VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

GENEROVÁNÍ TRÉNOVACÍCH DAT POMOCÍ GAN PRO ODHAD VĚKU Z FOTOGRAFIE

GAN GENERATED DATA FOR CNN AGE ESTIMATION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE **Bc. TOMÁŠ VENKRBEC**
AUTHOR

VEDOUCÍ PRÁCE **Ing. MICHAL HRADIŠ, Ph.D.**
SUPERVISOR

BRNO 2022

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2021/2022

Zadání diplomové práce

Student:	**Venkrbec Tomáš, Bc.**
Program:	Informační technologie a umělá inteligence
Specializace:	Strojové učení
Název:	**Generování trénovacích dat pomocí GAN pro odhad věku z fotografie** **GAN Generated Data for CNN Age Estimation**
Kategorie:	Zpracování obrazu

Zadání:

Sítě typu GAN (Generative Adversarial Networks) dokáží generovat fotorealistické fotografie tváří s definovanými vlastnostmi (muž/žena, blond/brunet, mladý/starý, brýle/bez brýlí, ...) [1, 2]. Je možné pomocí GAN generovat tváře se specifikovanou věkovou kategorií? Zlepšily by takto vygenerovaná trénovací data přesnost supervised natrénované sítě pro klasifikaci věku z fotografie tváře?

1. Prostudujte základy konvolučních sítí a GAN pro generování obrazu tváří.
2. Vyberte si současnou metodu využívající GAN pro generování tváře (s dostupnou open source implementací), analyzujte její podstatu a možnosti ovlivnit výslednou věkovou kategorii.
3. Navrhněte úpravu metody, aby dokázala generovat tváře se specifikovanou věkovou kategorii.
4. Vyberte datové sady vhodné na experimenty (data může poskytnout spol. Innovatrics).
5. Pomocí vybrané GAN metody vygenerujte tváře různých věkových kategorií. Natrénujte supervised CNN klasifikátor na reálných tvářích a následně i na vygenerovaných. Vyhodnoťte vliv množství přidaných vygenerovaných dat na přesnost sítě.
6. Porovnejte dosažené výsledky a diskutujte možnosti budoucího vývoje.
7. Vytvořte stručné video prezentující vaši práci, její cíle a výsledky.

Literatura:

1. Karras, Tero, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

2. Lin, Ji, et al. "Anycost gans for interactive image synthesis and editing." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

Při obhajobě semestrální části projektu je požadováno:

- Body zadání 1 až 3.

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

Vedoucí práce:	**Hradiš Michal, Ing., Ph.D.**
Konzultant:	Beszédeš Marián, Ing., PhD., Innovatrics
Vedoucí ústavu:	Černocký Jan, doc. Dr. Ing.
Datum zadání:	1. listopadu 2021
Datum odevzdání:	18. května 2022
Datum schválení:	1. listopadu 2021

Abstrakt

Cílem této práce je implementace některé z nejmodernějších metod generativních neuronových sítí a návrh jejího rozšíření o podmíněné generování. To bylo využito pro generování fotorealistických snímků lidských tváří se specifikovanými charakteristikami, jako například věk a pohlaví. K tomuto účelu byla sloučením a čištěním existujících anotovaných datových sad obličejů vytvořena velmi různorodá datová sada, čítající přes 230 tisíc vzorků. Hojně jsou v ní zastoupeny všechny věkové kategorie, pohlaví a různé etnické skupiny. StyleGAN2 generátorem natrénovaným na této datové sadě bylo dosaženo hodnoty FID 7,14. S poměrem syntetických dat bylo následně experimentováno při trénování klasifikátoru věku. V případě testovací podmnožiny datové sady bylo přidáním syntetických dat docíleno snížení střední absolutní chyby z 3,499 roku na 3,294 roku. U nezávislé testovací datové sady došlo ke snížení průměrné chyby z 4,012 roku na 3,875 roku.

Abstract

The goal of this thesis is to implement one of the state-of-the-art methods of generative adversarial networks and to propose its extension to conditional generation. This has been used to generate photorealistic images of human faces with specified characteristics such as age and gender. For this purpose, a highly diverse dataset of over 230,000 samples was created by merging and cleaning existing annotated face datasets. All ages, genders and different ethnic groups are well represented in it. StyleGAN2 generator trained on this dataset achieved a FID of 7.14. The synthetic data ratio was then experimented with during age classifier training. For the test subset of the dataset, the addition of synthetic data achieved a reduction in the mean absolute error from 3.499 years to 3.294 years. For the independent test dataset, a reduction in mean error from 4.012 years to 3.875 years was achieved.

Klíčová slova

podmíněné generativní neuronové sítě, StyleGAN, generování obličejů, odhadování věku, strojové učení, hluboké učení

Keywords

conditional generative adversarial networks, StyleGAN, face generation, age estimation, machine learning, deep learning

Citace

VENKRBEC, Tomáš. *Generování trénovacích dat pomocí GAN pro odhad věku z fotografie.* Brno, 2022. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Michal Hradiš, Ph.D.

Generování trénovacích dat pomocí GAN pro odhad věku z fotografie

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Michala Hradiše, Ph.D. Další informace mi poskytl Ing. Marián Beszédeš, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Tomáš Venkrbec
18. května 2022

Poděkování

Chtěl bych poděkovat vedoucímu mojí diplomové práce, panu Ing. Michalu Hradišovi, Ph.D., za mnoho cenných rad a připomínek. Za poskytnutí všech potřebných dat děkuji firmě Innovatrics a především panu Ing. Mariánu Beszédešovi, Ph.D. za velkou ochotu a pomoc při řešení mojí práce. Také si nesmírně vážím obrovské pomoci mojí rodiny, která mě ve všem od začátku do konce podporovala. Nemalé díky také patří všem mým přátelům, kteří tu pro mě byli vždy, když jsem potřeboval posílit motivaci k práci.

Obsah

Seznam obrázků

Kapitola 1

Úvod

Už od počátků vědeckého zkoumání metod strojového učení se objevovaly snahy napodobit strukturu a funkce lidského mozku, čímž vznikla podoblast zvaná „hluboké učení“. *Neuronové sítě* se brzy staly pro ni typickým přístupem.

Nevýhodou neuronových sítí v porovnání s lidským mozkem je však jejich závislost na velkém počtu přesně anotovaných dat. Získání dostatku takových dat může být ale časově i finančně náročná činnost, kterou je často nutné provádět manuálně. Odhadování věku je sice charakteristickou úlohou pro tuto oblast, ale přesto se existující datové sady potýkají s mnohými problémy. Některé věkové kategorie, především děti a senioři, jsou v nich málo zastoupené. Dalším limitujícím faktorem je nevyváženost pohlaví či etnik, která také může negativně ovlivňovat dosaženou přesnost sítě. Vzhledem ke způsobům, kterým jsou takové datové sady anotovány, přichází také problém s nejednoznačností anotací, které mohou být velmi nepřesné, kvůli rozdílu mezi skutečným a zjevným věkem osob.

Generativní neuronové sítě (anglicky *generative adversarial networks*, GAN) [11] můžou sloužit ke zmírnění těchto problémů. Hlavním přínosem této metody je přidání druhé neuronové sítě, *generátoru*. Z anglického názvu metody vyplývá, že sítě spolu navzájem soupeří. To spočívá v tom, že hlavním cílem generátoru je vytvářet unikátní vzorky, kterými bude schopný zmást druhou síť tak, aby generované vzorky považovala za skutečné. Tímto způsobem se sítě svojí soutěživostí vzájemně zdokonalují tak dlouho, než je generátor schopný generovat data nerozeznatelná od dat z distribuce, která byla použita k trénování.

V této diplomové práci navazuji na mojí bakalářskou práci, kde jsem porovnával různé typy generativních neuronových sítí, které jsem rozšířil o možnost podmíněného generování na základě zvolených vlastností, úspěšně je natrénoval a využil k syntéze lidských tváří. Získané poznatky zde využiji k rozšíření dnešních nejmodernějších přístupů GANů [24] o podmíněné generování. Pomocí natrénovaného generátoru bude možné získávat fotorealistické lidské tváře, ke kterým budou díky podmíněnému generování k dispozici informace o vlastnostech, jakými jsou například věk a pohlaví, a bude jimi možné doplnit skutečná data. Dalším zkoumaným způsobem získávání syntetických dat je projekce skutečného obrázku do latentního prostoru generátoru pomocí natrénovaného enkodéru [27].

Hlavním problémem, který si poté tato práce klade za cíl řešit, je to, zda lze pomocí takových syntetických dat zpřesnit klasifikace při odhadování věku osob, zejména v případě řídce zastoupených věkových kategorií. Dalším cílem práce je proto implementovat a natrénovat konvoluční neuronovou síť založenou na nejúspěšnějších metodách odhadování věku [9], pomocí které budu vliv přidaných dat vyhodnocovat. K účelům trénování obou typů sítí bude vytvořena nová datová sada, vhodně pokrývající všechny věkové kategorie a různé etnické skupiny.

Kapitola 2

Konvoluční neuronové sítě pro odhad věku

Vznik neuronových sítí byl kulminací vědeckých snah o vytvoření zjednodušeného matematického modelu způsobu, kterým mozek zpracovává získané informace. Základními částmi tohoto modelu jsou umělé neurony, výpočetní jednotky, které jsou abstrakcí skutečných nervových buněk. Tyto neurony jsou poté uspořádány do vrstev, které jsou navzájem propojené. Takový přístup je vhodný pro práci s daty ve vektorové formě, ale při práci s vícedimenzionálními daty, jakými jsou například obrázky, je kvůli převodu do vektorové podoby ztracena prostorová závislost dat. Z tohoto důvodu byly vytvořeny *konvoluční neuronové sítě* (anglicky *convolutional neural networks*, CNN), které tento problém řeší nahrazením neuronů konvolučními filtry.

Z mnoha druhů úloh, ve kterých konvoluční neuronové sítě dosáhly state-of-the-art výsledků jsou pravděpodobně nejtypičtějšími detekce a klasifikace objektů. Vznik datových sad jako *ImageNet*[1] a soutěžím na nich založených, například ILSVRC[2], byl hnacím motorem rozvoje architektur sítí. V roce 2012 soutěž ovládla architektura *AlexNet* [26], se kterou autoři docílili přesnosti 63 % při klasifikaci konkrétní třídy z 1000 možných tříd v podmnožině ImageNetu. Když v roce 2017 proběhl poslední ročník soutěže, architektury jako *ResNet* [13] dosahovaly přesnosti přes 80 %. Dnešní state-of-the-art modely již rutinně dosahují přesností přes 90 %.

Ačkoliv odhadování věku osob na základě fotky tváře je přirozeně spíše regresní úloha, přesnějších výsledků bylo dosaženo její transformací na vícetřídní klasifikační úlohu. Tato myšlenka byla popularizována metodou DEX [33]. Použití klasifikace místo regrese vedlo k mitigaci problému, kdy odlehlá data příliš zvyšovala hodnotu chyby, což zapříčinilo nestability v trénování. Ovšem kvůli nedostatku dat v méně zastoupených věkových kategoriích, nevyváženým třídám a nejednoznačným požadovaným výstupům v datových sadách směřovala vícetřídní klasifikace častěji k *přeučení* (anglicky *overfitting*) [9]. S nápadem rozdělit vícetřídní klasifikaci na mnoho binárních klasifikací přišli autoři metody *Ranking-CNN* [5], kde pro každou možnou třídu reprezentující konkrétní věk trénujeme jednoduchou konvoluční síť, jejímž úkolem je klasifikovat, zda je osoba na obrázku mladší či starší, než je hodnota věku odpovídající této sítí. Další přístup využívá metoda *Deep Label Distribution Learning* [10] (zkráceně DLDL), která s výstupem softmax vrstvy pracuje jako s rozdělením pravděpodobnosti a snaží se snižováním Kullback-Leibler divergence přiblížit skutečnému

[1] https://www.image-net.org/

[2] ImageNet Large Scale Visual Recognition Challenge

rozdělení pravděpodobnosti požadovaných výstupů. Tyto dva ve své době state-of-the-art přístupy zmírnily problémy, které přinášela vícetřídní klasifikace, a jejich kombinací bylo metodou *DLDL-v2* dosaženo opět nejlepších výsledků v celé řadě různých datových sad [9].

