a v případě 1024 bodů i 80.

[1] Sběr dat probíhal ještě během tvorby práce a v této době nebylo více jedinců k dispozici

Experiment nevyužívá testovací množinu. V trénovací množině se nachází 126 jedinců a ve validační 31 jedinců. Trénovací ani validační množina není nijak balancována. Během tohoto experimentu není použita křížová validace, zároveň nebyla nijak měněna velikost sítě, tj. počet parametrů a vrstev.

Cílem experimentu je ověřit, zda vícenásobné množení má vliv na odhad a domníváme se, že využití pouze validační množiny s výše definovanými podmínkami je pro tento úkol dostatečné.

4.1.2 Výsledky rozšiřování dat

Model PointNet je na každé kombinaci *počet bodů*, *sf* natrénován právě jednou, v tabulce 6.1 je vidět nejnižší dosažené MAE pro každý model.

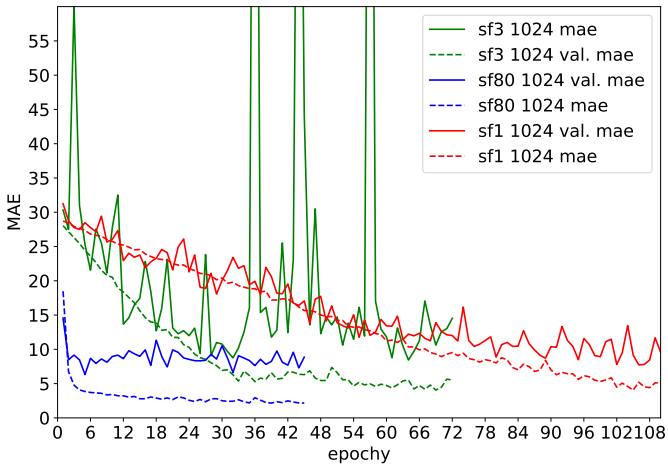
Tabulka 4.1 Výsledky modelů trénovaných na rozšířených množinách. Velikost rozšíření specifikuje parametr *sf* určující počet duplicit jednotlivých záznamů. Val. MAE je nejmenší validační MAE dosažené na modelu trénovaném na množině rozšířené parametrem *sf*. Počet bodů odpovídá velikosti vstupujícího mračna bodů.

Počet bodů	sf	Val. MAE
1024	1	6.51
1024	3	8.44
1024	5	6.65
1024	14	6.67
1024	21	6.56
1024	29	6.56
1024	80	**6.32**
4096	1	6.97
4096	3	7.18
4096	5	6.85
4096	14	6.01
4096	21	6.80
4096	29	**6.64**

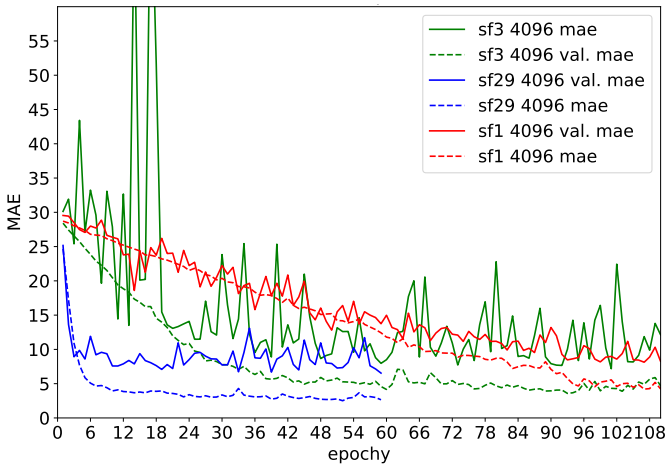
Množení u mračna bodů velikost 1024 bodů pro *sf* menší nebo rovno 29 dopadlo hůře, než pokud bychom mračna nemnožili tj. $sf = 1$, v případě $sf = 21, 29$ se rozdíl pohybuje pouze v setinách MAE. U množení *sf* rovno 80 došlo k mírnému zlepšení. Pro mračno bodů o velikost 4096 bodů dopadlo hůře pouze množení u *sf* rovno 3 v ostatních případech dopadlo lépe.

Na grafu 4.1a lze vidět trénovací a validační MAE během trénování modelu pro *sf* 1, 3, 80. Zlepšování validačního MAE se pro *sf* 80 zastaví po 5. epoše, kde dosáhne nejnižší hodnoty a následně se model začne přeučovat resp. stagnuje mezi hodnotami 6-12. Podobně se tak děje i u modelu reprezentující *sf* 3, který se začne přeučovat po 30 epoše, zároveň dosahuje vyšší oscilace během trénovaní. Poslední z modelů se začne přeučovat po 102 epoše.

Podobná situace nastává i pro počet bodů 4096, který je vidět na grafu 4.1b zde je místo *sf* 80 zobrazen průběh pro *sf* 29.



(a) Průběh trénování sítě pro parametr *sf* s hodnotou 1, 3 a 80 pro mračno o velikosti 1024 bodů, mae odpovídá trénovacímu MAE, val. mae odpovídá validačnímu MAE. Konvergence je rychlejší u namnožených dat, nedochází k velkému přeučení.



(b) Průběh trénování sítě pro parametr *sf* s hodnotou 1, 3, 29 pro mračno o velikosti 4096 bodů, mae odpovídá trénovacímu MAE, val. mae odpovídá validačnímu MAE. Konvergence je rychlejší u namnožených dat, nedochází k velkému přeučení.

Obrázek 4.1 Průběh validačního a trénovacího MAE během trénování sítě na rozšířené datové množině.

4.1.3 Závěry rozšiřování dat

Experiment ukázal, že duplikování má pro vyšší množící faktor pozitivní vliv a dokáže vylepšit odhad o několik desetin MAE. Pokud je hodnota nízká kolem 3, tak má naopak vliv negativní. Hodnota, od které je vliv pozitivní se liší podle počtu bodů.

Při vhodně zvoleném množícím faktoru kolem hodnoty 20 lze tuto metodu využít při balancování a rozšiřování datové množiny.

Experiment byl proveden na malém počtu dat, bez balancování a řádného otestování pomocí křížové validace.

4.2 Konfigurace modelů

V souvislosti s experimenty bylo potřeba určit hranice modelů, podle kterých je můžeme porovnávat. Pomocí rychlého experimentování na malé trénovací a validační množině, byly určeny velikosti mračen bodů, počet parametrů sítě, velikosti dávek a zafixovány hyperparametry modelů.

Netvrdíme, že zvolené konfigurace jsou nejlepším řešením. Při výběru jsme dělali kompromis mezi rychlostí trénování a přesností, kterou měly na použité množině. Experimenty zde nebudou uváděny. Všechny podoby použitých modelů se nachází v adresáři *models* na přiloženém médiu.

4.2.1 Velikosti mračen bodů a rozměry a počet pohledů

S rostoucím počtem bodů reprezentující mračno roste časová i paměťová složitost při trénování. Větší mračna bodů mohou potenciálně obsahovat více informací, menší jsou zase rychlejší pro trénování.

Některé sítě jako je např. PointNet nebo Xception jsou schopné na dostupných zařízení zpracovávat mračna až o velikosti 30 000 bodů s odpovídající velikostí dávky, např. PointNet je schopna tento mrak zpracovat pouze s velikostí dávky 4.

Zbylé sítě nejsou schopné velká mračna v paměti grafické karty uložit a pracovat s nimi. V práci proto použijeme mračna do velikosti 4096 bodů. Výjimku bude tvořit síť Xception, která je schopna tyto mraky zpracovat rychle a ve velkém počtu.

Vedle mračen bodů se budou v práci využívat i pohledy na trojrozměrné skeny použité v síti MVCNN. Každý sken bude mít celkem 7 pohledů z různých úhlů. Velikost každého obrázku bude 104 na 120 pixelů. Počet a rozměry pohledů byly voleny s ohledem na rychlost trénování sítě. Příklad pohledů je vidět na obrázku 4.2.

4.2.2 Počet parametrů modelů

Jednotlivé sítě byly testovány na různých počtech parametrů pomocí validační množiny. Počty parametrů jednotlivých modelů budeme držet kolem 700 000 parametrů. Jedná se kompromis mezi rychlostí trénování, schopností pojmout mračna bodů do velikosti 4096 bodů a přesností odhadu.

Výjimku bude tvořit síť Xception a MVCNN, které budou mít celkem 10 000 000 parametrů. MVCNN především, protože interně využívá Xception.

4.2.3 Separátní konvoluční sítě

Společně s úvodním experimentováním, při kterém se zvolily velikosti bodů a počet parametrů bylo část konvolučních vrtev používaných modelů vyměněno za separátní konvoluční vrstvy (SepConv). Ukázalo se, že lehce zlepšují výkon. Experimenty zde nebudeme uvádět, měly podobu shodnou s předchozími experimenty na stejné datové množině.

Obrázek 4.2 Příklad pohledů použitých jako vstup sítě MVCNN.

Shrnutí

V kapitole jsme se zabývali rozšiřováním dat v podobě mračna bodů, kterým namnožíme existující jedince a zvýšíme tak počet vzorků v použité datové množině, ukázalo se, že tento postup má pozitivní vliv při dostatečně velkém namnožení jedinců.

Dále jsme určili velikost a podobu vstupních dat, kterými budeme trénovat síť a velikosti jednotlivých sítí v podobě počtu parametrů v závislosti na předpokládané přesnosti odhadu a rychlosti trénování.

V následující kapitole porovnáme jednotlivé sítě pomocí konfigurací, které jsme zvolili v této části práce.

Kapitola 5

Porovnání modelů odhadující věk

Prvním z cílů práce je prozkoumat metodiky pro zpracování prostorových dat, porovnat chybovost a správnost odhadu věku a na základě dosažených výsledků navrhnout experimenty pro další cíle. Kapitola se skládá ze dvou různých experimentů.

V prvním experimentu porovnáme metodiky využívající mračna bodů pomocí křížové validace. Modely budou trénované na menší datové množině. Z výsledků vybereme nejvhodnější zástupce, které porovnáme v druhém experimentu.

Druhá část porovná modely vybrané v první části pomocí křížové validace na rozšířené množině. V této části se snažíme dosáhnout co největší přesnosti. Pomocí nejlepšího modelu následně zobrazíme jednotlivé odhady a posoudíme schopnost umělých neuronových sítí predikovat věk z trojrozměrných dat.

Jedním z cílů práce je rozšířit prostorová data o texturu obličeje. Tato problematika bude zahrnuta v této kapitole. Texturu obličeje budeme analyzovat pomocí sítě MVCNN, která posuzuje dvojrozměrné pohledy na sken s texturou.

5.1 Porovnání na malých datech

V této části porovnáváme různé architektury vybrané v kapitole 3 zpracovávající mračna bodů, jedná se o PointNet, PointNet++ (SSG i MSG), PointConv a Xception, společně se sítí posuzující pohledy na sken MVCNN. Posuzujeme zde chybovost odhadu sítě, velikost mračna bodů a délku trénování.

5.1.1 Provedení experimentu porovnávající modely na malých datech

V experimentu využíváme menší množinu o 450 jedincích[1]. Množina není nijak balancovaná ani rozšiřovaná. Je zde využita křížová validace o 6 foldech.

Jedince dělíme do trénovací a validační množiny tak, aby v každé byl zástupce daného věku, pokud je to možné. Pohlaví zde nezohledňujeme. V trénovací množině máme 375 jedinců, ve validační se jich nachází 75.

Metrika sledující chybovost modelu je MAE. Model využívá ztrátovou funkci MSE. Odhad vyhodnocujeme tak, že z každého foldu vybereme nejnižší dosaženou chybovost, tyto hodnoty následně zprůměrujeme. Výsledná hodnota udává chybovost modelu.

[1] V době provádění experimentu nebyla k dispozici všechna data.

Modely byly trénované na 500 epochách s předčasným zastavením v případě, že nedocházelo ke zlepšení.

5.1.2 Chybovost modelů na malých datech

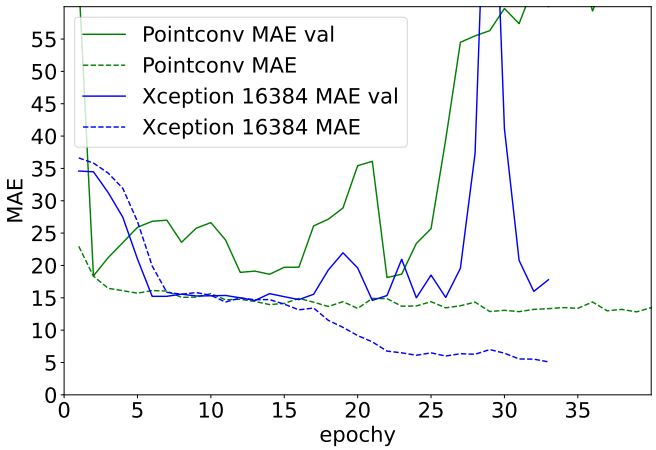
Jednotlivé modely, velikosti mračen, rozměry obrázků jsou společně s chybovostí zobrazené v tabulce 5.1.

Tabulka 5.1 Tabulka zobrazuje průměr minim validačních MAE dosažených v jednotlivých foldech (MV MAE) společně s minimy těchto validačních MAE každého foldu (MVF) pro typ sítě a velikost vstupu.

Typ sítě	Velikost vstupu	MVF	MV MAE
PointNet	512 bodů	9.26, 7.83, 11.74, 11.93, 8.27, 8.44	9.57
PointNet	1024 bodů	9.87, 9.44, 9.15, 10.42, 9.10, 8.44	9.40
PointNet	4096 bodů	9.75, 8.31, 9.90, 9.79, 9.54, 9.17	9.40
PointNet++ SSG	512 bodů	8.75, 10.17, 8.98, 9.72, 9.06, 9.60	9.38
PointNet++ SSG	1024 bodů	9.56, 10.32, 8.35, 9.33, 10.01, 9.98	9.59
PointNet++ SSG	4096 bodů	9.85, 9.75, 10.53, 10.13, 9.98, 9.76	9.75
PointNet++ MSG	512 bodů	8.97, 10.61, 8.43, 10.19, 10.51, 10.19	9.82
PointNet++ MSG	1024 bodů	8.76, 10.29, 8.49, 9.56, 8.34, 10.02	**9.25**
PointNet++ MSG	4096 bodů	9.32, 11.48, 10.91, 10.39, 10.83, 11.57	10.75
Poitnconv	1024 bodů	18.12, 10.84, 27.45, 15.15, 15.33, 12.27	10.84
Xception	1024 bodů	16.09, 15.44, 15.49, 14.71, 13.85, 14.22,	15.00
Xception	16384 bodů	14.58, 14.03, 14.51, 12.67, 12.34, 11.642370	11.64
Xception	32768 bodů	14.18, 14.77, 13.29, 13.017, 13.81, 14.34	13.90
MVCNN	$7 \times 104 \times 120$ pixelů	8.86, 8.87, 10.44, 9.80, 10.38, 8.74	9.52

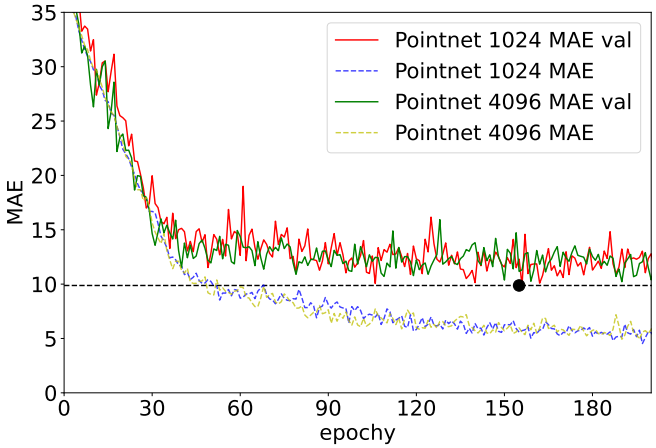
Nejmenší chyby při experimentu dosáhl model PointNet++ MSG pro 1024 bodů, jehož chybovost byla *9.25*. Nejhůře dopadla síť Xception a Pointconv. Sítě PointNet++ MSG, SSG, PointNet a MVCNN se pohybovaly pod *10* MAE.

Na grafu 5.1 je zobrazeno trénovací a validační MAE pro sítě Pointconv a Xception u nejlepších modelů. Obě sítě se začínají rychle přeučovat, validační MAE se vzdaluje od trénovacího. Pointconv kolem 17. epochy, Xception kolem 18. epochy.



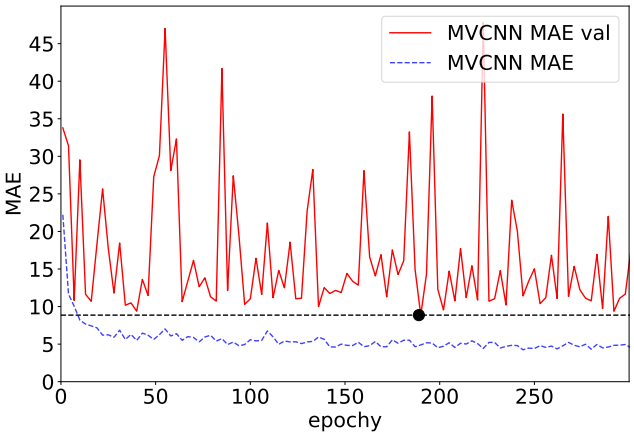
Obrázek 5.1 Změna trénovacího a validačního MAE (val) pro fold 0 u sítí Xception pro velikost vstupu 16 384 bodů a Pointconv pro velikost vstupu 1024 bodů. Na grafu je vidět přeučení sítě Pointconv od 17. resp. přeučení sítě Xception od 18. epochy.

Jiná situace nastává u zbylých sítí. Na grafu 5.2 zobrazující průběh trénování sítí PointNet pro 1024 a 4096 bodů se validační (val) MAE vzdaluje od trénovacího pomaleji, pro mračno bodů velikosti 512 je průběh obdobný. Nejnižší hodnoty jsme dosáhli kolem 150. epochy. Průběh trénovaní obou modelů je podobný. Validační MAE osciluje mezi nejnižší dosaženou hodnotou a hodnotou 15, přičemž se chybovost nezvyšuje.



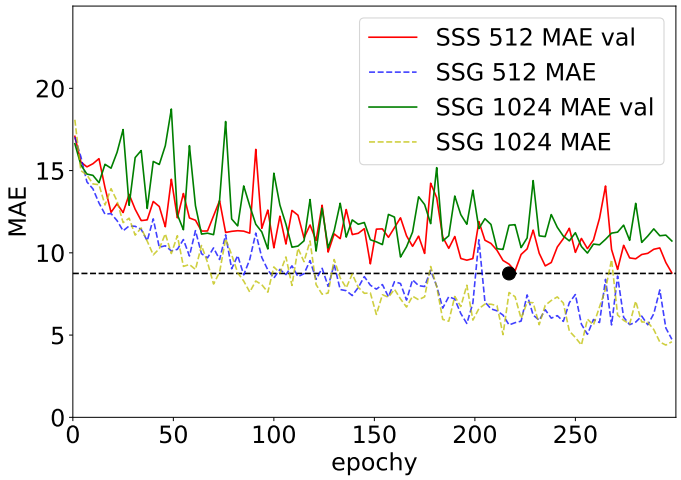
Obrázek 5.2 Změna trénovacího a validačního MAE pro fold 0 u sítě PointNet pro velikost vstupu 1024 a 4096 bodů. Tmavý bod označuje místo s nejnižším dosaženým MAE. Čárkovaná křivka označuje trénovací MAE, plná křivka validační MAE. Validační MAE u obou případů osciluje mezi hodnotami 10 a 15 MAE.

Oscilace MVCNN je výraznější, na grafu 5.3 se zlepšování trénovacího a validačního MAE výrazně zpomalí před 50. epochou. Model je trénován na 7 pohledech o velikosti 104×120 pixelů.



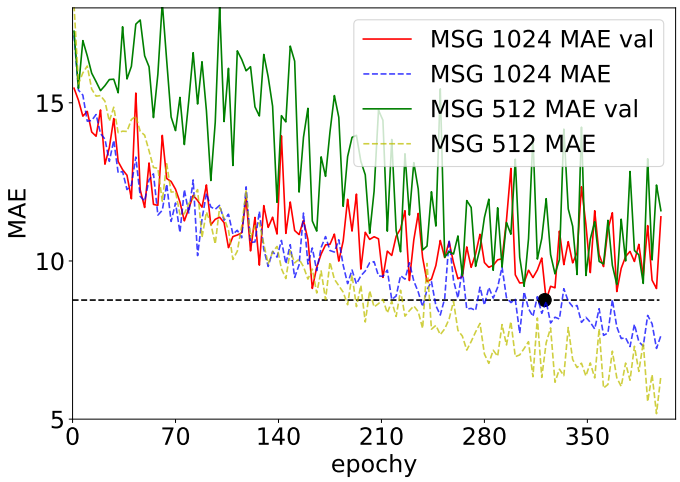
Obrázek 5.3 Změna trénovacího a validačního (val) MAE pro fold 0 u sítě MVCNN. Tmavý bod označuje místo s nejnižším dosaženým MAE. Čárkovaná křivka označuje trénovací MAE, plná křivka validační MAE. Na grafu je vidět nestabilita sítě MVCNN při trénování.

Křivky trénovacích a validačních MAE se nejméně liší u modelů PointNet++. Modelům trvá déle než se natrénují, ale méně se přeučují, průběh trénování sítě PointNet++ SSG na mračnech 512, 1024 je vidět na 5.4.



Obrázek 5.4 Změna trénovacího a validačního (val) MAE pro fold 0 u sítě PointNet++ SSG pro velikost vstupu 512 a 1024 bodů. Tmavý bod označuje místo s nejnižším dosaženým MAE. Čárkovaná křivka označuje trénovací MAE, plná křivka validační MAE.

Průběh modelu MSG pro mračna 512 a 1024 pak na 5.5. Průběh trénování modelů u mračen velikosti 4096 bodů je obdobný, pro přehlednost jej v grafech neznázorňujeme.



Obrázek 5.5 Změna trénovacího a validačního (val) MAE pro fold 0 u sítě PointNet++ MSG pro velikost vstupu 512 a 1024 bodů. Tmavý bod označuje místo s nejnižším dosaženým MAE. Čárkovaná křivka označuje trénovací MAE, plná křivka validační MAE.

Z experimentu vyplynulo, že lépe dokážou věk odhadovat sítě PointNet++ (obě verze), MVCNN a PointNet, hůře pak Xception a PointConv. Špatný odhad u sítě Xception je očekávaný, protože síť není určena ke zpracování prostorových dat.

Model Pointconv dopadl výrazně hůře, než se očekávalo, MAE dosáhlo pouze hodnoty *10.84* oproti nejnižší dosažené hodnotě *9.25*. V kapitole 3.1 byla tato síť vyhodnocena jako nejlepší. Zde je její chybovost za zbylými sítěmi vyjma modelu Xception, zároveň se začala rychle přeučovat. Špatný výsledek může být dán malou datovou množinou, ze které model nedokáže extrahovat vhodné vzory a naučit se je rozpoznávat.

Nejlépe dopadl model PointNet++ MSG s hodnotou MAE *9.25*, obě verze PointNet++ se učily déle, ale obě dosáhly nejlepších výsledků. U MSG se projevilo výraznější zlepšení o 0.5 MAE s větším mračnem bodů. Model SSG lépe odhadovat věk pro mračna bodů velikosti 512 bodů než 1024 a 4096.

Síť PointNet dosahuje lepších výsledků nad většími mračny bodů, tj. velikosti 1024 a 4096 než nad 512.

MVCNN je na množině nestabilní, validační MAE hodně osciluje při trénování, ale predikce dosahují výsledků shodných se zbylými sítěmi. Nestabilita může být dána tím, že počet parametrů modelu je oproti ostatním modelům až desetinásobný, což se projevuje na malé množině.

5.1.3 Porovnání doby trénovaní modelů

Tabulka 5.2 ukazuje průměrnou dobu epochy každé sítě. Nejdéle trvá 1 epocha modelu Pointconv *82.233s*, což je výrazně déle než u ostatních (a taky důvod, proč nebylo zkoušeno více bodů než 1024 u této sítě). Druhá v pořadí je síť PointNet++ MSG nad mračny velikosti 4096 bodů. Třetí Xception, která zpracovává mračna o 32 768 bodech, což je dáno hlavně velkým počtem bodů, které síť musí zpracovat.

Druhá nejpomalejší je síť PointNet++ MSG, které trvá v průměru *25s* zpracovat mračna velikosti 4096 bodů, *10s* velikosti 1024 bodů a *5s* velikosti 512 bodů. To je skoro trojnásobek doby, kterou k tomu potřebuje model SSG a mnohonásobně delší doba než k tomu potřebuje síť PointNet. Trénování této sítě je z časového hlediska nejnáročnější, protože vedle délky jedné epochy i nejdéle konverguje (potřebuje kolem 300 epoch).

Tabulka 5.2 V tabulce je zaznamenána průměrná rychlost trénovaní jedné epochy (TE) pro síť (Typ sítě) s odpovídajícím vstupem (Velikost vstupu).

Typ sítě	Velikost vstupu	TE
PointNet	512 bodů	0.035s
PointNet	1024 bodů	0.061s s
PointNet	4096 bodů	1.29s
PointNet++ SSG	512 bodů	1.035s
PointNet++ SSG	1024 bodů	3.068s
PointNet++ SSG	4096 bodů	11.256s
PointNet++ MSG	512 bodů	5.124s
PointNet++ MSG	1024 bodů	10.238s
PointNet++ MSG	4096 bodů	25.288s
Poitnconv	1024 bodů	82.233s
Xception	1024 bodů	1.306s
Xception	16384 bodů	12.555s
Xception	32768 bodů	21.053s
MVCNN	$7 \times 104 \times 120$ pixelů	9.840s

5.1.4 Výsledky porovnání modelů na malých datech

V experimentu se demonstrovalo použití všech sítí, přičemž se porovnával průběh a doba trénování, velikost vstupních mračen bodů, a přesnost odhadu, resp. velikost chyby odhadu.