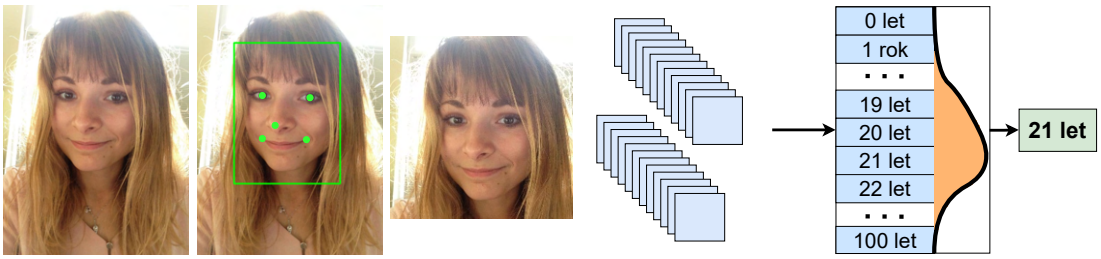
2.1 Odhadování věku

Pokud tuto úlohu pojmeme jako vícetřídní klasifikaci, může se v porovnání s klasifikací objektů nebo osob, kde můžeme mít tisíce různých tříd v jedné datové sadě, zdát jako příliš jednoduchá, jelikož je zde používán menší počet tříd, typicky $y \in 0, 1, \ldots, 100$, kde y je index třídy odpovídající věku osoby. Ve skutečnosti ale nejde o jednoduchý problém, jelikož je těžké získat takové datové sady, ve kterých budeme mít kompletně anotovaná data a vyvážené jednotlivé třídy. Dalším problémem může být to, jak moc se distribuce jednotlivých datových sad od sebe liší, kde k vytvoření dobré datové sady je nutné dbát na vyvážení věku, pohlaví, ras, atd. Kromě toho můžou být vzorky získané za různých podmínek a napříč vzorky se může lišit například osvětlení, stíny, kvalita fotografie, orientace obličeje fotografovaného subjektu či použití různých módních doplňků.

I po vytvoření datové sady, která všechny tyto faktory bere v potaz a je vyvážená [19] ale přichází problém, že skutečný (biologický) věk osoby se může výrazně lišit od zjevného věku stejné osoby na jednotlivých vzorcích [33].

Lidský věk je důležitý demografický údaj, a jeho automatické odhadování může být využitelné ve spojení s dalšími měkkými biometrickými údaji v různých odvětvích, například v oblasti zabezpečení či vytváření personalizovaného obsahu. V této práci se zabývám výhradně odhadováním věku na základě snímků obličeje, ale s využitím strojového učení můžeme věk odhadovat i s použitím jiných charakteristik, například z řeči [44], v oblasti medicíny ze snímků mozku z magnetické resonance [6] či rentgenových snímků ruky [46].

Jednotlivé kroky procesu odhadování věku pomocí konvolučních sítí jsou naznačeny na obrázku 2.1. V této kapitole se postupně budu věnovat každému z těchto kroků.



(a) Vstup (b) Detekce (c) Zarovnání (d) Klasifikace (e) Odhad věku

Obrázek 2.1: Postup odhadování věku z fotografie.

2.1.1 Zpracování obrazových dat

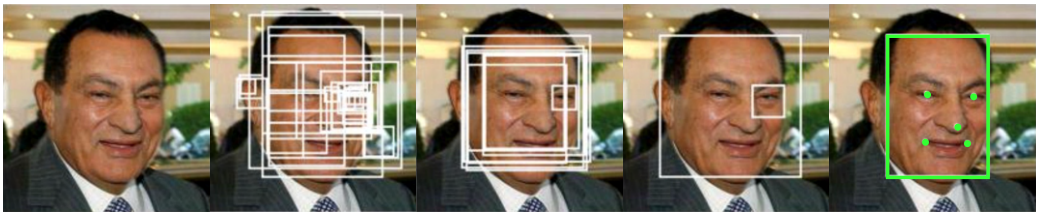
Aby byl model konvoluční neuronové sítě schopen odhadovat věk, je vhodné vstupní data nejdříve předzpracovávat před jejich vložením na vstup. Vzhledem k tomu, že na snímcích se může vyskytovat více obličejů, které mohou být v různých pozicích, není příliš jednoduché natrénovat model, který by s takto nekonzistentními daty dokázal pracovat. Pro přesnost klasifikace je důležité, aby stejný proces byl aplikován na vstupní vzorky z trénovací datové

sady, při validaci a provozu modelu a tím bylo garantováno, že na vstupu budou vždy stejně zarovnané snímky.

Detekce obličejů. Prvním krokem zpracování dat je detekce jednotlivých obličejů na vstupním snímku. Pro každý obličej ze vstupního obrazu získáme ohraničující rámeček (anglicky *bounding box*), a s těmito oblastmi původního snímku můžeme dále pracovat.

Vzhledem k tomu, že detekce obličejů je v oblasti počítačového vidění klasickým tématem, máme na výběr z mnoha metod. Nabízí se použít řešení, která jsou již dostupná v různých knihovnách orientovaných na počítačové vidění, jako jsou *OpenCV*[3] a *dlib*[4], jejichž implementace pracují jako *kaskádový klasifikátor Haarových příznaků* a klasifikátor na základě *histogramu orientovaných gradientů*. Tyto metody jsou sice dnes již zastaralé, ale jsou dostatečně přesné a umožňují real-time detekci obličejů. Dalším vývojem v oblasti počítačového vidění byla detekce dále zefektivněna a s nárůstem výpočetního výkonu je možné využívat i metod založených na hlubokém učení.

Jednou takovou metodou je *Multi-task Cascaded Convolutional Network* [45], zkráceně MTCNN, která spojuje detekci obličejů s detekcí význačných bodů v obličeji, které jsou využívány v další části předzpracování obličeje. Jak je z názvu metody patrné, skládá se z několika konvolučních sítí, konkrétně tří, které nejdříve detekují větší množství kandidátních bounding boxů, které odpovídají místům, kde mohou být obličeje. Další síť zamítá falešné detekované bounding boxy a spojuje je k vytvoření finální detekce, která je vstupem poslední sítě, provádějící detekci význačných bodů v obličeji. Na ukázku detekce obličejů a význačných bodů se můžeme podívat na obrázku 2.2.



Obrázek 2.2: Detekce obličejů a jejich význačných bodů pomocí MTCNN (převzato z [45], upraveno).

Význačné body a zarovnání obličeje. Často se můžeme setkat s detekcemi sestávajících se z 68 význačných bodů, ale pro účely zarovnání obličeje jich stačí pouze 5. Těmi jsou souřadnice středu levého a pravého oka (vypočítány na základě ostatních souřadnic očí), špičky nosu a levého a pravého koutku úst. Tyto význačné body jsou také výstupem zmíněné metody MTCNN (viz obrázek 2.2), čímž je při zpracování vzorku ušetřena práce, v jiném případě by bylo nutné k detekci význačných bodů použít opět zmíněné knihovny pro počítačové vidění, nebo jiné přístupy využívající hluboké učení.

S použitím těchto bodů je poté prováděna transformace k zarovnání obličeje podle referenčních souřadnic význačných bodů. Jde o *částečnou 2D afinní transformaci* (neboli *podobnostní transformaci*), tím pádem provádíme pouze škálování, rotaci a posun obličeje. Kvůli tomu je zachován tvar obličeje, ale není vždy možné transformovat obličej tak, aby pozice jeho význačných bodů odpovídaly těm referenčním, proto je řešení pouze aproximo-

[3] https://opencv.org/
[4] http://dlib.net/

váno metodou nejmenších čtverců. V případě, že obličej se nachází na okraji obrázku a po jeho zarovnání nám chybí část dat, získáme zbytek pomocí replikování okraje, jak je vidno na obrázku 2.3.

Obrázek 2.3: Zarovnání obličeje na základě pětice význačných bodů.

2.2 Architektury sítí

Se zlepšováním přesnosti odhadování věku není spojen pouze vývoj nových metod k trénování konvolučních neuronových sítí, ale také vývoj jejich architektur. V průběhu let byly postupně představovány architektury, které byly rychlejší, efektivnější a větší, než předešlé, čímž dosahovaly i lepších výsledků na stejných úlohách s použitím stejných metod učení [39].

Jak již bylo v úvodu kapitoly zmíněno, největší vliv na rozvoj konvolučních sítí měla celosvětová soutěž v klasifikaci, *ImageNet Large Scale Visual Recognition Challenge*. Velmi kvalitní datová sada *ImageNet* svým rozdělením na 1000 tříd tvořila takovou výzvu, která upoutala pozornost mnohých vědců. Ačkoliv se tato soutěž netýkala odhadování věku, průlomové technologie, které byly vytvořeny v rámci jednotlivých ročníků soutěže, ovlivňovaly vývoj také v této oblasti. Nyní si představíme některé architektury, které z této soutěže vzešly, jelikož na nich staví i některé dnešní state-of-the-art modely.

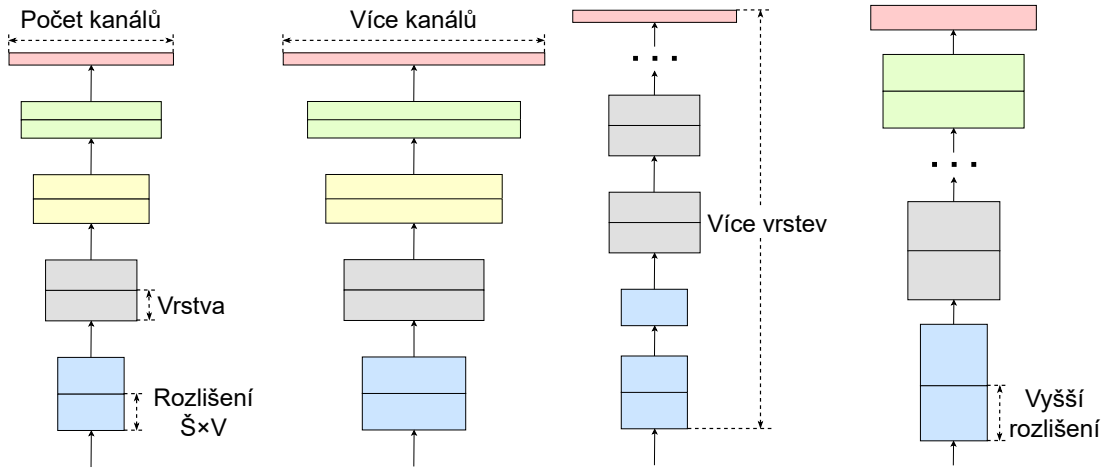
AlexNet. První architekturou, která způsobila významný pokrok v oblasti konvolučních sítí, byla *AlexNet* [26]. Ta byla vytvořena týmem vědců z univerzity v Torontu, který vedl Alex Krizhevsky, podle kterého je také pojmenována. Průlomové bylo použití velmi hluboké konvoluční neuronové sítě. Architektura AlexNet byla ve své době jednou z největších s více než 60 miliony trénovatelných parametrů, což samo o sobě přinášelo výzvy například v oblasti hardwaru a řešení přeučení. Autoři tyto problémy překonali přesunutím výpočtů na více grafických karet a použitím nových regularizačních technik jako *dropout* [38]. Natrénovaný model využívající architekturu AlexNet v soutěži ILSVRC 2012 obsadil první místo a s *top-5* chybou 15,3 % v porovnání s druhým místem, které dosáhlo chyby 26,2 % definitivně určil směr, kterým se novější modely budou ubírat.

VGG. Ačkoliv dosáhla „pouze“ druhého místa v ILSVRC 2014, jedná se o velmi kvalitní architekturu, kterou si volili autoři state-of-the-art metod k odhadování věku, kterým se tato práce věnuje [9, 10, 33]. Zároveň jsou dodnes často používány aktivace vrstev předtrénovaných modelů této sítě k výpočtu chyby, takzvané *perception loss*. Pomocí VGG bylo demonstrováno, že škálování sítí do šířky (počtem filtrů v konvolučních vrstvách) a do

délky (počtem konvolučních vrstev), umožněné zmenšením velikosti konvolučních jader na velikost 3×3, má prospěšný efekt na dosažitelnou přesnost sítě, kde se autorům podařilo docílit *top-5* chyby 7,3 % [37].

ResNet. Skutečnost, že trénování hlubších modelů konvolučních neuronových sítí přináší lepší výsledky, byla inspirací pro vznik nových architektur. Architektury jako VGG [37] se potýkaly s problémem, kdy při rostoucí hloubce sítě klesaly gradienty a tudíž se takové sítě špatně trénovaly a rovněž to tvořilo limit, jak hluboké takové sítě mohou být. Autoři architektury ResNet [13] tento problém vyřešili přidáním „zkratek“, neboli spojeními, které zachovávají gradient přeskakováním jedné nebo více konvolučních vrstev, zvaných *reziduální bloky*. Tento pokrok umožnil trénování sítí, které mohou mít stovky trénovatelných vrstev bez problému mizejících gradientů a s modelem využívajícím tuto architekturu autoři dosáhli na první místo v ILSVRC 2015 s 3,57% *top-5* chybou.

EfficientNet. Trend škálování sítí, který lze pozorovat na modelech vytvářených pro dosažení state-of-the-art přesnosti na datové sadě ImageNet, neustal ani poté, co se soutěž ILSVRC přestala pořádat. V dnešní době se můžeme setkat i s modely, které jsou tak obrovské, že jejich trénování vyžaduje specializované knihovny provádějící paralelizaci a rozdělující jednotlivé části sítě mezi více akcelerátorů [15]. Právě díky tomu, že zde narážíme na limity moderního hardwaru, byla vytvořena architektura *EfficientNet* [39]. Autoři systematicky analyzovali všechny způsoby škálování sítí (viz obrázek 2.4) a přišli se způsobem, jak efektivně škálovat sítě napříč všemi dimenzemi (hloubka, šířka, rozlišení).



(a) Baseline (b) Do šířky (c) Do hloubky (d) Zvyšování rozlišení

Obrázek 2.4: Způsoby škálování konvolučních neuronových sítí.

Zvyšování hloubky sítě přidáváním konvolučních vrstev je nejběžnějším způsobem zvětšování sítě. Intuice je taková, že hlubší síť je schopná se naučit složitější rysy a měla by lépe generalizovat [39]. Ve skutečnosti je ale takové modely těžší trénovat kvůli mizejícím gradientům [37] a i přes to, že byly představeny řešení tohoto problému, jako architektura ResNet [13], výhoda hlubší sítě rychle klesá, což na příkladu sítí ResNet dokazuje to, že zvýšení počtu trénovatelných vrstev ze 101 na 1000 již nepřináší vyšší přesnost.

Při rozšiřování sítí do šířky zvyšováním počtu konvolučních filtrů je cílem zachytit více jemnějších rysů. Ve skutečnosti toto ale poměrně rychle vede k potížím se zachycením

vysokoúrovňových rysů a přesnost získaná takovým škálováním opět velmi rychle klesá, na rozdíl od rostoucích nároků na výpočetní výkon [39].

Posledním zkoumaným typem škálování bylo zvyšování rozlišení jednotlivých vrstev, kde autoři článku také zjistili, že také přesnost získaná rozšiřováním této dimenze sítě nestoupá přímo úměrně s výkonem, co vede k jednoznačnému závěru, že není příliš výhodné škálovat jednotlivě jednotlivé dimenze konvolučních neuronových sítí. Pro vzorky s vyšším rozlišením je rozumné zvýšit také hloubku sítě, aby bylo možné konvolučními filtry lépe zachytit rysy, které tvoří více pixelů vstupu. Obdobně lze argumentovat, že pro vyšší rozlišení je třeba zvýšit šířku sítě zvýšením počtu filtrů, aby bylo možné zachytit více různých rysů. Je tedy intuitivní škálovat všechny dimenze sítě naráz [39].

Výsledkem práce bylo vyvinutí metody k rovnoměrnému škálování sítí, na základě dostupného výpočetního výkonu pro větší síť. Jako demonstraci svých poznatků vytvořili autoři řadu různě velkých modelů, jako přímou konkurenci již implementovaných modelů. Největší vytvořený model, *EfficientNetB7*, byl porovnán se state-of-the-art řešením té doby, sítí *GPipe* [15], a dosáhl stejné přesnosti na datové sadě ImageNet, ovšem s 8,4× menším počtem trénovatelných parametrů a 6,1× větší rychlostí inference. I v dnešní době sítě škálované na základě tohoto paradigmatu dosahují state-of-the-art výsledků, a v této oblasti jim tvoří konkurenci pouze sítě typu *Transformer* pro zpracování obrazu.

2.2.1 Základní části konvolučních sítí

Společně se strukturou organizace konvolučních vrstev v jednotlivých architekturách, k zefektivnění trénování přispíval i výzkum týkající se přidružených vrstev v sítích.

Aktivační funkce Softmax. Ve výstupních vrstvách klasifikačních modelů se typicky setkáváme s funkcí *softmax*, jejíž výhodný obor hodnot v rozsahu $(0; 1)$ se hodí tam, kde je výstupem pravděpodobnost jevů. Jde o zobecněnou verzi logistické funkce (sigmoidy), definované následovně:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.1}$$

Výhodou softmaxu oproti sigmoidní funkci je tedy její využití nejen na binární klasifikaci, ale obecně ke klasifikaci n tříd (viz rovnice 2.2), kde normalizací je zajištěno, že součet výsledného vektoru bude roven 1, a můžeme tedy s výstupem pracovat jako s pravděpodobnostním rozložením, čehož je v případě metod k odhadování věku využíváno [9, 10].

$$\sigma(\vec{x}) = \frac{e^{x_i}}{\sum_{j=0}^{n-1} e^{x_j}}, \text{ kde } i \in 0, \ldots, n-1 \tag{2.2}$$

Aktivační funkce ReLU. Od příchodu hlubokých konvolučních sítí se jejich nedílnou součástí stala právě tato nesaturující nelineární funkce, která v porovnání s dříve používanými saturujícími aktivačními funkcemi (jejich gradienty byly tím pádem blízké nule, například funkce *tanh*) dosáhla stejné hodnoty chyby několikanásobně rychleji [26]. Je také možné jí rychleji vyčíslit. Ovšem kvůli tomu, že pro záporné vstupní hodnoty neprobíhá aktivace, se může objevit *problém mizejících gradientů*. Z tohoto důvodu se častěji setkáváme s funkcí *LeakyReLU*, která „propustí“ malou část α záporného vstupu na výstup sítě (typicky například $\alpha = 0{,}01$), viz rovnice 2.3.

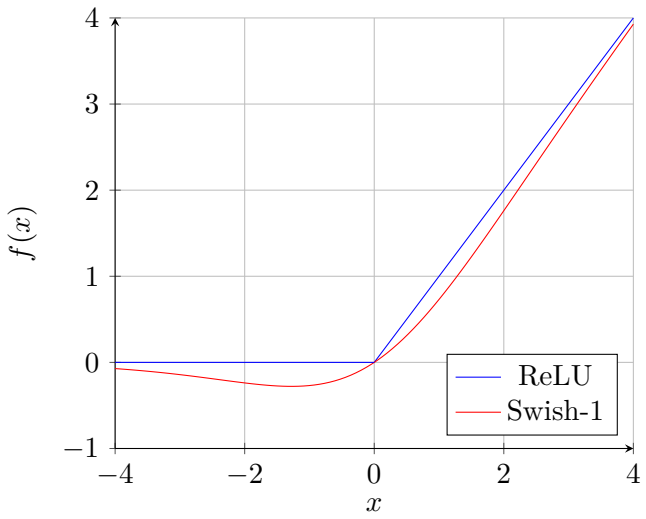
$$f(x) = \begin{cases} \max(0, x) & \text{pro } x \geq 0 \\ \alpha \cdot x & \text{jinak} \end{cases} \tag{2.3}$$

Tímto se sice ztrácí část výhod rychlosti výpočtu ReLU, ale to je v dnešní době zanedbatelné, proto se s touto přenosovou funkcí můžeme setkat v mnoha nejmodernějších typech konvolučních sítí, kterými se tato práce zabývá [5, 9, 10, 21, 22, 27].

Aktivační funkce Swish. Kombinací výše zmíněné aktivační funkce ReLU a logistické sigmoidy vznikla nová aktivační funkce *Swish* [31] (viz rovnice 2.4), případně *Swish-1* nebo *Sigmoid Linear Unit* (zkráceně SiLU) [7] v případě nastavení trénovatelného parametru β na konstantní hodnotu 1.

$$f(x) = x \cdot \sigma(\beta \cdot x) \tag{2.4}$$

Pouhým nahrazením aktivační funkce ReLU touto funkcí bylo dosaženo lepších výsledků ve většině úloh a proto je používána například v sítích EfficientNet. Rozdíly oproti ReLU jsou takové, že derivace funkce jsou spojité, což dovoluje trénování i se zápornými čísly (podobně jako LeakyReLU) a to, že se nejedná o monotónní funkci [31]. Průběh funkcí ReLU a SiLU lze vidět na obrázku 2.5.



Obrázek 2.5: Porovnání aktivačních funkcí ReLU a SiLU.

Regularizace a normalizace. U větších sítí nebo sítí, které se učí na malé datové sadě, se můžeme setkávat se *přeučením* (anglicky *overfitting*). Moderní architektury se s tímto vypořádávají použitím dropout vrstev [10, 39], kterou je v průběhu trénování náhodně deaktivována část vstupů.

Druhou technikou, kterou používají stejné architektury a také významně přispívá regularizaci, ale zároveň výrazně zrychluje a stabilizuje trénování, je normalizace. Je pomocí ní zabraňováno *vnitřnímu kovariančnímu posunu*, což znamená, že hodnoty různých vstupů mají různé distribuce. Existuje více typů normalizace, na základě toho, jaké dimenze vstupu jsou normalizovány, ale obecně se nejvíce rozšířila *dávková normalizace*, která normalizuje kanály napříč celou dávkou tak, aby jejich průměr byl 0 a směrodatná odchylka 1. Aby se předešlo tomu, že se změní to, co daná vrstva reprezentuje, jsou přidány parametry γ a β, značící faktor škálování a posunu, který je během trénování aplikován. Kromě toho je počítán klouzavý průměr parametrů normálního rozdělení reprezentující trénovací data, aby

mohla být normalizace aplikována i v průběhu inferencí po skončení trénování, kdy data nemusí přicházet v dávkách [16].

2.3 Metody trénování klasifikátorů věku

Trénování jakýchkoliv neuronových sítí je v principu hledání řešení vysokodimenzionálních funkcí, které mají v extrémním případě až miliardy parametrů. Tento n-dimenzionální prostor je *nekonvexního* charakteru, proto hledání globálního minima není triviální. V praxi je vhodným řešením i dostatečně vhodné lokální minimum, jelikož globální minimum se může přidáním dalších trénovacích dat změnit. Optimalizace se může snadno zastavit i v neoptimálních lokálních minimech, nebo v *sedlových bodech*, v jejichž okolí jsou nízké gradienty.

Adaptive moment estimation. Pro řešení tohoto optimalizačního problému se používají algoritmy založené na gradientním sestupu, kde nejznámější z nich je algoritmus *Adaptive moment estimation*, více známý jako *Adam*. Tento algoritmus kombinuje výhody dřívějších metod *RMSProp* a *AdaGrad* a je pomocí něj trénována většina dnešních modelů [24, 33]. Při použití optimalizátoru Adam není *learning rate* (česky *míra učení*) pro všechny parametry stejná, ale je ovlivněna také předchozími hodnotami gradientů. Nová hodnota parametru v trénovacím kroku t kombinuje aktuální gradient g_t s exponenciálně klesajícím průměrem gradientů (rovnice 2.5), jinak zvaným *hybnost* (anglicky *momentum*), a exponenciálně klesajícím poměrem druhých mocnin gradientů (rovnice 2.6), používaným ve dříve zmíněných algoritmech [25]. Rychlost exponenciálního poklesu je dána velikostí hodnoty hyperparametrů β_1 a β_2.

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t \tag{2.5}$$

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2 \tag{2.6}$$

Nová hodnota trénovatelného parametru θ je poté počítána na základě následující rovnice:

$$\theta_t = \theta_{t-1} - \alpha \cdot \frac{m_t}{\sqrt{v_t} + \hat{\epsilon}} \cdot \frac{\sqrt{1 - \beta_2^t}}{1 - \beta_1^t}, \tag{2.7}$$

kde α je mírou učení a $\hat{\epsilon}$ je zvolená malá hodnota sloužící k zabránění dělení nulou v případě nízkých gradientů.