Nejvíce se přeučovaly sítě Pointconv a Xception, nejvíce epoch na trénování vyžadují sítě PointNet++. S výjimkou prvních dvou zmiňovaných sítí se modely moc nepřeučují, síť MVCNN má však během trénování velkou oscilaci validačního MAE.

Podle rychlosti trénování jedné epochy jsou sítě seřazené následovně PointNet, Xception, PointNet++ SSG, PointNet++ MSG, Pointconv. Síť MVCNN nelze objektivně posoudit, protože nezpracovává mračna bodů, ale pokud bychom zpracovávání posuzovali podle mračna velikost 1024, pak se řadí před PointNet++ MSG.

Ukázalo se, že je lepší volit spíše mračna o velikosti 1024 bodů s výjimkou sítě PointNet++ SSG, která lépe odhaduje s mračnem velikosti 512 bodů. Při použití mračen velikosti 4096 bodů se odhad nezlepšuje a zároveň roste doba, po kterou se model musí trénovat.

Nejmenší chybu odhadu má síť PointNet++ MSG, nejhorší pak Xception a Pointconv. Síť založená na posuzování pohledů MVCNN dosahuje podobných výsledků jako sítě založené na mračnech bodů.

5.2 Porovnání na velkých datech

Předchozí experiment nastínil přesnost odhadu věku každého modelu. Experiment měl nedostatky hlavně v podobě datové množiny, která nebyla nijak upravována, tj. balancována a rozšiřována. Na tento problém se zaměříme v této části.

Z časových důvodů použijeme pouze část z přechozích modelů. V tomto experimentu budeme pracovat s mnohem větší množinou a testovat všechny modely by nebylo časově zvládnutelné.

5.2.1 Rozšíření a balancování datové množiny

Nedostatek dat a nerovnoměrnost jedinců zastoupených v datové množině je jedním z problémů, který může mít velký vliv na kvalitu odhadu věku.

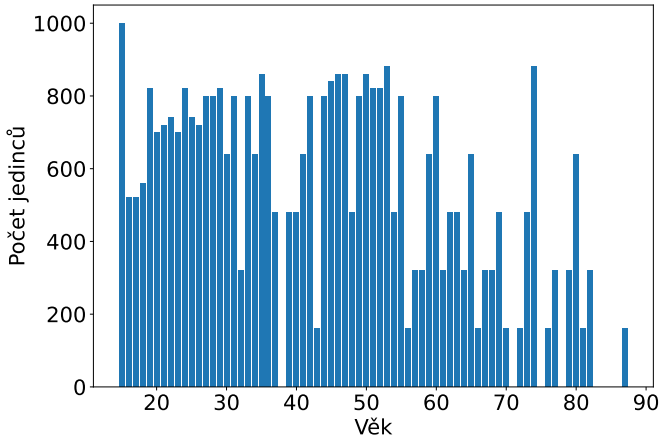
V kapitole 4 jsme využili vzorkovacího algoritmu, abychom rozšířili datovou množinu a ukázalo se, že rozšíření má pozitivní vliv na trénování a celkový odhad, pokud duplikujeme jedince dvacetkrát a více. V kapitole 2 byl popsán postup, jakým budeme množinu balancovat a rozšiřovat – modul *prepare_folds*.

Pomocí tohoto modulu vybalancujeme celou množinu podle věku. Nastavíme parametr *max_one* na hodnotu 8 a *least_samps* na hodnotu 20, to nám zajistí, že jedinec bude v datové množině zastoupen v podobě 20 až 160 duplicit v závislosti na tom, zda je potřeba daný věk balancovat. Zároveň budeme využívat křížovou validaci o 6 foldech.

Využitím modulu *prepare_folds* jsme z 542 jedinců vygenerovali 37791 vzorků, kde 37700 je v trénovací množině a 91 v množině validační tak, že duplicity jedince nejsou zároveň v trénovací a validační množině. Na počet jedinců se v trénovací množině vyskytuje 447 jednotlivců a ve validační 91.

Počty jedinců a jejich duplicit se mohou v jednotlivých foldech o pár jednotek lišit z důvodu algoritmu, který foldy vytváří, protože zohledňuje vyváženost trénovací i validační množiny podle věku. Množina není balancovaná podle pohlaví.

Příklad vyvážení jednoho z foldů je vidět na grafu 5.6



Obrázek 5.6 Diagram vybalancované datová množiny použité pro porovnání jednotlivých modelů. Množina je vybalancovaná podle věku s hodnotami parametrů *max_one*=8, *least_samps*=20 pro funkcionalitu provádějící balancování. Celkový počet vzorků je 37700.

5.2.2 Výběr modelů pro trénování na velkých datch

Vzhledem k tomu, že pracujeme s několikanásobně větší množinou, než jsme pracovali v předchozí části, již nebudeme porovnávat všechny alternativy tzn. kombinace modelu a počtu bodů na vstupu. V této sekci uvedeme modely a velikosti vstupujících mračen do těchto modelů společně s důvody, proč jsme je takto zvolili.

- *PointNet, 1024 bodů*: v prvním experimentu se ukázalo, že lepší odhad má model PointNet na větším mračnu o 1024 a 4096 bodech, zároveň je trénování sítě dostatečně rychlé, abychom takto velká mračna bodů mohli využít, také se ukázalo, že není velký rozdíl mezi odhadem provedeným nad mračnem o velikosti 1024 a 4096 bodů, aby bylo trénování rychlé volíme proto velikost mračna o 1024 bodech.
- *PointNet++ SSG, 512 bodů*: odhad provedený nad mračnem o 1024 bodech a 512 bodech byl lepší u mračna velikosti 512 bodů, proto volíme tuto možnost.

- *PointNet++ MSG, 512, 1024 bodů*: tato síť měla v předchozím experimentu nejlepší výsledky, proto ponecháme velikost mračna na 1024 bodech i přesto, že se bude trénovat výrazně déle než předchozí dvě možnosti, vyzkoušíme zároveň i mračno bodů velikosti 512.
- *Pointconv, 512 bodů*: síť byla trénována na 1024 bodech a její trénování trvalo výrazně nejdéle, přesto její chybovost byla jedna z nejhorších, kvůli dlouhé době trénování se omezíme na 512 bodů, domníváme se, že jedním z důvodů, proč síť měla špatné výsledky je, že potřebuje více dat k učení, protože je ze všech sítí nejsložitější, proto ji ponecháme v experimentu.
- *Xception*: jednoznačně nejhorší, ukázalo se, že vyšší přesnosti dosahuje na velikosti mračen kolem 16 384 bodů a výše, protože její výkon nebyl dostatečný a zároveň se nejedná o síť určenou pro zpracování prostorových dat nebude dále v experimentu použita.
- *MVCNN,* 104×120: síť měla podobné výsledky jako modely PointNet a PointNet++ SSG, doba trénování sítě je jedna z nejvyšších, přesto nijak nesnížíme rozměry pohledů a necháme je na stejné velikosti.

5.2.3 Provedení experimentu porovnávající modely na velkých datech

Sítě se testují pomocí křížové validace na 6 foldech. Opět používáme jako metriku MAE pro všechny modely. Ztrátová funkce je použita MSE. Počet epoch je stanoven na 150 s předběžným zastavením, pokud se validační MAE nelepší.

Počet epoch je snížen z důvodů delšího běhu každé epochy a předpokladu, že modely budou rychleji konvergovat, protože během epochy se budou trénovat na výrazně větší množině. Výsledky jsou vidět v tabulce 5.3

Tabulka 5.3 Výsledky jednotlivých modelů (Typ sítě) s příslušnými vstupy (Velikost vstupu) na vybalancované množině o 37700 vzorcích. Zobrazuje minimální validační MAE dosažené v každém foldu (MVF) společně s průměrem těchto minim (MV MAE).

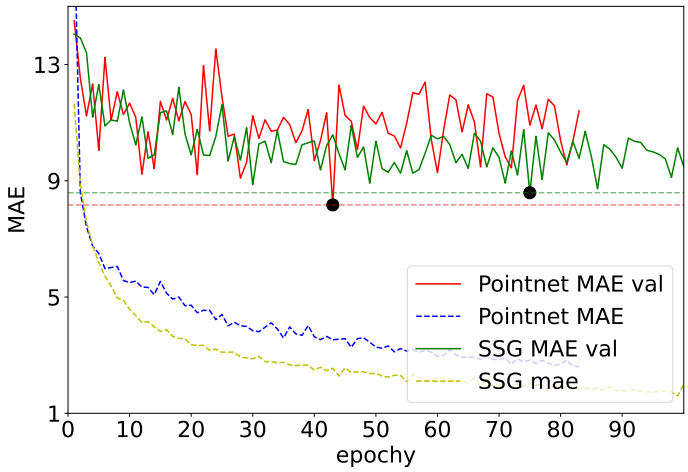
Typ sítě	Velikost vstupu	MVF	MV MAE
PointNet	1024 bodů	7.52, 9.15, 8.17, 9.46, 8.20, 7.78	8.38
PointNet++ SSG	512 bodů	7.22, 8.18, 8.59, 8.47, 7.93, 7.69	8.01
PointNet++ MSG	512 bodů	8.59, 9.91, 10.04, 8.45, 8.36, 9.87	9.20
PointNet++ MSG	1024 bodů	7.30, 9.87, 9.65, 8.79, 8.26, 9.50	8.90
Poitnconv	512 bodů	11.75, 16.18, 14.67, 12.19, 13.07, 14.87	13.79
MVCNN	$7 \times 104 \times 120$ pixelů	6.18, 8.15, 5.80, 7.00, 6.09, 6.85	**6.68**

Na vybalancované a rozšířené datové množině má většina modelů lepší výsledky než měla v prvním experimentu. PointNet se zlepšila z *9.40* na *8.38*, SSG z *9.38* na *8.01* u MSG došlo k menšímu zlepšení z *9.82* na *9.20* a z *9.25* na *8.90*, MVCNN se zlepšila z *9.52* na *6.68*, poslední síť Pointconv má, jako v předešlém případě, špatný odhad.

Vzhledem k tomu, že v prvním případě byla použita Pointconv na mračnu o 1024 bodech a v tomto experimentu na 512 bodech nelze říct, zda došlo ke zlepšení nebo zhoršení. Nicméně, i tak má síť stále špatné výsledky.

Ze sítí zpracovávající mračno bodů je v tomto experimentu nejlepší PointNet++ SSG, která zlepšila odhad o *1.37* MAE. Nejlepší sítí celkově je MVCNN, zpracovávající pohledy, která se zlepšila o 2.84 MAE na *6.68* MAE.

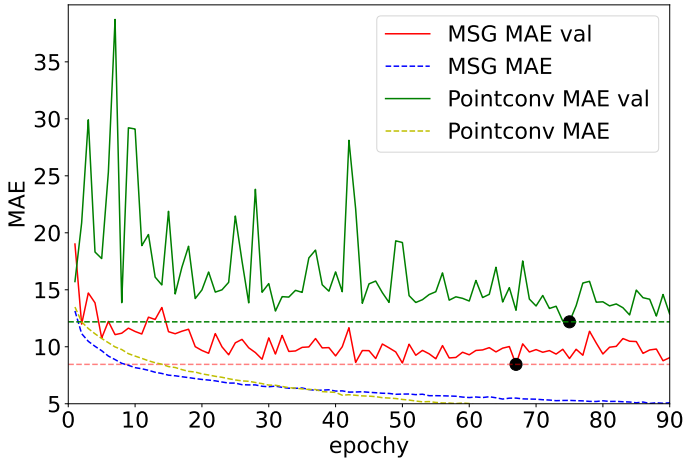
Na grafu 5.7 je znázorněn průběh trénování modelů PointNet a PointNet++ SSG pro fold 0. V tomto konkrétním případě dosáhla PointNet lepšího výsledku. PointNet konverguje rychleji a nejnižšího validačního MAE dosahuje mezi 40. a 45. epochou, poté se začíná přeučovat, validační



Obrázek 5.7 Na obrázku je zobrazeno porovnání průběhu validačního (val) a trénovacího MAE sítě PointNet++ SSG pro velkosti vstupu 512 bodů a sítě PointNet o 1024 bodech společně se zobrazeným nejmenším MAE, kterého sítě dosáhly v podobě tmavých bodů.