Křížová entropie. Chybovou funkcí, která je v případě některých metod odhadování věku představených v této práci optimalizována, je křížová entropie. Její použití je logické vzhledem k tomu, že výstupy softmax vrstev trénovaných modelů jsou pravděpodobnostní distribuce, kde v ideálním případě chceme, aby tyto předpovězené distribuce měly co nejmenší míru rozdílu od skutečných distribucí. Křížová entropie mezi libovolným vektorem požadovaných výstupů y a výstupem softmax vrstvy modelu $\hat{y}$ pro n klasifikovaných tříd se spočítá následovně:

$$L\left(\vec{y}, \hat{y}\right) = -\sum_{i=0}^{n-1} y_i \cdot \log(\hat{y}_i). \tag{2.8}$$

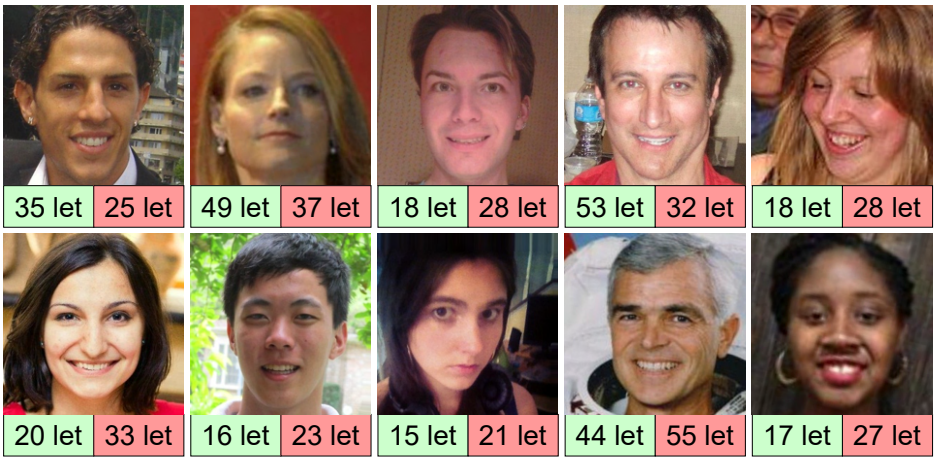
Speciálními případy křížové entropie jsou *binární křížová entropie*, kde jsou klasifikovány pouze 2 třídy (užíváno metodou Ranking-CNN [5]) a *kategorická křížová entropie*, kde je očekávaný vektor zakódovaný pomocí *one-hot encoding*, znamenající, že právě jedna třída bude mít pravděpodobnost 1 a ostatní 0, co v případě klasifikování věku platí [33] a lze si tímto ušetřit počítání oproti obecné křížové entropii.

2.3.1 Deep EXpectation (DEX)

Cílem metody DEX [33] nebylo přímo odhadovat skutečný věk osob, nýbrž jejich zdánlivý věk. Jedná se o úlohu, která byla v té době v počátcích a teprve docházelo k vytváření datových sad, které budou obsahovat i takové informace. Jedna taková datová sada byla vytvořena pro účely soutěže *ChaLearn Looking at People 2015* [8] sestávající se z 4699 fotografií osob, kde jednotlivé odhady zdánlivého věku byly získány crowdsourcingem. Ukázka osob s porovnáním jejich skutečného a odhadovaného věku, zvýrazňující problém, který jejich vzájemný rozdíl může přinášet, je k vidění na obrázku 2.6. Metoda DEX v této soutěži skončila na prvním místě, se *střední absolutní chybou* 3,1 roku, se kterou autoři metody výrazně překonali i hranici přesnosti lidských odhadů, kterou označili autoři soutěže [33]. Nutno podotknout, že tohoto výsledku dosáhli použitím 20 natrénovaných modelů najednou, ale i použitím jedné neuronové sítě se podařilo dosáhnout výsledné střední chyby 3,22 roku.

Jak již bylo v úvodu kapitoly zmíněno, úspěch této metody tkvěl v tom, že se zde autoři rozhodli úlohu odhadování věku přeformulovat z regresního na klasifikační problém. Kvůli omezené velikosti soutěžní datové sady a obličejových datových sad obecně byli autoři nuceni vytvořit vlastní datovou sadu IMDB-Wiki[5], obsahující přes půl milionu snímků společně s anotacemi skutečného věku. Navíc autoři využili ke klasifikaci síť, která nejdříve byla trénovaná na klasifikaci objektů datové sady ImageNet, kterou poté trénují na vlastní datové sadě a dotrénují na cílové datové sadě, technikou zvanou *transfer learning*.

Využitá předtrénovaná síť používala architekturu VGG-16 [37] a byla k ní pro klasifikaci přidána softmax vrstva se 101 neurony odpovídajícím jednotlivým věkům v rozsahu 0–100, která byla dotrénována (anglicky *fine-tuning*). K provedení odhadu věku zde namísto použití neuronu s nejvyšší pravděpodobností autoři prováděli vážený součet hodnot jednotlivých výstupních neuronů. K dosažení ještě větší přesnosti bylo navrženo použití více natrénovaných modelů konvolučních sítí naráz, kde každý byl trénovaný na trochu jiné podmnožině cílové datové sady a finální odhad věku je průměrem odhadů věků těchto jednotlivých sítí.



Obrázek 2.6: Porovnání skutečného (zelený) a zdánlivého (červený) věku podle ~40 odhadů.

[5] https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/

2.3.2 Deep Label Distribution Learning (DLDL)

Vlastností datových sad k odhadování věku, které předchozí metoda nevyužívala naplno, bylo to, že existuje vztah mezi sousedními třídami plynoucí z toho, že získané anotace nemusí být přesné a může být určitá nejistota spojená s jejich ground-truth hodnotou. Laicky řečeno, tvář osoby, které je 30 let, je pravděpodobně podobnější tvářím osoby o rok starší, než tvářím 80letých osob a tato informace by mohla být efektivně využita během trénování klasifikátorů. Proto se ukázalo jako užitečné nahradit klasifikaci pomocí křížové entropie vstupních dat a výstupu softmax vrstvy jiným způsobem [10].

V této metodě se pracuje s celou pravděpodobnostní distribucí, která je výstupem softmax vrstvy. Poté je jako chybová funkce minimalizována Kullback-Leibler (zkráceně KL) divergence mezi modelem předpovězeným rozdělením pravděpodobnosti a rozdělením pravděpodobnosti anotací trénovacích dat. U většiny trénovacích dat ovšem disponujeme pouze jedinou hodnotu odpovídající věku, proto si je nutné pravděpodobnostní rozdělení určit ručně. Vycházíme z předpokladu, že podobně staré osoby jsou si podobné a jako jedno možné řešení bylo vybráno použití normálního rozdělení, se střední hodnotou μ odpovídající ground-truth věku osoby. Výběr správné směrodatné odchylky σ je složitější, pokud nám není poskytnuta, ale autoři ověřili, že zvolení rozumné hodnoty ručně (např. $\sigma = 2$) funguje dostatečně dobře [10].

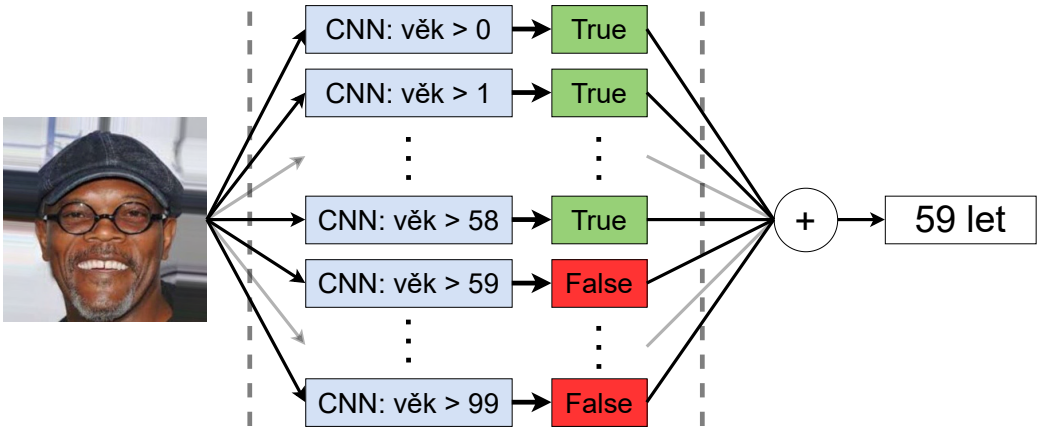
Pomocí této trénovací metody na síti typu VGG se podařilo docílit state-of-the-art výsledků napříč různými datovými sadami, v případě *ChaLearn 2015* byla dosažená hodnota střední absolutní chyby 3,51 roku. Může se tedy zdát, že tato metoda nepřinesla lepší výsledky než předchozí představená metoda, proto je potřeba zmínit, že tohoto výsledku bylo dosaženo bez použití externích datových sad při trénování, šlo pouze o fine-tuning sítě, jenž byla trénována ke klasifikaci na jiné datové sadě. Do té doby nejlepší síť nevyužívající externí data dosáhla střední absolutní chyby 5,7 roku.

2.3.3 Ranking-CNN

Na to, jak efektivně využívat anotace věku, se zaměřili i autoři metody *Ranking-CNN*. Způsob, kterým řeší nejasnost očekávaných výstupů, se liší od metody DLDL. Metodou Ranking-CNN ohodnocujeme každý vstup pomocí více jednoduchých konvolučních sítí, jednou pro každou klasifikovanou třídu, kde každá provádí binární klasifikaci (viz obrázek 2.7). V případě odhadování věku si tuto binární klasifikaci lze vyložit jako rozhodnutí, zda je osoba mladší nebo starší než věk, který konkrétní síti odpovídá. Výsledný odhad věku je reprezentován počtem dílčích sítí, které vyhodnotily věk jako pravděpodobněji starší, než je jejich hranice [5].

Velkou výhodou této metody je to, že každá síť se efektivně učí pomocí všech dostupných trénovacích dat, takže je snížena závislost na velkém počtu trénovacích dat. V případě klasifikování věku v rozmezí 0–100 let je nutné natrénovat 100 samostatných sítí. To ale netrvá příliš dlouhou dobu, vzhledem k tomu, že jde o jednoduché konvoluční sítě, které vycházejí z předtrénované základní sítě na datových sadách s obličeji, jejíž váhy potom použijeme pro fine-tuning těchto jednotlivých modelů provádějící binární klasifikaci.

Evaluace metody na datových sadách ChaLearn Looking at People nebyla provedena, ale byla vyhodnocena na datové sadě *Morph* [32], která se také často používá pro porovnání různých metod pro odhadování věku. Zde dosáhla metoda střední absolutní chyby 2,96 roku, překonávající metodu DEX dosahující chyby 3,25 roku, ale zaostávající za metodou DLDL, představenou v podobnou dobu, která dosáhla chyby 2,42 roku.



Obrázek 2.7: Schéma fungování metody Ranking-CNN.

2.3.4 DLDL-v2

Dosud zmíněné metody dosáhly v době svého představení nejlepších výsledků na různých datových sadách, přičemž využívaly prakticky stejnou architekturu sítě a lišily se ve výpočtu chyby. Právě zlepšení použité architektury sítě bylo jednou z hlavních motivací autorů metody *DLDL-v2*. Kromě toho autoři dokázali, že metoda Ranking-CNN se implicitně učí distribuci vstupních anotací a tím funguje na podobné bázi jako DLDL. Další problém, na který bylo zaměřeno, byl nesoulad cílů trénování s ohodnocovacími metrikami v případě metody DLDL, kterému se tato nová metoda vyhýbá pomocí regresního modulu, vypočítávající výsledný odhad podobně jako v případě metody DEX. Ten v průběhu trénování minimalizuje *L1 loss* mezi odhadnutým a skutečným věkem osoby. Druhou chybovou funkcí je zde KL divergence mezi rozděleními pravděpodobností softmax vrstvy a očekávaného výstupu, stejně jako v metodě DLDL [9].

Co se týče vylepšení architektury sítí, bylo upuštěno od velkých VGG sítí ve prospěch menšího plně konvolučního modelu s menšími počty konvolučních filtrů. Celkový počet trénovatelných parametrů sítě je zde 36× menší, a vede k 3× rychlejší inferenci. Napříč tomu se ve spojení s optimalizovaným spojením předchozích metod jedná o velmi efektivní metodu, která i v dnešní době dosahuje state-of-the-art výsledků napříč žebříčky. Bez jakýchkoliv externích trénovacích dat je střední absolutní chyba na datové sadě *ChaLearn 2015* 3,14 roku a na datové sadě *Morph* pouhých 1,97 roku [9].

Kapitola 3

Generativní neuronové sítě pro generování obličejů

Koncept generativních neuronových sítí (anglicky *generative adversarial networks*) je v oblasti strojového učení stále poměrně nový, neboť se do povědomí začal dostávat teprve v roce 2014, kdy jej představil americký vědec Ian Goodfellow, ještě v době svého doktorského studia na Montrealské univerzitě [11]. Netrvalo dlouho, než o tuto revoluční technologii začalo projevovat zájem mnoho výzkumníků z celého světa a ukázal se její skutečný potenciál. Tím, že jde o generativní model, jsou jedním možným výstupem syntetická data ze stejné distribuce, jako byla trénovací množina. Užití takových dat může asistovat při trénování jiných modelů, neboť je ubrán lidský faktor z procesu trénování, jelikož trénování s učitelem (anglicky *supervised training*) je závislé velkém počtu typicky označkovaných dat.

Svůj podíl na vzniku generativních neuronových sítí se od počátku snaží obhájit další přední výzkumník v této oblasti, Jürgen Schmidhuber. „Otec hlubokého učení“, jak je mu přezdíváno, přišel s podobnou myšlenkou již v roce 1990, kdy jí ve svém článku nazval „artificial curiosity“ (česky *umělá zvědavost*) [35]. Teprve mnohé pokroky ve výpočetní technice a dalších relevantních odvětvích umělé inteligence vedly k tomu, že se tato tehdy nadčasová technologie mohla konečně realizovat. Další průkopník umělé inteligence, Yann LeCun, ji dokonce označil jako „nejzajímavější nápad ve strojovém učení za posledních 10 let“ [3]. V rámci této práce se věnuji jejímu uplatnění v rámci syntetizování fotorealistických lidských tváří, což je jedna z klasických úloh sloužících jako spolehlivé měřítko rozvoje této technologie.

Princip generativních neuronových sítí je zřetelnější, použijeme-li doslovný překlad anglického *generative adversarial networks*, čili *generativní soupeřící sítě*. V základu existuje dvojice sítí, *generátor* a *diskriminátor*, které se liší svými cíli. Úlohou generátoru je tvořit vzorky nerozeznatelné od skutečných, jinak řečeno pocházející ze stejné distribuce. Diskriminátor poté klasifikuje, zda jsou na vstupu vzorky skutečné, nebo falešné. Obě sítě poté tvoří trénovací cyklus, kde výstup generátoru je vstupem diskriminátoru společně s reálnými daty, a na základě klasifikací je trénován generátor, aby svými vzorky byl schopen lépe „oklamat“ diskriminátor. Toto základní paradigma se v průběhu vývoje nových architektur měnilo, a v některých případech je diskriminátor přesněji spíše kritikem, jelikož jeho rolí je pouze hodnotit kvalitu vzorků [12, 24].

3.1 Vývoj generativních neuronových sítí

Schopnost generovat fotorealistické snímky, které jsou pro generativní neuronové sítě symbolické, s nimi nebyla však spojená od jejího počátku. V době svého představení byla technologie pouhým konceptem, schopným generovat pouze rozmazané snímky v nízkém rozlišení, s mnohými problémy v procesu trénování. Mimoto, koncept byl ověřen pouze na vícevrstvých perceptronech, které nejsou ideálním řešením pro zpracování obrazu.

3.1.1 Hluboké konvoluční GAN

Krokem, který významně přispěl ke kvalitě generovaných snímku (viz obrázek 3.1), byl přechod od plně propojených sítí ke konvolučním sítím, vytvořením nové třídy sítí – *hlubokých konvolučních generativních sítí* (zkráceně DCGAN). Autoři ukázali, že síť se efektivně učí cílovou distribuci dat pomocí experimentu, při kterém procházeli latentní prostor generátoru. Modifikováním hodnoty šumu v libovolném směru docházelo k hladkým interpolacím výsledného generovaného vzorku. Dalším experimentem byla vektorová aritmetika, kdy sčítáním a odčítáním šumů generujících různé snímky docházelo k sémantickým změnám výsledného snímku, který odpovídal předpokladům po aplikování stejného výrazu na původní snímky [30]. Tato vlastnost generativních neuronových sítí je důležitá k tomu, aby bylo možné podmíněné generování, jak si ukážeme později v této kapitole.

S rostoucí kapacitou sítě přibývaly také obtíže, které přinášelo trénování, proto bylo provedeno mnoho kroků k tomu, aby trénování bylo stabilnější, mnoho z nichž je stále standardem i u moderních sítí. Kromě vypuštění většiny plně propojených vrstev se začala používat dávková normalizace, aktivační funkce ReLU a její varianty, pro regularizaci se začaly používat *dropout* vrstvy a v neposlední řadě bylo prováděno snižování rozlišení dat pomocí konvolucí s krokem, namísto sdružovacích (anglicky *pooling*) vrstev. Jako efektivní se ukázalo také trénování pomocí optimalizátoru Adam [30].

Ačkoliv tyto kroky vedly k stabilizaci trénování, stále se nejednalo o jednoduchý proces, a bez důkladného ladění hyperparametrů trénování často vedlo pouze k částečnému nebo úplnému *mode collapse*, kdy pro libovolný vstupní šum byly výstupy stejné. Složitost trénování DCGAN sítí rostla exponenciálně se zvyšujícím se rozlišení vstupních dat, v případě generování obličejů se nepodařilo dostat nad rozlišení 128×128 se 3 barevnými kanály.

(a) Původní GAN

(b) Hluboké konvoluční GAN

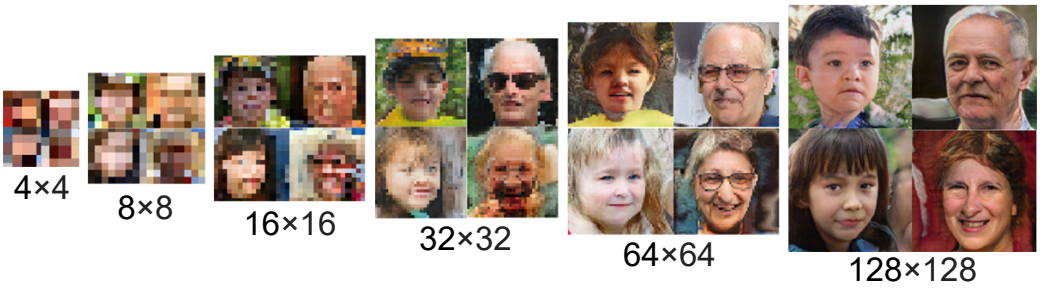
Obrázek 3.1: Tváře generované sítěmi založenými na nejstarších architekturách GAN (převzato z [11, 30], upraveno).

3.1.2 Progresivně rostoucí GAN

O pouhé dva roky později bylo umožněno stabilní trénování generativních neuronových sítí s rozlišením více než 1024×1024 pixelů. Jelikož výzkumníci společnosti NVIDIA, kteří za progresivně rostoucími generativními neuronovými sítěmi stojí, dali zdrojové kódy i natrénované váhy veřejně k dispozici, netrvalo dlouho, než se objevily demonstrace toho, jak je pomocí této metody možné generovat například fotorealistické tváře, často pouze těžce rozeznatelné od tváří skutečných osob[1].

Změna metodologie spočívala v tom, že dvojice sítí je trénována postupně, počínaje na malém rozlišení (například 4×4) a vrstvy s vyšším rozlišením jsou přidávány v průběhu trénování. Tímto způsobem se síť nejdříve učí základní strukturu dat a poté stále jemnější rysy (viz obrázek 3.2). V předchozích přístupech probíhalo trénování všech vrstev najednou [21].

Při přidávání nových konvolučních bloků vyššího rozlišení zůstávají bloky s nižším rozlišením stále trénovatelné. Navíc, adaptace nového bloku do sítě též probíhá postupně, jelikož je prováděn vážený součet jeho výstupu se škálovaným výstupem předchozího bloku před vstupem do dalších bloků sítě. Váha nového bloku je postupně zvýšena až na 1, kdy se stává plně integrovaným a před přidáním dalšího bloku ještě probíhá jeho dotrénování. Autoři prokázali, že toto nejen vede k podstatnému zvýšení stability, ale také ke zvýšené variabilitě generovaných vzorků a s dvojnásobným až šestinásobným urychlením trénování [21].



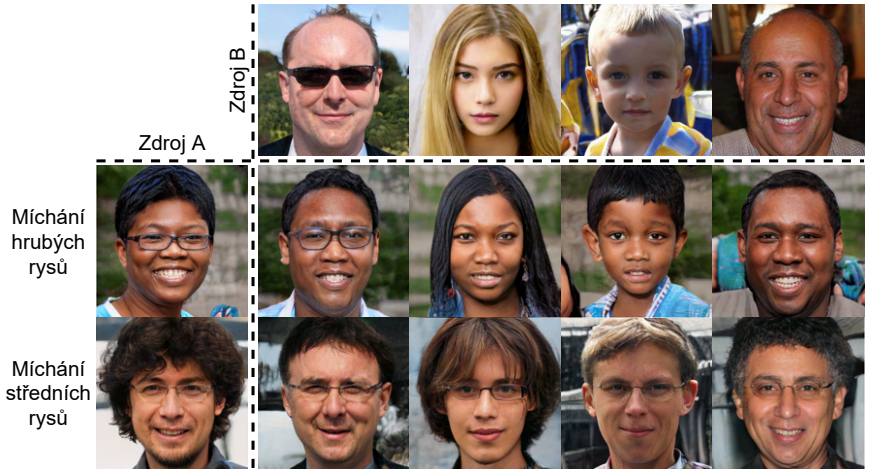
Obrázek 3.2: Progresivní zvyšování rozlišení vzorků během trénování.

3.1.3 StyleGAN

Posledních několik let platí, že StyleGAN [22, 23, 24] je de facto synonymem pro state-of-the-art v oblasti nepodmíněného generování syntetických obrázků. Jedná se o třídu architektur, které jsou vyvíjeny stejnými autory, kteří vytvořili také předchozí architekturu.

Původní StyleGAN [23] stavěl na progresivně rostoucích generativních neuronových sítích a jeho vylepšení se týkaly zejména generátoru. Způsob, kterým dřívější generátory zpracovávaly vstupní šum, byl následovný. Náhodný šum z, neboli latentní kód z latentního prostoru $\mathcal{Z}$, je přeskládán do maticové formy, aby mohl být zpracován první konvoluční vrstvou. Jakákoliv informace o požadovaném výstupu syntézy musí tedy procházet celou sítí, což z generátoru dělá efektivně černou skříňku. StyleGAN přišel s alternativním přístupem, kde je nejdříve přidána vícevrstvá plně propojená síť $f : \mathcal{Z} \to \mathcal{W}$, mapující latentní kód z na bod w v alternativním latentním prostoru $\mathcal{W}$. Tato změna vede k tomu, že výsledný latentní prostor je snadněji separovatelný, kvůli absenci závislosti na distribuci trénovacích dat. Tento latentní kód, nazývaný *styl*, je poté distribuován mezi všechny konvoluční vrstvy společně s náhodným šumem, který zvyšuje variabilitu vzorků [23].

[1] https://thispersondoesnotexist.com/

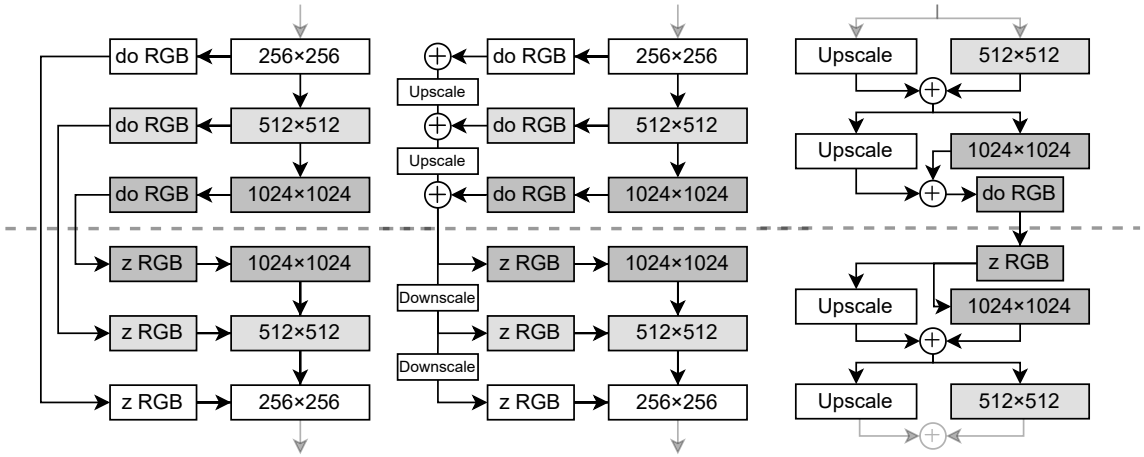


Obrázek 3.3: Míchání stylů snímků dvou různých tváří (převzato z [23], upraveno).