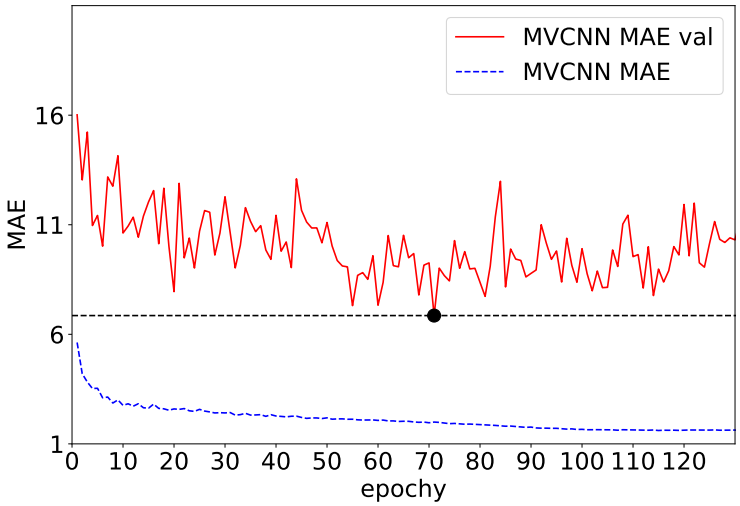
MAE roste, zatímco trénovací stále klesá. SSG model konverguje déle a nejnižšího MAE dosahuje kolem 75. epochy, poté hodnota validačního MAE stagnuje.

Graf 5.8 zobrazuje průběh trénování sítí Pointconv a PointNet++ MSG na foldu 0. Obě sítě mají pozvolnou konvergenci a síť MSG dosahuje minima v 67. epoše, Pointconv 77. epoše. Ani jedna ze sítí se nepřeučuje, ale hodnota trénovacího MAE je velmi nízká.



Obrázek 5.8 Ukázka průběhu trénovacího a validačního MAE (val) sítě PointNet++ MSG pro velikost vstupu 512 bodů a sítě Pointconv. Tmavý bod zobrazuje nejmenší dosažené MAE příslušné sítě. Obě sítě mají velmi nízké trénovací MAE.

Průběh trénování sítě MVCNN na foldu 0 je vidět na grafu 5.9. Nejlepšího MAE dosahuje kolem 70. epochy a následně se začíná přeučovat. Oproti předchozímu experimentu si můžeme všimnout, že rozšířená množina zároveň snížila míru oscilace validačního MAE během trénování.



Obrázek 5.9 Ukázka průběhu a přeučení sítě MVCNN na validačním (val) a trénovacím (mvcnn mae) MAE foldu 0. Tmavý bod označuje místo s nejmenším dosaženým MAE.

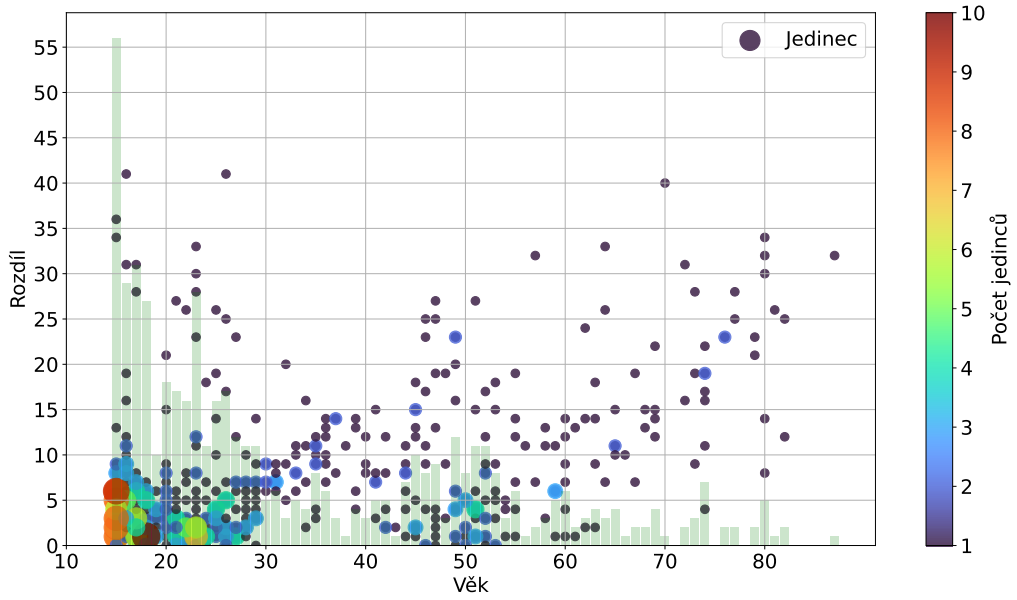
U sítí nedochází k výraznému přeučení kromě sítě MVCNN. Modely při trénování oscilují kolem nejnižší dosažené hodnoty validačního MAE, které se nezhoršuje. Nicméně, trénovací MAE se pohybuje kolem hodnoty 1.

5.2.4 Vyhodnocení výsledků na velkých datech

Nejlépe dopadla síť MVCNN, která dosáhla chybovosti *6.68* MAE. Tento model použijeme pro odhad věku na jednotlivcích ve validační množině. Protože jsme využili křížovou validaci, můžeme provést predikci věku na validační množině každého foldu a tedy nad každým jedincem.

V této části budeme pozorovat v jakých věkových kategorií model nejvíce chybuje, v jakých predikuje největší chyby a zda lze určit např. věkové intervaly, ve kterých k těmto chybám dochází.

Na obrázku 5.10 je zobrazena vzdálenost odhadu provedená modelem MVCNN od její skutečné hodnoty pro všechny věkové kategorie od 15-87 let.



Obrázek 5.10 Body představují jedince se stejnou chybou odhadu vůči skutečné hodnotě pro daný věk. Barva a velikost bodu reflektují počet jedinců se stejnou chybou v daném věku. Zelené sloupce na pozadí odpovídají distribuci jedinců v daném věku. Největší hustota bodů je v intervalu 10-30 let, kde velká část spadá pod chybovost 5 let.

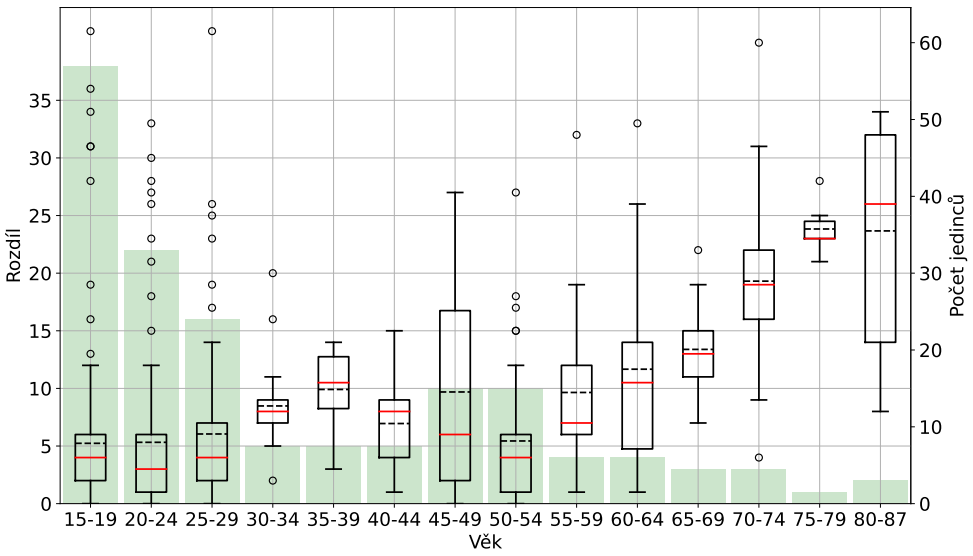
Nejvíce jedinců se nachází v intervalech od 15 do 30 a kolem věku 50. Především v intervalu od 15 do 30 let je většina jedinců odhadnuta do chybovosti 10 let a výrazně méně jedinců je odhadnuto nad touto hranicí. Mnoho jedinců se nachází dokonce pod hranicí chybovosti 5 let.

Od 30 let se zvyšuje chybovost, pod 5 lety se skoro žádní jedinci nevyskytují. Většina jedinců v těchto intervalech jsou odhadování s chybovostí mezi 5-15 lety.

Se vzrůstajícím počtem zástupců kolem 50 let se snižuje i chybovost. Po 55 roce je odhad již velmi špatný, málo zástupců se chybou dostane pod 10 let, po 70 roce pod 15 let.

Na grafu je 5.11 jsou znázorněny kvartily věkových intervalů vytvořených na základě 5.10. Model odhaduje s velkou přesností jedince z věkové hranice 15-29, na druhou stranu má několik outlierů, jejichž chybovost se pohybuje kolem 30 let.

S rostoucím věkem roste i chybovost, kromě zástupců z intervalu 50-54 let. Nejhorší výsledky mají jedinci starší 60 let, kde většina zástupců má chybovost nad 10 let. Věkové intervaly 45-49, 60-64 a 80-87 mají velký rozptyl chybovosti.

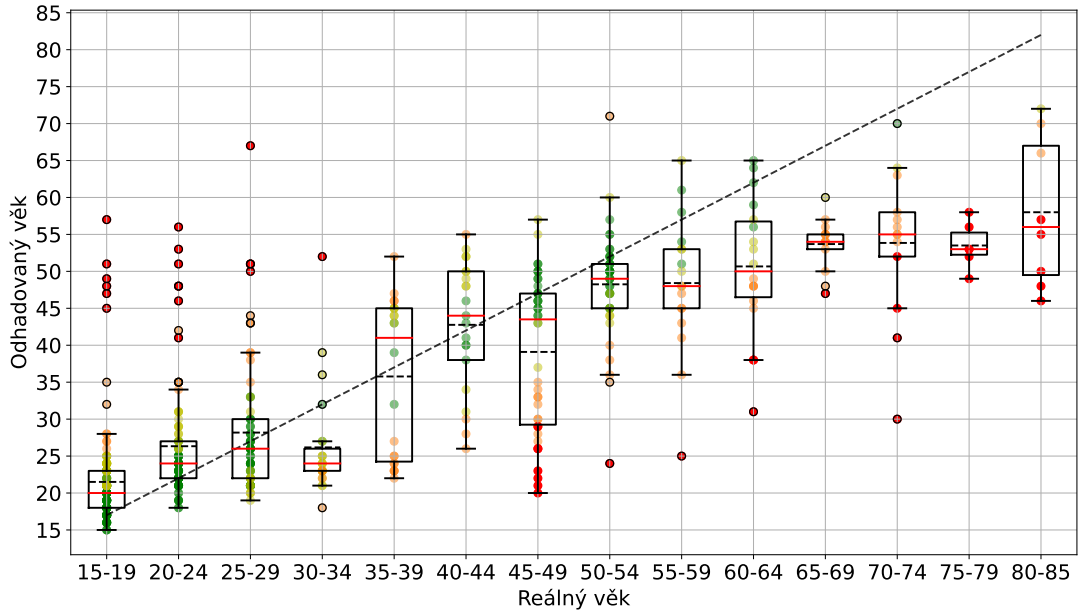


Obrázek 5.11 Krabicový graf zobrazující kvartily a outliery jednotlivých věkových intervalů posuzující rozdíly mezi odhadem a skutečným věkem jedinců z příslušných intervalů. Sloupce v pozadí demonstrují poměr zastoupení jedinců v daném intervalu. Červená úsečka představuje medián, černá čárkovaná střední hodnotu. K nejmenší chybovosti dochází ve věkovém intervalu 15-30 let, nicméně zde leží i několik outlierů s vysokou chybovostí.

Na grafu 5.12 jsou znázorněny konkrétní hodnoty odhadu jedinců z věkových intervalů. Černá čárkovaná úsečka představu medián příslušného intervalu např. pro interval 15-19 se jedná o hodnotu 17. Body v grafu představují odhad věku jednoho zástupce. Zelená barva bodu představuje chybovost věku menší než 5 let, žlutá menší než 10 let, oranžová nižší než 20 let a červená vyšší než 20 let.

Mladším jedincům modely spíše přisuzují věk o něco vyšší, než je jejich reálný. S rostoucím věkem model spíše odhaduje věk nižší než je reálný, nejvíce je tomu pak u jedinců starších 65 let, kde je zároveň odhad velmi nepřesný a většina odhadnutých jedinců má chybovost větší než 20 let.

Hodně zástupcům z intervalu 45-59 let je odhadován věk kolem 20 let, na druhou stranu jen v málo případech je věk vyšší než skutečný.



Obrázek 5.12 Krabicový graf zobrazující kvartily a outliery jednotlivých věkových intervalů vůči odhadnutému věku. Černá čárkovaná úsečka představuje medián intervalu. Červená úsečka představuje medián odhadu, černá čárkovaná střední hodnotu. Červené body značí jedince, jejichž chybovost přesáhla 20 let. Starší jedinci po 55 roce jsou podhodnocováni.

Shrnutí dosažených výsledků

V kapitole jsme porovnávali modely neuronových sítí, které zpracovávají skeny lidských tváří a odhadují z nich věk zachycených jedinců. Využili jsme k tomu 2 přístupy, pomocí mračen bodů zachycující bodovou strukturu, a pomocí různých pohledů na sken zohledňující texturu. Chybu predikce jsme měřili pomocí MAE.

Ukázalo se, že věk nejlépe odhadoval model MVCNN, který analyzoval dvojrozměrné pohledy, jeho průměrná chybovost dosahovala *6.68* let. Z modelů analyzující mračna bodů dopadl nejlépe model PointNet++ SSG s průměrnou chybovostí *8.01* let.