Dalšími výhodami použití stylů je možnost míchání různých stylů, kdy k určitému počtu konvolučních vrstev je přidáván styl w_1 a ke zbytku styl w_2, viz obrázek 3.3. Tím, ve které části sítě začneme přidávat styl w_2 můžeme ovlivnit, zda budou měněny hrubé nebo jemnější rysy vzorku, který by byl získán generováním pouze pomocí stylu w_1. Vlastnosti prostoru $\mathcal{W}$ je možné využít k tomu, aby se předešlo vzorkování z odlehlých oblastí latentního prostoru, technikou zvanou *style truncation* (česky *zkrácení stylu*). Styl zkrátíme tak, že nejdříve získáme mapováním mnoha náhodných bodů z styly $w \in \mathcal{W}$, které jsou zprůměrovány a je proveden vážený součet s původním stylem. Výsledný styl není tak odlehlý, a vede k vizuálně mnohem kvalitnějším vzorkům. Zkrácení je využíváno pouze pro demonstrační účely, při trénování využíváno není, jelikož snižuje variabilitu vzorků. Naopak míchání stylů vzorků během trénování slouží jako regularizační technika [23].

StyleGAN2 [24] vznikl jako vylepšení původní architektury po odhalení a opravení některých jejích slabých stránek, které se projevovaly ve formě artefaktů ve výsledných snímcích. Nejvýznamnější artefakt, který se týkal 99,9 % generovaných vzorků, byl spojen s užitou normalizací, která sloužila k integraci stylu do jednotlivých konvolučních vrstev. Původně tato technika v rámci jednoho konvolučního bloku fungovala tak, že namapovaný styl w byl nejdříve upraven pomocí naučené afinní transformace, a poté zastával roli parametrů adaptivní instanční normalizace (zkráceně AdaIN), kde byl normalizován výstup předchozí konvoluce s přidaným náhodným šumem [23]. Operace AdaIN byla ve StyleGAN2 nahrazena modulací a následnou demodulací vah konvoluční vrstvy pomocí stylu. Během modulace jsou váhy konvoluční vrstvy škálovány pomocí stylu a poté jsou demodulovány, aby jejich standardní odchylka byla rovna jedné [24].

Další změny se týkaly progresivního růstu sítě, jelikož bylo zjištěno, že ve stávající formě měl negativní efekt na invarianci vůči posunu. To opět vedlo k jistým artefaktům, kdy například detaily tváře jako zuby a oči měly preferenci pro určitou lokaci, a během úprav stylu vedoucím k otáčení hlavy zůstávaly stále stejně orientované. Z tohoto důvodu byly navrženy dvě varianty architektur obou sítí, které je možné použít k nahrazení progresivního růstu sítě. Prvním bylo použití *zkratek* (anglicky *skip connections*), kde výstup každé vrstvy generátoru je konvertován do RGB a je sečten s RGB výstupem předchozí vrstvy, který byl zvětšený na vyšší rozlišení. Takto je získán jediný výstup, který je vstupem diskriminátoru, kde je proces opačný, barevné obrázky jsou konvertovány zpět na vysokodimenzionální feature mapu a jsou vstupem konvoluční vrstvy společně s výstupy předchozích konvolučních



(a) Progresivní růst, MSG-GAN (b) Zkratky (c) Reziduální spojení

Obrázek 3.4: Porovnání architektur využívající progresivní růst sítí s variantami se zkratkami a reziduálními spojeními.

vrstev. Druhým způsobem bylo použití *reziduálních spojení*, podobně jako u dříve zmíněné architektury ResNet[13]. To se od zkratek liší tím, že vstupem každé konvoluční vrstvy je součet výstupu předchozí vrstvy s jejím zvětšeným (v případě generátoru) či zmenšeným (v případě diskriminátoru) vstupem. Porovnání těchto způsobů je k vidění na obrázku 3.4. Nejlepších výsledků bylo docíleno použitím generátoru využívající zkratky a diskriminátorem využívající reziduální spojení. Takto již není vynuceno samostatné trénování vrstev nižšího rozlišení, jelikož struktura sítě zůstává po celou dobu stejná. Bylo zjištěno, že použitím zkratek se generátor chová podobně jako při progresivním růstu sítě, kdy během trénování mají na výsledném snímku větší podíl postupně vyšší a vyšší rozlišení [24].

Poslední výrazné změny se týkaly latentního prostoru $\mathcal{W}$. Již dříve byla představená metrika *perceptual path length* (zkráceně PPL) [23], která pomocí aktivací vrstvy předtrénované konvoluční sítě architektury VGG měří rozdíl v aktivacích u dvojice snímků (*perceptual loss*), které byly získány jako náhodný vzorek a jeho interpolace v latentním prostoru $\mathcal{W}$ [23]. Jak už bylo zmíněno, motivací použití latentního prostoru $\mathcal{W}$ namísto $\mathcal{Z}$ byla právě jeho lepší separovatelnost, která je právě pomocí PPL odhadována. Byla pozorována korelace mezi velikostí PPL a vizuální kvalitou generovaných obrázků. Z toho důvodu byla v rámci StyleGAN2 představena *path length regularizace* [24], jejímž cílem bylo donutit síť modelovat latentní prostor $\mathcal{W}$ tak, aby náhodné kroky dané velikosti kroku vedly k nenulovým změnám pevné velikosti v generovaném snímku. Toto v praxi funguje velmi dobře v datových sadách, které jsou víc strukturované, jakými jsou třeba v rámci této práce používané datové sady lidských tváří, ale výsledky na komplexních datových sadách jako ImageNet jsou použitím této regularizace horší. V neposlední řadě, mít „jednoduchý“ latentní prostor $\mathcal{W}$ vede k tomu, že je snazší ho invertovat, čili natrénovat enkodér, který pro vstupní vzorek určí styl w, kterým generátor vygeneruje co nejpodobnější snímek [24].

Současný výzkum tohoto typu generativních sítí se více než dalším zvyšováním kvality generovaných vzorků zabývá tím, jak využitelné jsou takové modely. V roce 2021 byl představen StyleGAN3 [22], který dále nezvýšil vizuální kvalitu generovaných snímků, ale soustředil se využitelností na video, zajištěním ekvivariance vůči posunu a rotaci. Další architekturou, již od jiných autorů, je StyleGAN-XL [34], která byla představena v době psaní této práce. Jejím hlavním přínosem je zejména úspěšné škálování na velké různorodé datové sady jako ImageNet.

Obrázek 3.5: Snímky vygenerované celým anycost generátorem a jeho částmi využívající menší počet kanálů či menší cílové rozlišení (převzato z [27], upraveno).

3.1.4 Anycost GAN

Nevýhodu architektury StyleGAN2 [24], kterou je výpočetní náročnost, se pokouší řešit *Anycost GAN* [27]. Vzhledem k tomu, že StyleGAN generátory jsou trénovány tak, aby umožňovaly jednoduché natrénování enkodéru, je možné je velmi dobře použít pro úpravy snímků. Existují způsoby, pomocí kterých lze pro daný generátor vypočítat úpravy stylu, které svojí aplikací povedou k určitým sémantickým změnám – v případě obličejů tím může být například přidání úsměvu, změna barvy vlasů, změna pohlaví, věku, atd. Takové případy užití jsou poté realizovány typicky na zařízeních s menším výpočetním výkonem, než jakým disponují specializované zařízení, na kterých jsou takto velké sítě s desítkami milionů parametrů trénovány. V případě osobního počítače, využívajícího k výpočtům pouze procesor, trvají inference minimálně několik sekund.

Autoři se inspirovali moderními programy pro práci s grafikou, nebo pro úpravu videa. Takové programy uživatelům poskytují v reálném čase náhled výsledku, který poté lze vygenerovat v plné kvalitě, když je to nutné. Stejnou funkcionalitu nabízí i představený anycost generátor. Ten umožňuje generovat náhled výsledku, který byl vygenerován v menším rozlišení pouhou částí generátoru a/nebo použitím menší části kanálů uvnitř sítě, čímž je docíleno až dvanáctinásobného zrychlení. Generátor je trénovaný tak, aby výsledek generovaný libovolnou částí generátoru byl co nejvíce shodný s výsledkem, který byl vygenerován celým generátorem (viz obrázek 3.5).

Schopnost generovat snímky v různých rozlišení spočívá v tom, že každý blok generátoru má možnost produkovat barevný snímek, jak již bylo zmíněno u StyleGAN2. Problémem je, že tyto barevné snímky jsou škálovány a sčítány, a na diskriminátor je přímo napojený jen snímek v nejvyšším rozlišení. Snímky produkované ve vrstvách s nižším rozlišením díky tomu nevypadají realisticky, protože to není cílem trénování. To by částečně řešeno v architektuře MSG-GAN [20], kde každá vrstva produkující barevný snímek v generátoru byla připojena na odpovídající blok diskriminátoru. Tím bylo docíleno, že byly generovány barevné snímky ve všech rozlišeních, ale došlo k poklesu hodnoty metriky FID. Anycost generátor na tomto principu značně zakládá. Během každé iterace je trénováno pouze jediné rozlišení, a části sítě s vyšším rozlišením jsou přeskočeny. Aby bylo zajištěno, že vzorky generované v různých rozlišeních budou co nejpodobnější, je přidán *consistency loss*, skládající se ze *střední kvadratické chyby* a *perceptual loss*. Anycost *subgenerátory* generují kvalitní vzorky, které dosahují nižšího FID, než StyleGAN2 [24] trénovaný pouze v tomto rozlišení [27].

3.2 Syntéza lidských obličejů

Jak bylo v této kapitole dosud ukázáno, syntetizování lidských obličejů je vhodnou počáteční úlohou během vývoje nových typů generativních neuronových sítí. Vzhledem k tomu, že cílová doména nám je tolik přirozená, jelikož lidské tváře vídáme prakticky nepřetržitě celý život, jsme schopni velmi dobře subjektivně určit kvalitu generovaných tváří. Tato vlastnost poté zjednodušuje analýzu natrénovaných modelů, protože v dnešní době používané metriky jako *Fréchet inception distance* (FID) pouze porovnávají distribuce skutečných a generovaných snímků na aktivacích jiné natrénované sítě [14]. Ačkoliv je použití takových metrik nepopíratelně důležité, zachycení drobných nuancí, jakými jsou například různé artefakty v generovaných vzorcích, je pomocí nich prakticky nemožné.

Může se tedy zdát, že generování obličejů je pouze demonstrativní aplikací generativních sítí, ale již existují různá praktická využití této technologie. Přináší však také různá nebezpečí, která jsou nevyhnutelná, a vychází z dosud nevídané věrohodnosti generovaných vzorků.

Generování nových identit

Pokud poskytneme natrénovanému generátoru náhodný šum, výstupem je snímek tváře osoby, která v trénovací sadě neexistuje. Je tomu tak z toho důvodu, že generátor snímky z datové sady nikdy „nevidí“ a učí se distribuci datové sady pouze nepřímo pomocí diskriminátoru. Že tomu tak je, se dá ověřit například pomocí hledání nejbližších sousedů náhodného vzorku v datové sadě (viz obrázek 3.6), měřených na aktivacích nějaké předtrénované sítě.

Obrázek 3.6: Porovnání vygenerovaného obličeje s jeho několika nejbližšími sousedy (podle vpravo dole vyznačené části obličeje) z datové sady FFHQ (převzato z [21], upraveno).

Syntézy takových smyšlených osob by se jistě daly využít například k tvoření reklam, kterým přátelsky vypadající tvář dodává na věrohodnosti, bez potřeby platit skutečné modely [29]. Dalším využitím je samozřejmě rozšiřování existujících datových sad, kterým se tato práce zabývá, ale v rámci této podkapitoly bude pozornost věnována pouze takovým případům užití, které využívají existující tváře jako obrazovou předlohu.