Chybovost odhadu věku záleží na stáří jedinců. Mladší jedince bylo možné odhadnout přesněji, jedná se o věkové kategorie od 15 do 30 let, u špatných odhadů byl zástupcům přisuzován větší věk spíše než nižší.

Starší osoby byly odhadovány s horší přesností, konkrétně se jednalo o osoby starší 65 let, kde chyba ve většině případů dosahovala 10 a více let. Těmto zástupcům se ve všech případech přisoudil nižší věk, než byl reálný.

Velikost chyby odhadu příslušného věku byla závislá na počtu zástupců. Pro více zástupců byl odhad přesnější, pro méně zástupců se přesnost snižovala. Tento fakt jsme se snažili vykompenzovat vybalancováním jedinců pomocí duplicit.

V následující kapitole se budeme zabývat, jaký vliv má pohlaví na odhad provedený z trojrozměrných dat. Využijeme k tomu modely PointNet, PointNet++ SSG a MVCNN, které měly buď nejlepší výsledky (SSG, MVCNN) anebo jsou rychlé při trénování.

Kapitola 6

Návrh modelu predikující věk na základě pohlaví

Podle studie [10] má stáří a pohlaví významný vliv na tvar obličeje. Tento fakt zohledníme v tomto experimentu. Experiment se dělí na dvě části.

V první části vybalancujeme datovou množinu podle věku i pohlaví. Vytvoříme několik modelů PointNet, které budou zohledňovat pohlaví a porovnáme výsledky.

V druhé části použijeme PointNet++ SSG a MVCNN, které na dostatečně rozšířené množině vyhodnotíme nejdříve bez informace o pohlaví a následně ji explicitně dodáme do modelu.

Věk budeme odhadovat pomocí regrese. Jako ztrátovou funkci k trénování využijeme MSE a jako metriku pozorování použijeme MAE validační množiny.

6.1 Experiment na množině vybalancované podle věku i pohlaví

V tomto experimentu je využita síť PointNet na mračnu velikosti 1024 bodů. Pomocí této sítě odvodíme modely, které budou zohledňovat pohlaví a vyhodnotíme výkonnost těchto modelů.

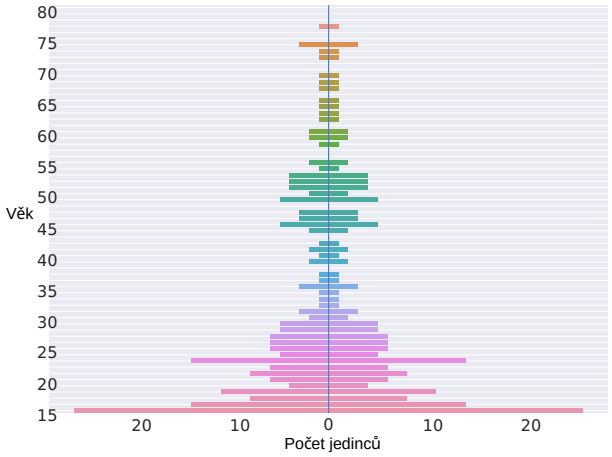
6.1.1 Rozdělení datové množiny

Datová množina má nerovnoměrné rozdělení mužů a žen v daných letech. Data vybalancujeme podle pohlaví. Pro každý věk určíme minimum počtu mužů a žen a přebytečné muže, resp. ženy z datové množiny odstraníme. V případě, že věk nemá ani jednoho zástupce daného pohlaví, věk z datové množiny odstraníme. K vybalancování využijeme třídu *DatasetBalancer* z modulu *data_reader*. Výsledné vyrovnání je možné vidět na 6.1.

Množinu rozdělíme na validační a trénovací. Validační množinu vytvoříme tak, že z každého věku odebereme jednoho zástupce obou pohlaví, pokud je v daném pohlaví pouze jeden zástupce, necháme jej v trénovací množině.

Trénovací množinu vybalancujeme nastavením parametru *max_one* na hodnotu 4. To nám zajistí, že každý jedinec bude pro doplnění využit nejvíce čtyřikrát, abychom do množiny nezavedli nějakého jedince vícekrát, než je potřeba, což by mohlo mít špatný dopad na finální odhad.

Zároveň vykompenzujeme data, která jsme ztratili balancováním množiny podle pohlaví tak, že každého jedince nakopírujeme nastavením parametru *least_samps* na hodnotu 5. Tím získáme

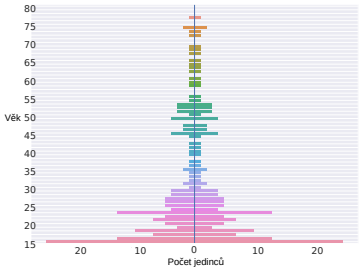


Obrázek 6.1 Vybalancovaná datová množina podle pohlaví a věku.

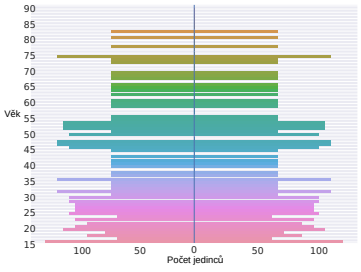
trénovací množinu o velikosti 4848 vzorků, kterou validujeme na 66 vzorcích. Vybalancovanou trénovací množinu podle pohlaví lze vidět na 6.2a a podle pohlaví a věku na 6.2b.

Pro experiment máme malou datovou množinu, kterou jsme rozdělili na trénovací a validační množinu. K dispozici nemáme testovací množinu, kterou bychom modely adekvátně ohodnotili. Otestování provedeme tak, že postupem popsaným výše vytvoříme tři různé trénovací množiny, kterým odpovídají příslušné validační množiny.

Ve všech validačních množinách může být obsažen stejný vzorek, pokud v příslušném věku není dostatek zástupců, v opačném případě se stejné vzorky ve validačních množinách nevyskytují. Tímto postupem oddělíme nakopírované jedince tak, aby se nevyskytovali kopie zároveň v trénovací a validační množině.



(a) Trénovací množina vybalancovaná podle pohlaví a věku, originální bez doplnění.



(b) Trénovací množina vybalancovaná podle pohlaví a věku a doplněná o duplikace.

Obrázek 6.2 Trénovací množina použitá pro experiment s pohlavím vybalancovaná podle pohlaví a věku.

6.1.2 Odhad věku na vybalancované množině podle pohlaví a věku

- V experimentu zkoumáme, zda je pohlaví jedním z faktorů, podle kterého lze zlepšit odhad věku. Vytvoříme *referenční model* (RM), který bude odhadovat věk bez zohlednění pohlaví, jehož výkonnost následně porovnáme s modely, které věk zohledňují.
- Nejdříve prozkoumáme, zda odhad provedený nad zástupci jednoho pohlaví je lepší než odhad provedený referenčním modelem. To provedeme tak, že z množiny odfiltrujeme muže, resp. ženy a model natrénujeme nad těmito množinami. Vytvoříme tak *mužský* (MM) resp. *ženský* (ŽM) model. Vzhledem k provedenému balancování na začátku experimentu, budou oba modely, odhadující pouze ženy, resp. pouze muže, trénované nad stejným množstvím dat. Nakonec vyzkoušíme chybovost odhadu modelu ŽM na množině, ve které jsou pouze muži a naopak.
- V dalším modelu již zahrneme informaci o pohlaví, kterou do modelu vložíme explicitně (EXT). Model bude mít na vstupu mračno bodů a pohlaví zakódované pomocí one-hot kódování. Tato informace bude připojena ke klasifikátoru provádějící odhad.
- Poslední model bude komplexní model (KM) složen z modelů, které odhadovali věk pouze pro jedno pohlaví, tj. MM a ŽM. Odhad obou modelů bude agregován a klasifikován finálním klasifikátorem v podobě hustě propojené vrstvy.

6.1.3 Výsledky jednotlivých modelů

Jednotlivé modely byly natrénované na 3 různých množinách. Nejmenší dosáhnuté MAE množiny, která trénovací množinu validovala je vidět v tabulce 6.1 společně se střední hodnotou těchto výsledků.

MAE *RM* je 8.87. Modely, *MM* resp. *ŽM*, které byly trénované pouze na mužských zástupcích, resp. ženských, dosáhly hodnoty 8.49 resp. 8.58, tedy přesnost těchto modelů se zlepšila o 0.3 resp. 0.21, pokud výsledky těchto modelů agregujeme dohromady pomocí *KM*, dosáhneme mírného zlepšení o hodnotu 0.19, tedy z hodnoty 8.87 na 8.62 oproti RM.

Menšího zlepšení dosáhneme i v případě, že do modelu explicitně zahrneme informaci o pohlaví *EXT*, kde jsme dosáhli hodnoty 8.60, což byla i nejlepší dosažená hodnota z modelů odhadující věk na celé množině.

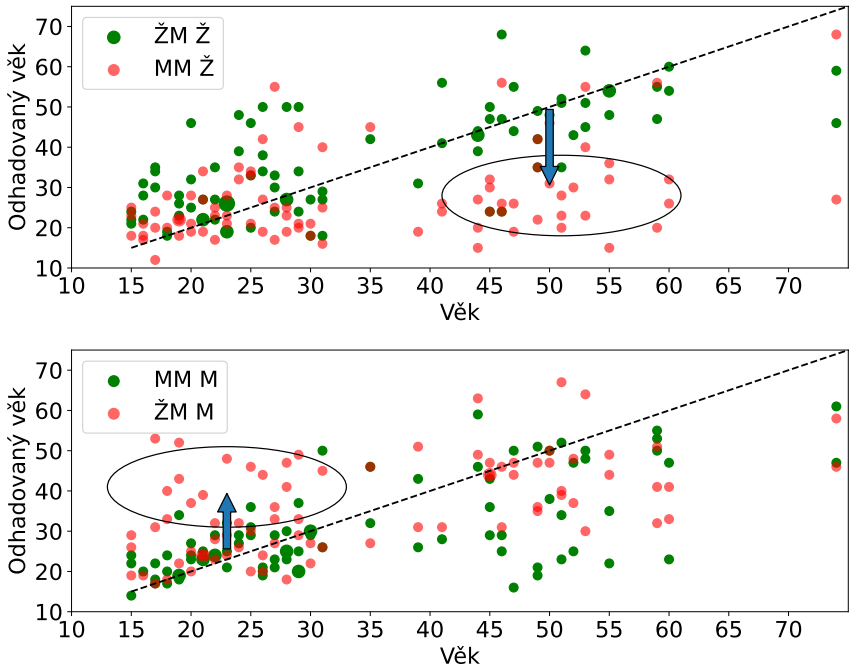
Tabulka 6.1 Výsledky modelů trénovaných na vybalancované množině podle pohlaví a věku na 3 foldech. Nejmenší dosažené MAE každého foldu je zobrazeno ve sloupci val. MAE mn. Průměr minim je zobrazen ve sloupci průměr val. MAE. RM je referenční model trénovaný na celé množině. EXT je model, do kterého je explicitně zavedena informace o pohlaví. MM je model trénovaný na mužích. ŽM je model trénovaný na ženách. M odpovídá vyhodnocení na množině, ve které jsou pouze muži, Ž naopak množině, ve které jsou pouze ženy. A je množina, ve které jsou muži i ženy. KM je komplexní model, který spojuje MM a ŽM. EXT A model dosáhl nejlepšího výsledku.

Typ modelu a vyhodnocení	val. MAE mn	průměr val. MAE
RM A	8.71, 8.25, 9.41	8.79
MM M	8.13, 8.97, 8.36	8.49
ŽM Ž	8.14, 8.64, 8.97	8.58
MM Ž	11.83, 13.85, 14.75	13.48
ŽM M	12.47, 9.53, 10.48	10.82
EXT A	8.87, 8.08, 8.85	**8.60**
KM A	8.36, 8.97, 8.54	8.62

Zohledněním pohlaví v obou případech, jsme dosáhli lepšího výsledku oproti RM, naopak když jsme vyhodnotili muže na ŽM a ženy na MM chybovost odhadu narostla. Na grafu 6.3 vidíme změnu odhadu mužů na ŽM oproti MM resp. žen na MM oproti ŽM.

Na horním grafu je zobrazena predikce žen na ŽM oproti predikci žen na MM. Model natrénovaný na mužích u starších žen mezi lety 40-60 odhaduje ženy jako mladší, než je jejich skutečný věk. Podobná situace nastává i u mladších žen, kde ale není tolik výrazná.

Na dolním grafu máme opačnou situaci, kde muže odhadujeme pomocí MM, resp. ŽM. ŽM posuzuje mladší muže v letech 15-30 podobně jako muže v letech 45-60. Model zvyšuje věk mladším jedincům i starším jedinců, pokud situaci posuzujeme oproti MM.