Projekce a úprava tváře

Pomocí enkodéru, který je v případě StyleGAN [23, 24] generátorů možné natrénovat, můžeme v latentním prostoru $\mathcal{W}$ najít co nejlepší reprezentaci libovolného snímku, který pomocí generátoru lze poté různě upravovat. Prvním způsobem je míchání stylů (viz obrázek 3.3), pomocí kterého je možné měnit různé rysy, ať už hrubé, jako pohlaví, stáří,

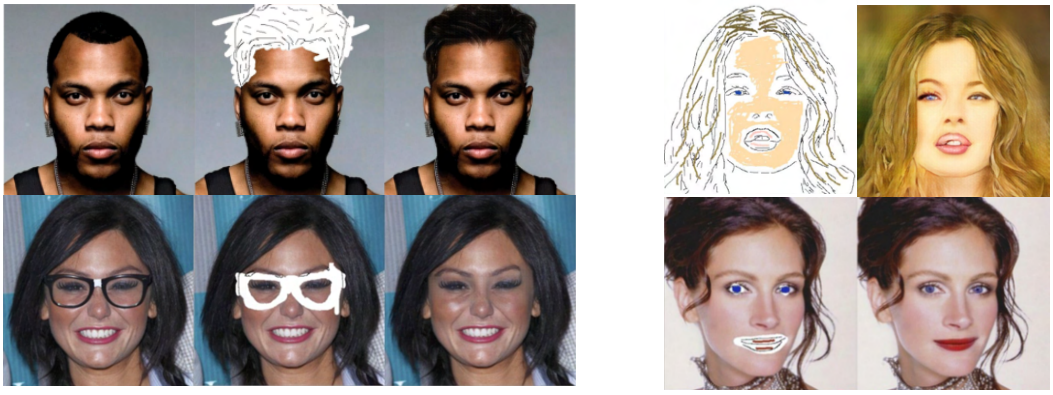
barva pleti či natočení hlavy, tak velmi jemné, jako například pouze barva pozadí. V tomto ohledu je ale zajímavější a předvídatelnější upravování přímo stylu w. Bylo zjištěno, že při interpolaci mezi dvěma body $w_1, w_2 \in \mathcal{W}$ dochází ke spojitým změnám výsledného snímku a implicitně i sémantických informací, které snímek reprezentuje. Pro binární sémantickou informaci (pohlaví, úsměv, ...) poté existuje v latentním prostoru rozhodovací hranice [36]. S použitím této hranice tedy můžeme vypočítat vektor Δw, který sečtením s původním stylem w vede k očekávané změně originálního snímku, jak je vidět na obrázku 3.7.

Obrázek 3.7: Úpravy tváře pomocí vypočítaných vektorů Δw.

Úpravy bez manipulace latentního kódu

Setkat se můžeme i s dalšími metodami úprav obličejů, kde nejsou využívány StyleGAN generátory a jejich latentní prostor. Takové sítě často využívají například architekturu *U-Net* v generátoru, aby vstupem mohl být místo šumu přímo snímek k upravení. Jelikož jsou tyto úlohy více specializované, žádá si jejich trénování použití složitějších chybových funkcí.

Dokreslení tváře. Cílem je umožnit generovat obličeje na základě náčrtů, či podle nich doplňovat chybějící místa v již existujícím snímku. Příkladem systému provádějící takovou úlohu může být SC-FEGAN [18]. Zde je vstupem sítě původní obrázek s maskou značící oblasti k doplnění, a poté jako vstup uživatele je náčrt a barevná maska, které slouží jako „návod“ pro síť, jakým způsobem má původní obrázek dokreslit. Autoři dosáhli kvalitních výsledků dokonce i v případech, kdy je třeba snímek dokreslit celý, viz obrázek 3.8.



Obrázek 3.8: Dokreslování obrázků pomocí systému SC-FEGAN (převzato z [18], upraveno).

Obrázek 3.9: Frontalizace různě natočených tváří pomocí metody Dual-Attention GAN (převzato z [42], upraveno).

Frontalizace tváře. Jedná se o úlohu, při které je cílem vygenerování syntetické tváře dívající se přímo dopředu ze vstupního snímku, na kterém může být hlava natočená libovolným směrem. Klíčové je samozřejmě co nejvěrněji zachovat identitu člověka na původním snímku. Aktuální state-of-the-art řešení v této oblasti staví na generativních neuronových sítích a využívá v dnešní době velmi populární *attention* mechanismus, využívaný v sítích typu Transformer [41]. Tímto řešením je *Dual-Attention GAN* [42], které spolehlivě produkuje věrné frontalizace subjektů natočených i o 90°, jak je vidět na obrázku 3.9. Praktické využití této transformace lze najít při vytváření datových sad, kde může sloužit jako augmentace dat. Obzvlášť vhodná se tedy může projevit při trénování s malými datovými sadami, kde použití frontalizace může vést ke zlepšení výsledků sítě [42].

Image-to-image translation (I2I). Doslovně přeloženo jako *překlad obrázku na obrázek*, je další působivou aplikací generativních neuronových sítí. Základní úlohou takových sítí je poté hledání mapování z jedné do druhé domény. Výsledný obrázek je potom tvořen na základě struktury vstupního obrázku a uměleckého stylu cílové domény. GANy excelují jak v supervised I2I (metoda *pix2pix* [17]), tak v unsupervised I2I (metoda *CycleGAN* [47]), které není závislé na získání velkého počtu dvojic dat z obou domén.

Problém, který je v této oblasti stále nedořešený, je trénování s nevyváženými datovými sadami. Takové modely sice zvládají generovat realistické snímky, ale selhávají při mapování struktury ze vstupního snímku. Ukázalo se, že k tomuto účelu lze využít i StyleGAN [24] architekturu, pomocí transfer learning. Síť předtrénovaná pro generování fotorealistických tváří je dotrénována pomocí datové sady reprezentující cílovou doménu. Jelikož struktura cílového obrázku je generována v počátečních vrstvách, dodržování stejné struktury vynutíme přidáním chybové funkce, která porovnává RGB výstup cílové sítě a zdrojové v těchto vrstvách pomocí střední kvadratické chyby. Nakonec pomocí techniky zvané *layer swapping* nahradíme počáteční vrstvy cílové sítě těmi ze zdrojové a inferencí získáváme snímky, které úspěšně napodobují strukturu tváří v požadovaném stylu [2]. Ukázka této metody při mapování lidských tváří do různých cílových domén je vidět na obrázku 3.10.

Obrázek 3.10: Přesun struktury tváře do cílové domény (převzato z [2], upraveno).

Potenciální negativní dopady na společnost

Takto rychlý rozvoj technologií schopných produkovat fotorealistické syntetické snímky ovšem nese také své stinné stránky. S tím, jak přibývají nové aplikace generativních modelů a u již existujících se zvyšuje jejich důvěryhodnost, zvyšuje se také jejich potenciál pro použití jak v oblastech, kde mohou být užitečné, tak také v oblastech, kde mohou být použity s nekalými úmysly [22]. Jakákoliv fotografie na internetu může být vygenerovaná a je jen velmi malá šance, jak na to přijít. V případě falešných profilů na sociálních sítích bývalo k určení jejich důvěryhodnosti užíváno reverzního vyhledávání fotografií, a nyní jsou k dispozici generativní modely, které jsou schopné vygenerovat prakticky nekonečné množství unikátních identit, nedohledatelných pomocí takového systému.

Na základě nedávno provedené studie bylo dospěno k závěru, že generované obličeje dosahují takových kvalit, že jsou prakticky nerozeznatelné od skutečných. V první části studie 315 účastníků dosáhlo při klasifikaci průměrné přesnosti pouze 48,2 %, i přes to, že jim byly slíbeny peníze navíc, pokud ve srovnání s ostatními účastníky dosáhnou nadprůměrných výsledků. V druhém experimentu bylo 219 účastníků, kteří měli stejnou úlohu a motivaci, ale byli informováni o tom, jak poznat syntetický obličej, a po každém obličeji měli k dispozici zpětnou vazbu, zda byl jejich odhad úspěšný. I přes to, průměrná přesnost se zvýšila pouze na 59 %. Třetí experiment se týkal důvěryhodnosti tváří, a syntetické tváře dosáhly o 7,7 % lepšího hodnocení, než skutečné [29].

Způsob, jak jednoduše lze fotorealistické obličeje získat spojený s tím, jak důvěryhodné a věrohodné se zdají, je tedy jasným důvodem, proč se na internetu začínají objevovat. Na sociální síti LinkedIn bylo identifikováno nejméně 1000 profilů užívajících generované profilové fotografie [4]. Podobný princip lze aplikovat i na videa, s image-to-image translation modely, takzvané *DeepFakes*, které mění obličej ve zdrojovém videu za jiný. V této oblasti nebylo dosud dosaženo skutečné nerozeznatelnosti falešných videí a generovaná videa vyvolávají zatím spíše efekt *uncanny valley* (česky *strašidelné údolí*), kde vyobrazené osoby vypadají jako lidé, ale typicky existují nějaké artefakty bijící do očí a prozrazující, že jde o padělek. I přes to se ale objevují instance, kde byly taková videa použity k pokusům o krádež, nebo o ovlivnění strany protivníka během války.

Jde o problém, který může velmi rychle přerůst ve velmi vážný, jelikož v blízké budoucnosti mohou existovat nerozeznatelné video i audio DeepFakes [4]. Existují výzkumy o trénování modelů sloužících k odhadování syntetických snímků, a v současné době dosa-

hují i dobrých výsledků, ale vzhledem k povaze generativních neuronových sítí, toto povede pouze k trénování lepších modelů, které tuto metodu ochrany překonají. Jako mnohem robustnější řešení se ukazuje například přidání otisků přímo do trénovacích dat [43].

3.3 Způsoby výběru věkové kategorie

V oblasti state-of-the-art generování existují dva hlavní způsoby, kterými je možné syntetizovat tváře na základě vybrané věkové kategorie, přičemž nejsou omezené pouze na věk generovaných tváří, ale fungují obecně pro jakoukoliv cílovou doménu a zvolenou podmínku.

Úprava latentního kódu

Dříve zmíněnou techniku užívanou pro sémantické úpravy obrazu lze využít i pro ovládání výsledné věkové kategorie v případě generování obličejů, jak bylo demonstrováno v různých pracích [27, 36]. Problém však dělá to, že vektory Δw jsou počítány na základě rozhodovací hranice binárních sémantických informací. Pokud se jedná konkrétně o věkové kategorie, ty můžeme do dvou tříd rozdělit na „mladí“ a „staří“. Takové rozdělení ale neumožňuje jemnou kontrolu nad věkem výsledného obličeje. Při interpolacích stylu mezi oběma třídami sice dochází ke spojité změně zdánlivého věku, ale nelze přesně určit jeho hodnotu u takto generovaných snímků.

V rámci této práce jsem této možnosti výběru věkové kategorie nevyužil, ale přesto zde využívám snadné invertovatelnosti StyleGAN [24] generátoru k natrénování enkodéru, který jsem implementoval jako další způsob augmentace dat pro zlepšení regularizace. Nabízí se ale prostor pro další výzkum, jak během trénování regularizovat latentní prostor generátoru tak, aby spolehlivě uchovával takové informace a bylo je možné přesně inferovat.

Podmíněné generativní neuronové sítě

Alternativním přístupem je rozšíření základních nepodmíněných sítí o podmínky, matematicky řečeno, změnit generátor na $G(z|y)$ a diskriminátor na $D(x|y)$, kde y jsou *značkami* (anglicky *label*) jednotlivých tříd a x je klasifikovaným vzorkem. Tyto značky bývají typicky reprezentovány celočíselnou hodnotou z konečného počtu možných hodnot, která je potom vstupem *vkládací* (anglicky *embedding*) vrstvy, konvertující jí na vektor fixní délky, který je poté dalším vstupem jednotlivých sítí.

Jde o poměrně jednoduchý způsob, jakým umožnit kontrolu nad generovanými vzorky a v oblasti GANů je používaný už od jejich počátku [28]. Rozdělení trénovacích vzorků do smysluplných tříd a trénování s nimi je také spojováno se zvýšenou stabilitou trénování, jelikož sítě se tyto informace učí implicitně.

V této práci jsem úspěšně rozšířil nepodmíněný model StyleGAN2 [24] o podmíněné generování na základě věku a pohlaví, které je snadno rozšiřitelné i o další třídy. Konkrétněji o tomto bude pojednáno později, v kapitole 5.

Kapitola 4

Datové sady

Poměrně náročnou částí této práce bylo utvoření vhodné datové sady, pomocí které budu všechny modely trénovat a vyhodnocovat. Jedním z cílů, který si tato práce vytyčila, bylo použití syntetických dat ke zpřesnění klasifikací věku především v méně zastoupených věkových kategoriích, kde typicky bývá přesnost horší. Ačkoliv méně zastoupené jsou typicky věkové kategorie dětí a seniorů, více praktické z hlediska využitelnosti je soustředění se na zpřesnění klasifikací věku dětí. Tento cíl tedy významně ovlivňoval výběr datových sad, protože pouze malá část již dostupných obsahuje také anotované fotografie dětí.

Vhodné datové sady přesto přináší různé unikátní překážky. V rámci jedné datové sady nebo při jejich kombinování mohou vznikat věkové, pohlavní či rasové nevyváženosti. Jednotlivé snímky mohou dosahovat rozličných vizuálních kvalit, nemusí vždy obsahovat obličej či může existovat jednoznačný *bias* v rámci datové sady, který způsobí, že variabilita vzorků napříč datovými sadami může být velmi vysoká.