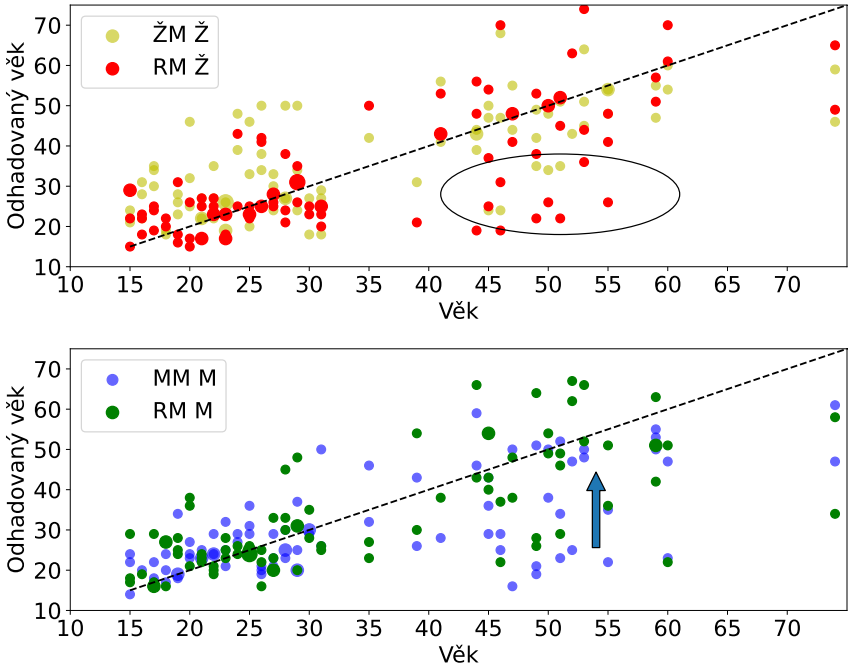


Obrázek 6.3 Porovnání odhadu ŽM na ženách (ŽM Ž) a MM Ž na ženách (MM Ž) a MM na mužích (MM M) a ŽM na mužích (ŽM M). Model natrénovaný na mužích MM podhodnocuje starší jedince, ŽM nadhodnocuje mladší jedince.

Model trénovaný na mužích přisuzuje starším mužům nižší věk, než je skutečný. Tedy model nenachází dostatek vzorů v mužské populaci, aby adekvátně odlišil starší od mladších. Což může být zapříčiněno tím, že se změny u mužů ve vyšším věku projevují méně výrazně.

Naopak ženy jsou ve starším věku odhadnuti s malou chybou. Tedy model nalezl vzory, podle kterých je schopen oddělit starší od mladších, a změny ve stáří se u žen projevují více.

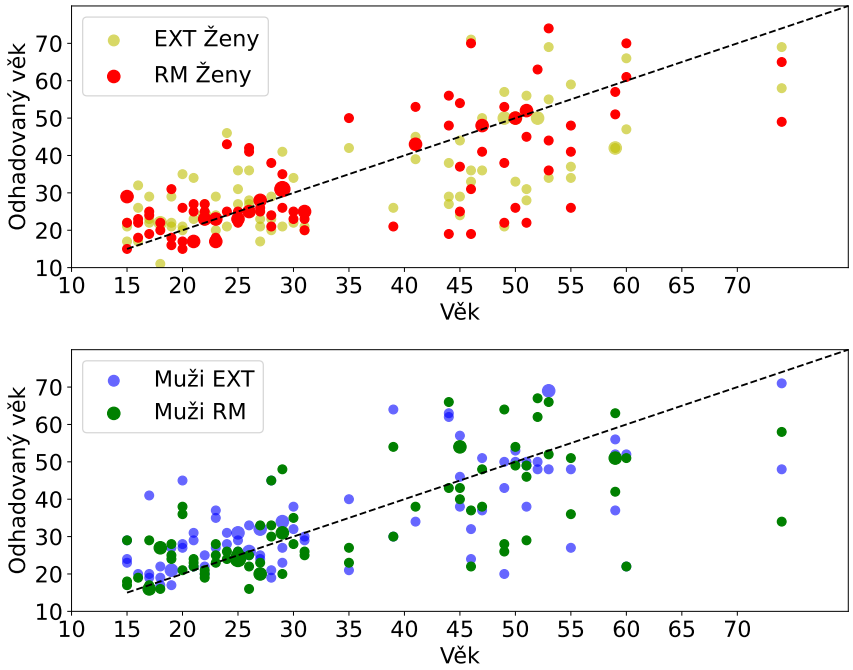
Na grafu 6.4 jsou znázorněny odhady *RM* na ženské a mužské množině oproti odhadu *MM* resp. *ŽM* modelu.



Obrázek 6.4 Porovnání odhadu ŽM na ženách (ŽM Ž) a referenčího modelu odhadující ženskou populaci (RM Ž) a odhad MM na mužích (MM M) v porovná s referenčním modelem RM odhadující populaci mužů (RM M). RM více podhodnocuje starší i mladší ženy oproti ŽM, naopak oproti MM nadhodnocuje starší muže.

Referenční model lehce vylepšuje odhad mužů u starších jedinců, kteří byli odhadnuti jako mladí, a naopak některé jedince v intervalu od 45-60 let nadsazuje více než mužský model. U žen naopak starším jedincům věk snižuje.

Graf 6.5 porovnává referenční model s modelem zohledňující pohlaví. Zohledněním pohlaví v modelu EXT jsou převážně starší ženy posunuty blíže k jejich skutečnému věku. Mladší muži jsou naopak lehce nadhodnoceny v mladších kategoriích.



Obrázek 6.5 Porovnání odhadu EXT na ženách (EXT Ž) a referenčního modelu odhadující ženskou populaci (RM Ž) a odhad EXT na mužích (EXT M) v porovná s referenčním modelem RM odhadující populaci mužů (RM M). RM více podhodnocuje starší i mladší ženy oproti EXT, naopak EXT lehce nadhodnocuje mladší muže.

Obecně jsou ženy spíše nadhodnoceny a muži podhodnoceni. V případě explicitního zavedení informace o pohlaví jsou pak modely schopny lehce redukovat tyto dysbalance a zlepšit odhad.

6.2 Experiment na množině vybalancované podle věku

V předešlém experimentu jsme na vybalancované množině podle pohlaví a věku porovnávali modely, které zohledňovaly pohlaví s referenčním modelem, který pohlaví nezohledňoval. Ukázalo se, že pohlaví má drobný vliv *0.19* MAE zlepšení.

V tomto experimentu porovnáme síť MVCNN posuzující texturu skenu s modelem PointNet++ SSG zpracovávající mračna bodů. Vytvoříme nejdříve referenční modely, které porovnáme s modely zohledňující pohlaví.

6.2.1 Návrh experimentu

Množinu budeme balancovat pouze podle pohlaví. Nastavíme parametr *max_one* na hodnotu 4 a *least_samps* na hodnotu 5 obdobně jako v předešlém experimentu.

Nebalancujeme množinu podle pohlaví, proto máme o několik jedinců více. V trénovací množině je celkem 5888 zástupců a ve validační 91. Opět používáme křížovou validaci o 6 foldech.

Ztrátová funkce je MSE. Metrika úspěšnosti MAE. Využíváme PointNet++ SSG na velikosti mračen 512 a MVCNN se standartní velikostí pohledu.

6.2.2 Výsledky jednotlivých modelů

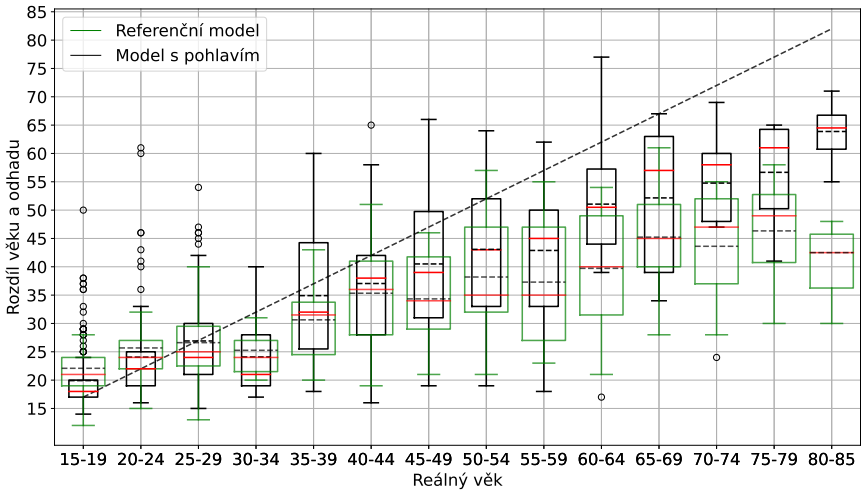
Chybovost jednotlivých foldů a celkovou chybovost je vidět v tabulce 6.2. V obou případech dochází ke zlepšení. Pokud zahrneme pohlaví u mračna bodů dostaneme zlepšení o *0.83* a u pohledů dokonce o *1.27*.

Tabulka 6.2 Výsledky referenčních modelů trénovaných na celé množině oproti stejným modelům zohledňující pohlaví. Ve sloupci Val. MAE mn. jsou ukázány nejmenší hodnoty MAE každého foldu, sloupec Val. MAE zobrazuje průměr těchto minim. U obou sítí došlo ke zlepšení, pokud jsme v modelu zohlednili pohlaví.

Typ modelu	Val. MAE mn.	Val. MAE
Referenční SSG	9.49, 8.68, 9.98, 8.96, 8.16, 8.65	8.99
SSG s pohlavím	7.89 , 8.21, 8.14, 7.26, 7.93, 9.56	8.16
Referenční MVCNN	8.53, 8.93, 8.51, 10.55, 8.85, 9.63	8.84
MVCNN s pohlavím	7.57 , 7.28, 8.40, 6.88, 6.90, 8.39	**7.57**

Na grafu 6.6 lze vidět posun jednotlivých krabicových grafů modelu MVCNN zohledňující pohlaví blíže k čárkované přímce, která reflektuje skutečnou hodnotu věku, oproti modelu nezohledňující pohlaví v podobě zelených krabicových grafů a tedy graf zobrazuje viditelné zlepšení. Nicméně v některých věkových kategorií jako je 35-39 let nebo 50-54 let se u zahrnutí pohlaví zvyšuje rozptyl hodnot.

Velký posun blíže ke skutečnému věku je vidět u kategorie 80-85 let. Naopak zhoršení nastává v mladší věkové kategorii 30-34 let, kde se zvýší rozptyl a medián se posune dále od referenční přímky.



Obrázek 6.6 Zelené krabicové grafy zobrazují odhad věku referenční modelu MVCNN. Černé krabicové grafy zobrazují odhad modelu MVCNN, který explicitně zohledňuje pohlaví. Čárkovaná přímka zobrazuje skutečný věk. Černé krabicové grafy se přibližují blíže k přímce značící lepší odhad věku.

Shrnutí

V kapitole jsme se zabývali, zda má pohlaví vliv na odhad věku z trojrozměrného skenu. Z experimentů vyšlo, že pohlaví má pozitivní vliv na odhadovaný věk, a to jak u reprezentace skenu v podobě mračna bodů, tak u modelu posuzující pohledy a zohledňující textury. Navíc se ukázalo, že větší dopad má zohlednění pohlaví u modelu posuzující pohledy.

Ukázalo se, že modely natrénované na pouze mužských, resp. ženských jedincích a následně aplikované na opačná pohlaví v případě mužských modelů podhodnocují převážně starší ženy a v případě ženských modelů nadhodnocují muže.

V následující kapitole se budeme zabývat přesností určování věkových hranic, přesněji věkové hranice 15, 18 a 40 let.

Kapitola 7

Návrh modelu na základě věkových kategorií

Posledním cílem práce je zaměřit se na mladší jedince v rozmezí 15-18 let. V kapitole se budeme zabývat klasifikací věkových hranic, tj. zda jsou jedinci mladší (včetně) nebo starší 15 resp. 18 let a porovnáme rozdíly mezi věkovou hranicí 40 let.

Dále vytvoříme komplexní model, který bude zohledňovat jednotlivé věkové hranice a budeme pozorovat míru chybovosti, podle které posoudíme, zda odlišení věkových hranic bude mít vliv na celkový odhad.

7.1 Klasifikace jedinců

V experimentech určených na učení věkové hranice jedinců používáme binární klasifikaci. Využíváme ztrátovou funkci *kategorickou křížovou entropii* a metriku *míru přesnosti*. Modely jsou také posuzovány pomocí metrik *precision*, *recall* a F_1.

Nejdříve trénujeme rozpoznávání věkových hranic na sítích PointNet, PointNet++ SSG a MVCNN. Tento experiment z časových důvodů provádíme pouze na 4 z 6 foldů, které byly použity v 6.2.1. U každého foldu bereme maximální míru přesnosti, a průměrujeme je dohromady.

Dále na šesti foldech natrénujeme dva nejlepší zástupce a vyhodnotíme jejich úspěšnost na všech 6 foldech. Zároveň dosažené výsledky porovnáme pomocí metrik *precision*, *recall* a F_1.

7.1.1 Odhad věkové hranice 15 let

K určení, zdali je jedinci 15 let nebo je starší 15 let použijeme model PointNet. V dostupné datové množině jsou jedinci pouze věku 15 let nebo starší. Natrénovaný model na takové datové množině dosahuje míry přesnosti kolem *92%* na všech foldech.

Problém je, že kvůli nepřítomnosti zástupců mladších 15 let model klasifikuje všechny zástupce jako starší 15 let. Dosahuje vysoké přesnosti i přes fakt, že žádného jedince neklasifikuje jako 15letého, precision je rovno 0.

Cíl práce u věkové hranice 15 let z důvodů podoby poskytnuté datové množiny nelze splnit, proto se v dalších sekcích zaměříme pouze na klasifikaci 18leté hranice a tu porovnáme s hranicí 40 let.

Volba věkové hranice 40 let byla zvolena na základě výsledků zobrazených v 6.6, kde se přesnost odhadu věku u starších jedinců začíná odklánět od skutečné hodnoty pro většinu odhadovaných jedinců.

7.1.2 Odhad věkových hranic 18 a 40 let

Hranice určujeme pomocí modelů PointNet, PointNet++ SSG a MVCNN na 4 foldech. Výsledky jednotlivých modelů jsou zobrazeny v tabulce 7.1.

Tabulka 7.1 Tabulka zobrazuje typ modelu společně s věkovou hranicí u které odhaduje, zda je jedinec mladší (včetně) nebo starší, modely jsou trénované na 4 foldech. Věkové hranice jsou 18 a 40. Sloupec val. Acc zobrazuje maximální míru přesnosti, kterou klasifikace dosáhla u příslušného foldu. Průměr těchto maxim je zobrazen ve sloupci p. val. Acc. Nejlepších výsledků dosáhla síť MVCNN.

Typ modelu	val. Acc	p. val. Acc
PointNet 18	0.85, 0.86, 0.90, 0.86	0.87
PointNet 40	0.82, 0.81, 0.88, 0.84	0.84
SSG 18	0.86, 0.85, 0.85, 0.90	0.87
SSG 40	0.83, 0.85, 0.88, 0.89	0.86
MVCNN 18	0.88, 0.91, 0.94, 0.93	**0.92**
MVCNN 40	0.93, 0.91, 0.88, 0.94	**0.92**

Podobně jako v předcházejících experimentech největší přesnosti dosahuje síť MVCNN a to až *92*% jak pro určení věkové hranice 18 let, tak určení věkové hranice 40 let. V některých foldech dosahují i přesnosti *94*%.

Modely zaměřující se na mračna bodů klasifikují s přesností horší až o *8*% v případě sítě PointNet při klasifikaci věkové hranice 40 let s přesností *84*% a *6*% v případě sítě SSG s přesností *86*%. Obě sítě klasifikují 18 letou hranici shodně s přesností *87*%.

Věková hranice 40 let se v případě posuzování mračna bodů identifikuje hůře než věková hranice 18 let. MVCNN posuzující pohledy identifikuje obě hranice se stejnou přesností.

Vzhledem k podobné přesnosti sítí PointNet a SSG se při trénování na 6 foldech omezíme pouze na síť PointNet a MVCNN.

Pomocí křížové validace vyhodnotíme všechny jedince celé datové množiny na všech 6 foldech. Dosažené výsledky jsou vidět v tabulce 7.2. Míra přesnosti klesla u sítě MVCNN a to především pro odhad hranice 18 let.

Tabulka 7.2 Výsledky modelů PointNet a MVCNN pro věkové hranice 18 a 40 let trénované na 6 foldech. Sloupec val. Acc zobrazuje maximální míru přesnosti, kterou klasifikace dosáhla u příslušného foldu. Průměr těchto maxim je zobrazen ve sloupci p. val. Acc. Nejlepších výsledků dosáhla síť MVCNN.

Typ modelu	val. Acc	p. val. Acc
PointNet 18	0.84, 0.85, 0.85, 0.84, 0.91, 0.85	0.86
PointNet 40	0.80, 0.86, 0.82, 0.80, 0.82, 0.90	0.84
MVCNN 18	0.84, 0.92, 0.93, 0.85, 0.84, 0.93	0.89
MVCNN 40	0.89, 0.93, 0.91, 0.85, 0.88, 0.94	0.90

Dále v tabulce 7.3 jsou modely měřeny pomocí metrik precision, recall a f_1. Model MVCNN klasifikuje jedince mladší 18 let s velkou přesností, kde je hodnota precision *0.89* a 40 letou hranici s hodnotou *0.91*. U této sítě je vysoký i recall *0.85* resp. *0.91* a f_1.

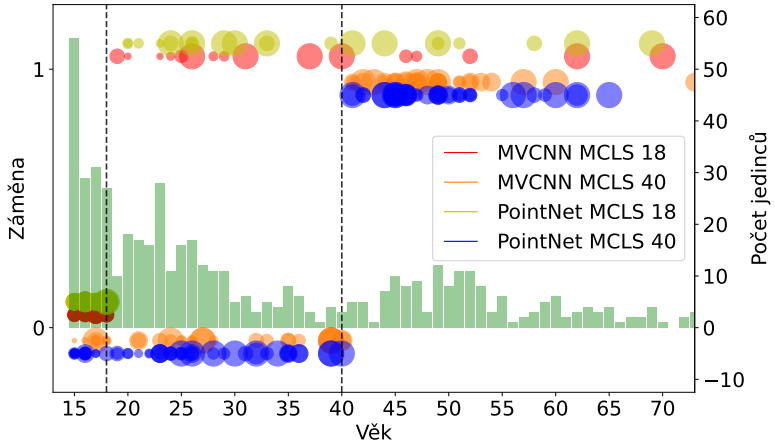
Oproti tomu má síť PointNet horší výsledky u obou věkových hranic a zejména hranici 18 let identifikuje s nízkou přesností, precision je *0.78*.

Na grafu 7.1 je vidět četnost záměny klasifikující hranici 18 a 40 let v každém věku pro obě pozorované sítě.

Jedinci starší mezi lety 40-60 se méně chybně klasifikují jako mladší 18 let, nicméně u let 63 a 70 dochází k záměně za mladší 18 let. K chybné klasifikaci jedince jako staršího 18 let, který je

Tabulka 7.3 Výsledky porovnání klasifikace věkových hranic pro metriky precision, recall a f_1. Síť MVCNN má v případě věkové hranice vysokou hodnotu všech metrik. Zároveň má pro příslušné hranice vyšší hodnoty ve všech metrikách oproti modelu PointNet.

Typ modelu	precision	recall	f_1
PointNet 18	0.78	0.54	0.64
PointNet 40	0.88	0.85	0.86
MVCNN 18	0.89	0.85	0.66
MVCNN 40	0.91	0.91	0.91



Obrázek 7.1 Na grafu jsou vidět všichni špatně klasifikovaní jedinci u obou věkových hranic. Velikost jednotlivých bodů odpovídá počtu jedinců v dané věkové kategorii, u kterých došlo k záměně. Hodnoty jsou převážené počtem všech jedinců v daném věku. MCLS 18 resp. 40 označuje věkovou hranici, která byla posuzována příslušnou sítí MVCNN nebo PointNet. Na pozadí je zobrazena distribuce jedinců v příslušných věkových kategorií.

mladší 18 let, dochází ve shodném poměru ve všech věkových kategorií mezi 15-17 lety v případě sítě MVCNN i sítě PointNet.

U osob kolem 40 let až k osobám starým 55 let dochází k záměně, po 60 roku už k záměně nedochází s výjimkou věku 80 pro věkovou hranici 40 let. K záměně za jedince starší 40 let dochází i u mladších osob kolem let 15-18, v případě MVCNN nejvíce v 17 letech, v případě PointNet rovnoměrně.

7.2 Komplexní model zohledňující věkové kategorie

V předchozím experimentu dokáže síť MVCNN rozpoznat věkovou hranici 18 let resp. 40 let s přesností 92%. Zbylé sítě analyzující mračna bodů shodně s 87% přesností pro 18 let resp. kolem 85% pro 40 let.

Tento fakt zohledňujeme v následujících experimentech. Používáme zde model odhadující věk pomocí regrese, kterému explicitně předáváme informaci o věkové hranici jedinců. Tento model porovnáváme s komplexní modelem, který interně využívá modely, které provádí odhad věkové hranice pomocí klasifikace. Tyto modely také porovnáváme s referenčním modelem, který rozdělení věkových hranic nezohledňuje.

Experiment je prováděn na stejné datové množině jako v předchozím případě, opět pouze na 4 z 6 foldů. Je využita ztrátová funkce MSE a metrika MAE.

Vzhledem k tomu, že modely PointNet a PointNet++ SSG dosahují při klasifikaci stejných odhadů a zároveň se model PointNet snadněji upravuje, omezíme se dále v kapitole pouze na model PointNet jako zástupce modelu, který pracuje s mračny bodů. Model budeme porovnávat s modelem MVCNN zpracovávající pohledy.

7.2.1 Referenční model

Pro každou ze sítí PointNet a MVCNN využíváme referenční model, pomocí kterého porovnáváme dopady komplexního modelu. Tento model je natrénovaný na stejné množině bez doplňujících informací na vstupu.

7.2.2 Model s explicitně přidanými věkovými hranicemi

Model je implementován pomocí sítě PointNet, na vstup sítě jde mimo mračno bodů i informace o jedinci, zda je mladší nebo starší 18 let resp. 40 let. Tato informace je zakódovaná pomocí one-hot kódování v podobě 2 nových příznaků.

Vstup s hranicemi je napojen do klasifikátoru sítě vytvořeného z hustě propojených vrstev (Dense), k již existujícímu klasifikátoru je přidána ještě jedna vrstva o 64 jednotkách, která zpracovává výstup klasifikátoru doplněného o informaci věkové hranice. Výstup této vrstvy jde do poslední výstupové vrstvy o jedné jednotce, která provádí finální regresi – odhad.

V tabulce 7.4 se porovnává chybovost tohoto modelu a referenčního modelu.

Tabulka 7.4 V tabulce jsou zobrazeny výsledky referenčního modelu (ref) oproti modelu, kterému jsou explicitně předané věkové hranice (ext). Val. MAE mn. zobrazuje nejmenší MAE dosažené v příslušném foldu, Val MAE průměr těchto minim. Explicitní zavedení mělo velký dopad na celkový odhad.

Typ modelu	Val. MAE mn.	Val MAE
PointNet ref.	8.41, 8.99, 7.96, 8.43	8.44
PointNet ext.	4.73, 4.21, 4.70, 5.02	4.67

Explicitně přidané věkové hranice výrazně pomáhají snižovat chybovost odhadu modelu, v tomto případě až o *3.77* MAE. V komplexním modelu místo přímého vložení věkových hranic tyto hranice odhadneme pomocí klasifikačních modelů.

7.2.3 Komplexní model

Komplexní model provádí odhad pomocí sítě PointNet nebo MVCNN. Klasifikátor modelu je shodný s klasifikátorem modelu s explicitně přidanými věkovými hranicemi s rozdílem, že místo explicitně vložených informací o věkové hranici tuto hranici odhadují, z mračna bodů resp. jednotlivých pohledů, natrénované modely provádějící klasifikaci ze sekce 7.1.2.

Tyto externě vložené modely jsou zmražené, tedy slouží pouze k odhadu věkové hranice a během trénování se jejich váhy nemění.

Celý model je tedy složen ze 3 stejných modelů PointNet resp. MVCNN, kde 2 modely jsou klasifikační a netrénují se, a 1 provádí regresi a trénuje se společně s klasifikátorem.

Počet parametrů u modelu složeného ze sítě PointNet se pohybuje kolem 2 100 000, přičemž se trénuje kolem 700 000 parametrů. U sítě MVCNN je to kolem 30 000 000 parametrů, kde se trénuje kolem 10 000 000 parametrů.

Modely se trénují na 4 foldech, které jsou shodné s foldy, na kterých se trénovaly klasifikační modely. Takto se v modelu zamezí pronikání jedinců příslušného foldu validační množiny do množiny trénovací jiného foldu.

7.2.4 Výsledky modelů zohledňující věkové kategorie

Ke snížení chybovosti došlo u sítě MVCNN, u které se chyba odhadu snížila o *0.29* na hodnotu *8.35* oproti referenčnímu modelu. U sítě PointNet nedošlo při použití komplexního modelu k žádnému zlepšení. Hodnota chyby odhadu zůstala na *8.44*. Výsledky jsou vidět v tabulce 7.5

Tabulka 7.5 V tabulce jsou zobrazeny výsledky referenčních modelů PointNet a MVCNN (ref.) oproti modelu, který interně provádí klasifikaci věkových hranic (kom). Val. MAE mn. zobrazuje nejmenší MAE dosažené v příslušném foldu, Val MAE průměr těchto minim. Komplexní model odhadoval lépe pouze u sítě MVCNN.

Typ modelu	Val. MAE mn.	Val MAE
PointNet ref.	8.41, 8.99, 7.96, 8.43	8.44
PointNet kom.	8.50, 8.91, 8.16, 8.20	8.44
MVCNN ref.	9.95, 8.53, 9.25, 7.63	8.84
MVCNN kom.	8.80, 7.69, 8.15, 8.76	**8.35**

Výkonnost modelů PointNet interně klasifikující věkové kategorie nejsou dostatečně přesné na to, aby byl model schopen z jejich odhadů vylepšit celkovou predikci a dosáhnout takového zlepšení, kterého model dosáhne v případě, že se věkové intervaly do modelu vloží explicitně.

Naopak přesnost modelů MVCNN klasifikující věkové kategorie 18 a 40 let jsou dostatečně přesné, aby se odhadnutý interval projevil.

Určitou roli může hrát trénovací míra přesnosti klasifikačních sítí během trénování komplexního modelu. Při trénování dochází k odhadům intervalů nad trénovacími daty, kde se klasifikují podle trénovací míry přesnosti, u modelů PointNet se jedná o cca 98% přesnost, kdežto u sítí MVCNN je trénovací míra přesnosti 100%.

Vzhledem k tomu, že u modelů PointNet tato přesnost dosahuje pouze 98% může vést nepřesnost k silnému šumu zavedenému během trénování, který snižuje odhad celé sítě.

Shrnutí

. V kapitole jsme se zaměřili na klasifikaci věkových hranic 15, 18 a 40 let. Ukázalo se, že z důvodu nedostatku dat v podobě jedinců mladších 15 let nelze věkovou hranici rozpoznat.

Věkovou hranici 18 let bylo možné rozpoznat s přesností *89*% v případě 40 leté hranice to bylo *91*%. Větší přesnost vykázala síť MVCNN, která posuzuje texturu trojrozměrného skenu než sítě posuzující mračna bodů.

Ukázalo se, že k záměně jedinců u jednotlivých věkových hranic dochází napříč celým věkovým spektrem, méně pak u starších jedinců u věkové hranice 18 let a u mladších jedinců u věkové hranice 40 let.

V neposlední řadě zohlednění věkové hranice v modelech odhadující věk vylepšuje celkový odhad u modelu, který byl vytvořen z modelů MVCNN. V případě modelu posuzující mračno bodů nemělo zohlednění věkových hranic klasifikovaných pomocí vložených sítí žádný dopad, pravděpodobně z důvodu slabší míry přesnosti sítě při klasifikaci věkové hranice.

Závěr

V práci jsem se zabýval odhadem věku z trojrozměrného skenu lidské tváře. Celkově se práce skládala ze tří cílů.

Prvním cílem bylo porovnat metodiky strojového učení vhodné pro zpracování trojrozměrných dat na odhadu věku ze skenu lidské tváře. Celkově jsem porovnával 5 různých architektur umělých neuronových sítí. Jednalo se o 4 sítě, které posuzovali mračno bodů PointNet, PointNet++, Pointconv, Xception a jednu síť, která posuzovala pohledy na sken z různých úhlů MVCNN.

Ukázalo se, že nejmenší chybovost odhadu věku měla síť MVCNN, která věk odhadovala z pohledů s průměrnou chybovostí 6.68 let. Nejlepší sítí, která věk odhadovala z mračna bodů byla síť PointNet++ ve verzi SSG s průměrnou chybovostí 8.01 let. Ukázalo se, že modely chybně odhadují především starší ročníky, což mohlo být zapříčiněno nerovnoměrným zastoupením jednotlivých věkových kategorií v datové množině.

Druhý cíl se skládal z posouzení dopadu pohlaví na celkový odhad a ze zohlednění věkových hranic 15 a 18 let. Dopad pohlaví na celkový odhad jsem testoval na sítích PointNet, PointNet++ SSG a MVCNN. Na všech sítí došlo ke snížení průměrné chyby odhadu. U sítě PointNet++ SSG posuzující mračna bodů to bylo o 0.83 let oproti referenčnímu modelu a u sítě MVCNN dokonce o 1.27 let.

Dále jsem posoudil věkové hranice 15 a 18 let. Z dostupné datové množiny nebylo možné navrhnout experiment, který by vhodně klasifikoval hranici 15 let, protože nebyla dostupná data jedinců mladších 15 let, proto jsem navrhl alternativní experiment s věkovou hranicí 40 let, kterou jsem zvolil na základě výstupů předcházejících experimentů. Ukázalo se, že modely jsou schopné určit věkovou hranici s mírou přesnosti 89% pro věkovou hranici 18 let a 90% pro věkovou hranici 40 let. Toto jsem zohlednil v modelu, který interně využíval sítě natrénované na klasifikaci těchto hranic a snížil celkovou chybu o 0.49 let.

Posledním cílem bylo porovnat modely pracující s trojrozměrnými strukturami s modelem, který zohledňuje texturu obličeje společně s trojrozměrnou strukturou. Tohoto jsem dosáhl pomocí sítě MVCNN, která posuzuje dvojrozměrné pohledy na sken. Ukázalo se, že síť zohledňující texturu dosahovala ve všech měřeních lepších výsledků než sítě, které pracovaly pouze s trojrozměrnou strukturou v podobě mračna bodů.

Práce vznikla ve spolupráci s katedru antropologie a genetiky člověka Přírodovědecké fakulty UK, která poskytla data a závěry zohlední ve svém výzkumu.

Budoucí práce

V práci jsem využil pouze jeden z modelů zohledňující texturu, v navazující práci by bylo možné porovnat modely, které dosahují lepších odhadů na trojrozměrných datech zpracovávající pohledy a porovnat, zda se celkový odhad nezlepší na lepší architektuře.

Také by bylo možné porovnat výsledky dosažené sítí MVCNN se sítí, která zpracovává pouze jeden pohled jako je např. sít Xception a porovnat, zda posuzování pohledů z více úhlů má pozitivní dopady na odhad věku anebo je odhad založený na pouze dvojrozměrné informaci dostatečný.

V neposlední řadě by bylo možné spojit model zpracovávající mračno bodů s modelem posuzující pohledy a prozkoumat dopady na odhad modelu založeného pouze na zpracování mračen bodů a na modelu zpracovávající pouze pohledy.

Bibliografie

1. SAWANT, Manisha M; BHURCHANDI, Kishor M. Age invariant face recognition: a survey on facial aging databases, techniques and effect of aging. In: [online]. Springer, 2019, sv. 52, s. 981–1008 [cit. 2022-04-23]. Č. 2. Dostupné z DOI: https://doi.org/10.1007/s10462-018-9661-z.
2. SVEIKATA, Kestutis; BALCIUNIENE, Irena; TUTKUVIENE, Janina et al. Factors influencing face aging. Literature review. In: [online]. 2011, sv. 13, s. 113–116 [cit. 2022-04-23]. Č. 4. Dostupné z: https://sbdmj.lsmuni.lt/114/114-02.pdf.
3. HUTTON, T.J.; BUXTON, B.F.; HAMMOND, P.; POTTS, H.W.W. Estimating average growth trajectories in shape-space using kernel smoothing. *IEEE Transactions on Medical Imaging* [online]. 2003, roč. 22, č. 6, s. 747–753 [cit. 2022-05-01]. ISSN 1558-254X. Dostupné z DOI: 10.1109/TMI.2003.814784.
4. ENLOW, Donald H; HANS, Mark G. *Essentials of facial growth* [online]. WB Saunders Company, 1996 [cit. 2022-05-01].
5. A.F. VALINSKIY, Kriminoloji. *Yuridicheskie nauki* [online]. 1999 [cit. 2022-04-23].
6. GUNAY, Asuman; NABIYEV, Vasif V. Automatic age classification with LBP. In: *2008 23rd International Symposium on Computer and Information Sciences* [online]. Istanbul, Turkey: IEEE, 2008, s. 1–4 [cit. 2022-04-24]. Dostupné z DOI: 10.1109/ISCIS.2008.4717926.
7. CHEN, Shixing; ZHANG, Caojin; DONG, Ming; LE, Jialiang; RAO, Mike. Using Ranking-CNN for Age Estimation. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* [online]. Honolulu, HI, USA: IEEE, 2017, s. 742–751. ISBN 978-1-5386-0457-1. Dostupné z DOI: 10.1109/CVPR.2017.86.
8. TAHERI, Shahram; TOYGAR, Önsen. On the use of DAG-CNN architecture for age estimation with multi-stage features fusion. *Neurocomputing* [online]. 2019, roč. 329, s. 300–310 [cit. 2022-04-23]. ISSN 0925-2312. Dostupné z DOI: https://doi.org/10.1016/j.neucom.2018.10.071.
9. ŠTERN, Darko; PAYER, Christian; GIULIANI, Nicola; URSCHLER, Martin. Automatic age estimation and majority age classification from multi-factorial MRI data. In: [online]. IEEE, 2019, sv. 23, s. 1392–1403 [cit. 2022-04-23]. Č. 4. ISSN 2168-2208. Dostupné z DOI: 10.1109/JBHI.2018.2869606.
10. MYDLOVA, Miriama; DUPEJ, Ján; KOUDELOVA, Jana; VELEMINSKÁ, Jana. Sexual dimorphism of facial appearance in ageing human adults: a cross-sectional study. In: [online]. Elsevier, 2015, sv. 257, 519–e1 [cit. 2022-04-23]. ISSN 0379-0738. Dostupné z DOI: https://doi.org/10.1016/j.forsciint.2015.09.008.

11. KOUDELOVÁ, Jana; HOFFMANNOVÁ, Eva; DUPEJ, Ján; VELEMINSKÁ, Jana. Simulation of facial growth based on longitudinal data: age progression and age regression between 7 and 17 years of age using 3D surface data. In: [online]. Public Library of Science San Francisco, CA USA, 2019, sv. 14, s. 1–16 [cit. 2022-04-23]. Č. 2. Dostupné z DOI: 10.1371/journal.pone.0212618.
12. FRANÇOIS, Chollet. *Deep learning v jazyku Python: knihovny Keras, Tensorflow* [online]. Grada Publishing, as, 2019 [cit. 2022-05-01]. ISBN 978-80-247-3100-1.
13. GUO, Yulan; WANG, Hanyun; HU, Qingyong; LIU, Hao; LIU, Li; BENNAMOUN, Mohammed. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence* [online]. 2020, roč. 43, č. 12, s. 4338–4364 [cit. 2022-04-23]. ISSN 1939-3539. Dostupné z DOI: 10.1109/TPAMI.2020.3005434.
14. SU, Hang; MAJI, Subhransu; KALOGERAKIS, Evangelos; LEARNED-MILLER, Erik. Multi-view convolutional neural networks for 3d shape recognition. In: *Proceedings of the IEEE international conference on computer vision* [online]. 2015, s. 945–953 [cit. 2022-04-23].
15. QI, Charles R; SU, Hao; MO, Kaichun; GUIBAS, Leonidas J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition* [online]. 2017, s. 652–660 [cit. 2022-04-23].
16. QI, Charles Ruizhongtai; YI, Li; SU, Hao; GUIBAS, Leonidas J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems* [online]. 2017, roč. 30 [cit. 2022-04-23].
17. YAN, Xu; ZHENG, Chaoda; LI, Zhen; WANG, Sheng; CUI, Shuguang. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* [online]. 2020, s. 5589–5598 [cit. 2022-04-23].
18. WU, Wenxuan; QI, Zhongang; FUXIN, Li. Pointconv: Deep convolutional networks on 3d point clouds. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* [online]. 2019, s. 9621–9630 [cit. 2022-04-23].
19. MARTÍN ABADI, et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems* [online]. 2015 [cit. 2022-05-04]. Dostupné z DOI: 10.5281/zenodo.4724125.
20. VAN ROSSUM, Guido; DRAKE, Fred L. *Python 3 Reference Manual* [online]. Scotts Valley, CA: CreateSpace, 2009 [cit. 2022-05-04]. ISBN 1441412697. Dostupné z: python.org.
21. CHOLLET, François. *Xception: Deep Learning with Depthwise Separable Convolutions* [online]. arXiv, 2016 [cit. 2022-05-04]. Dostupné z DOI: 10.48550/ARXIV.1610.02357.
22. WU, Zhirong; SONG, Shuran; KHOSLA, Aditya; YU, Fisher; ZHANG, Linguang; TANG, Xiaoou; XIAO, Jianxiong. 3D ShapeNets: A deep representation for volumetric shapes. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* [online]. Boston, MA, USA, 2015, s. 1912–1920 [cit. 2022-05-04]. ISBN 978-1-4673-6964-0. Dostupné z DOI: 10.1109/CVPR.2015.7298801.

Obsah přiloženého média

- *README.md* ..... popis obsahu média
- *data_utils*
  - *data_reader.py*
  - *inputs_MVCNN.py*
- *models* ..... implementace modelů neuronových sítí
- *Notebooks* ..... adresář obsahující jupyter notebooky s experimenty
- *pnet2_layers* ..... pomocné funkcionality
- *tf_ops* ..... pomocné funkcionality
- *data_splitter.py*
- *history_utils.py*
- *prepare_folds.py*
- *tf_cross_validation_form_files.py*
- *tf_cross_validation_MVCNN.py*
- *tfrec_utils.py*
- *to_tfrecord.py*
- *view_maker.py*
- *view_maker.py*
- *text* ..... adresář s textem práce