Pokud se přesuneme od snímků k anotacím, i u těch se objevují další výzvy. Vzhledem k povaze toho, jak část datových sad vzniká – získáním veřejných nelicencovaných snímků, může docházet k nekonzistencím v anotacích. Pokud se jedná o věk, pouze u části vzorků lze získat *skutečný* věk. Často je tedy nutné zbylá data anotovat ručně, s čímž pomáhají crowdsourcingové služby, kde je možné najmout si pracovníky na dálku na různé manuální činnosti, kterou je právě například anotace obrázků. Tímto způsobem můžeme získat alespoň *zjevný* věk, byť to přidává určitou míru nejasnosti do dat. Z toho důvodu některé datové sady u vzorků neposkytují konkrétní věk, ale rozsahy, ve kterých se s největší pravděpodobností skutečný věk vyskytuje.

V této kapitole se krátce podíváme na všechny použité datové sady v rámci této práce, způsob, kterým byly filtrovány a na statistiky vytvořené finální datové sady.

Appa-real

Jak název datové sady naznačuje, jejím specifikem je to, že byla vytvořena společně s anotacemi skutečného i zjevného věku, kde každý snímek byl ohodnocen průměrně 38 lidmi [1]. Jedná se o celkem malou datovou sadu, čítající pouze bezmála 7600 vzorků. Anotace skutečného věku mají granularitu 1 rok. Další výhodou, kterou zde crowdsourcing přinesl, bylo to, že byly vyfiltrovány nekvalitní snímky a snímky s více obličeji. Zároveň je poměrně dobře věkově vyrovnaná, bohužel kvůli malému počtu vzorků tato kvalitní vlastnost mezi ostatními datovými sadami zaniká.

UTKFace

Podobně vizuálně kvalitními snímky a rozumně vyváženým věkem disponuje také datová sada UTKFace[1]. Počet vzorků je zde vyšší, přes 24 tisíc. Další výhodou je, že zde byl kladen větší důraz na zastoupení co nejvíce etnik. Bohužel, vzhledem k tomu, že jde o data získaná *in-the-wild* (data volně dostupná na internetu), anotace věku jsou odhadnuté pouze strojově, pomocí již dříve zmíněné metody DEX, nicméně byly alespoň ručně překontrolovány během čištění dat.

OUI-Adience

Specifikem této datové sady[2] je to, že záměrně obsahuje takové snímky, které co nejlépe reprezentují „skutečné“ podmínky. Mnoho vzorků obsahuje více obličejů, komplikované světelné podmínky a objevují se různé úrovně kvality snímků. Velkou výzvu pro trénování neuronových sítí přináší i anotace věku, které mají velmi hrubou granularitu, věkové kategorie 0–60 let jsou rozděleny celkem do 8 různých skupin, z nichž 4 patří dětem a mladistvým. Skládá se z více než 26 tisíc vzorků a vzhledem ke svým vlastnostem je využívána jako „zátěžový test“ k porovnání různých metod.

FRGC-IJCB

Původně vytvořená pro účely soutěže pořádanou vládou USA, tato datová sada[3] je rozdílná zase tím, že neobsahuje in-the-wild fotografie. Vytvořená byla specificky pro potřeby soutěže, a proto obsahuje 60 tisíc vzorků čtyř tisíc různých osob, kde jednotlivé fotografie jsou pouze velmi málo odlišné. Snímky jsou ve velmi vysoké kvalitě, a výhodou jsou zde přesné anotace věku, ovšem je zde velmi silný bias věkové kategorie mladých dospělých, kteří tvoří většinu této datové sady.

Flickr-Faces-HQ

Spíše známá pod zkratkou FFHQ[4], tato vizuálně velmi kvalitní datová sada se proslavila zejména v oblasti generativních neuronových sítí, jelikož byla představena společně s metodou StyleGAN [23]. Z toho důvodu zde byl kladen důraz především na co největší variabilitu vzorků v co nejvyšším rozlišení. Vzhledem k tomu, že primární užití bylo v oblasti nepodmíněného generování, neexistují oficiální anotace. V rámci tohoto projektu tedy používám strojové anotace, které mi byly poskytnuty. FFHQ se řadí mezi větší datové sady, které jsem použil, tvořena je 70 000 vzorky.

FairFace

Pravděpodobně nejdůkladněji vytvořenou datovou sadou z celé šestice je FairFace [19]. Jejím cílem byla právě „férovost“ vůči všem etnickým skupinám, jelikož v posledních letech se začaly častěji objevovat případy, kdy právě komerční aplikace strojového učení prokazovaly sníženou přesnost na rozdílných rasách. I datové sady jako UTKFace, které se vliv etnik snažily minimalizovat, byly predominantně tvořeny bílou rasou, poměrem vyšším než 50 %. V méně zpracovávaných datových sadách tento podíl roste až k 90 % [19]. Menší nevýhodou

[1] https://susanqq.github.io/UTKFace/
[2] https://chalearnlap.cvc.uab.cat/dataset/26/description/
[3] https://www.nist.gov/programs-projects/face-recognition-grand-challenge-frgc
[4] https://github.com/NVlabs/ffhq-dataset

je také to, že velká část tváří je natočená různými směry ve velmi velkých úhlech, ale to lze poměrně snadno filtrovat.

Výsledkem práce autorů bylo shromáždění více než 100 tisíc vzorků, vyvážených podle 7 etnických skupin. Byl kladen důraz i na vyvážení podle věku a pohlaví, bohužel ale jde ale o anotace najatými pracovníky do 9 věkových kategorií. Osoby mladší 20 let jsou zde zařazené do jedné z kategorií 0–2, 3–9, 10–19, což není pro potřeby přesné klasifikace příliš ideální.

4.1 Filtrování dat

Je zřejmé, že při slučování mnoha různých datových sad může dojít k datovým nekonzistencím. Ačkoliv pro člověka jsou tyto rozdíly zanedbatelné, v případě neuronových sítí by mohly přinášet problém a potenciálně omezit stabilitu trénování a dosaženou přesnost. Pravděpodobně nejdůležitější je sjednocení polohy tváře ve snímku. Cílem je mít takovou sadu, kde budou všechny tváře stejně velké, a budou co nejlépe zarovnány na střed. Pro to je využito metod již zmíněných v podkapitole 2.1. Toto pomáhá při trénování jak klasifikátoru věku, tak generátoru, jelikož v této práci využitý generátor založený na metodě StyleGAN2 [24] není invariantní vůči posunu.

Rovněž vhodné je vyfiltrovat vzorky, které jsou odlehlé v rámci celé nové distribuce. K určení, o jaké vzorky se jedná, jsem využil anotace, které mi byly pro mé datové sady vygenerovány a poskytnuty společností Innovatrics. Z mnoha dostupných anotací popisujících vlastnosti daného snímku jsem poté vybral jas, rotací okolo os Y a Z (sklon a otočení hlavy) a skóre, kterým jejich nástroj ohodnocuje vizuální kvalitu snímku. Pro každou z těchto vlastností jsem na základě subjektivních dojmů určil podíl nejvzdálenějších vzorků (2–5 %), který bude smazán. Užitím tohoto podílu p jsem následně u jednotlivých vlastností (kromě skóre) ze všech hodnot určil kvantily $Q_{p/2}$ a $Q_{100-p/2}$, a veškeré vzorky, jejichž hodnoty byly mimo rozsah $\langle Q_{p/2};\ Q_{100-p/2}\rangle$ jsem odstranil. Stejně tak byly pročištěny veškeré vzorky, kterým chyběly anotace věku a ty, na nichž detektor nedetekoval tvář. Na výběr vyfiltrovaných snímků podle jednotlivých vlastností je možné se podívat na obrázku 4.1.

(a) Jas (b) Rotace hlavy (c) Skóre

Obrázek 4.1: Nezarovnané snímky, které byly předem vyfiltrovány na základě anotací popisujících jejich kvalitu.

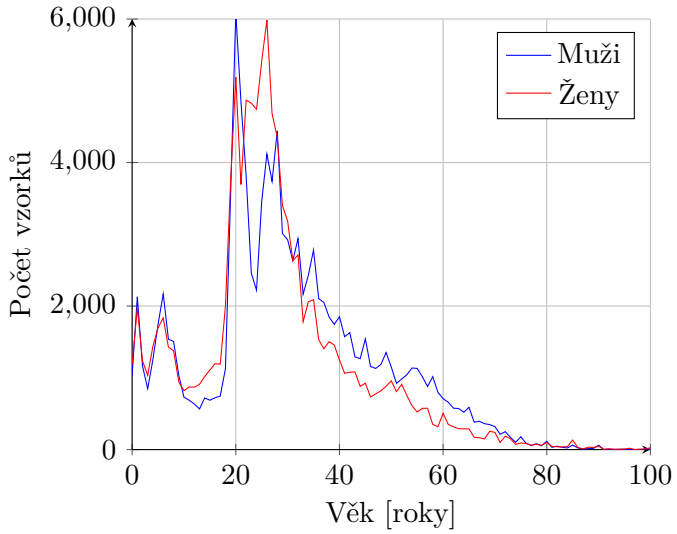
4.2 Vlastnosti vytvořené datové sady

Po sjednocení všech zmíněných datových sad a smazání vzorků, na kterých nebyla detekována tvář, bylo dosaženo celkového počtu 278 919 vzorků. Po filtrování chybějících anotací a velmi vysokých či nízkých hodnot sledovaných vlastností zbylo 233 368 vzorků, detailní statistika filtrovaných vzorků je k vidění v tabulce 4.1.

	Datová sada					
Filtrovaná hodnota	Appa-real	UTKFace	OUI	FRGC	FairFace	FFHQ
Neplatné hodnoty	25	8	763	15893	68	0
Sklonění tváře	109	122	375	220	3313	791
Otočení tváře	37	14	39	0	9319	211
Jas	495	1175	1189	1612	6484	2154
Skóre	28	70	69	9	959	0
Celkem	**694**	**1389**	**2435**	**17734**	**20143**	**3156**

Tabulka 4.1: Statistika filtrování na základě anotací jednotlivých datových sad.

Filtrované a zarovnané vzorky byly poté rozděleny do tří množin – trénovací, validační a testovací, v poměru zhruba 76 : 12 : 12. Výsledná datová sada je dobře vyvážená podle pohlaví, tvoří jí 117 566 snímku mužů a 115 802 žen. Rozložení věku je k vidění na grafu 4.2.



Obrázek 4.2: Rozložení věku a pohlaví ve výsledné datové sadě.

Jak je v grafu vidět, z hlediska věku datová sada není vyrovnaná tak přesně, jako podle pohlaví, ale přesto vrchol, který je mezi 20–35 lety nezpůsobuje problém při trénování, jak bude vidět později v kapitole Vyhodnocení výsledků. S omezením maximálního věku na 70 let během trénování jsou všechny věkové kategorie dostatečně zastoupené. Ke zjemnění granularity, kterou do datové sady přináší datové sady, které nedisponují přesnými anotacemi věku, byl využit klasifikátor věku poskytnutý společností Innovatrics.

Kapitola 5

Implementace

S použitím velké datové sady lidských obličejů, kterou jsem v rámci této práce vytvořil, jsem následně trénoval tři různé typy konvolučních neuronových sítí. Tato kapitola se konkrétněji zabývá způsobem, kterým jednotlivé sítě byly implementovány, jak byly laděny, hledány správné nastavení hyperparametrů a v neposlední řadě tím, jak bylo trénováno velké množství modelů během prováděných experimentů.

5.1 Klasifikátor věku

Jelikož cílem této práce bylo zkoumat vliv přidání syntetických dat na přesnost odhadování věku, je rozumné začít implementací právě této sítě, aby šlo co nejdříve stanovit *baseline* na skutečných datech, kterou bude cílem zlepšit.

Použité knihovny a nástroje

Tato síť, stejně jako všechny zbylé v této práci, je implementována v jazyce *Python*, který je v dnešní době prakticky standardem v oblasti strojového učení. Konkrétněji, klasifikátor jsem implementoval pomocí knihovny *TensorFlow*[1] ve verzi 2.8.0. Jádro knihovny TensorFlow je spíše nízkoúrovňovým API, umožňujícím jemnou kontrolu nad trénováním a strukturou sítě. Tato funkcionalita se hodí při implementaci složitějších modelů sítí, jakými jsou například generativní neuronové sítě, ale pro účely klasifikátoru bylo dostačující použít vysokoúrovňové API *Keras*, které je společně s TensorFlow distribuováno. Dalšími výhodami kromě efektivního trénování a dobré podpory zpětných volání Keras také snadno umožňuje použít předtrénované modely známých architektur jako VGG [37] a EfficientNet [39, 40].

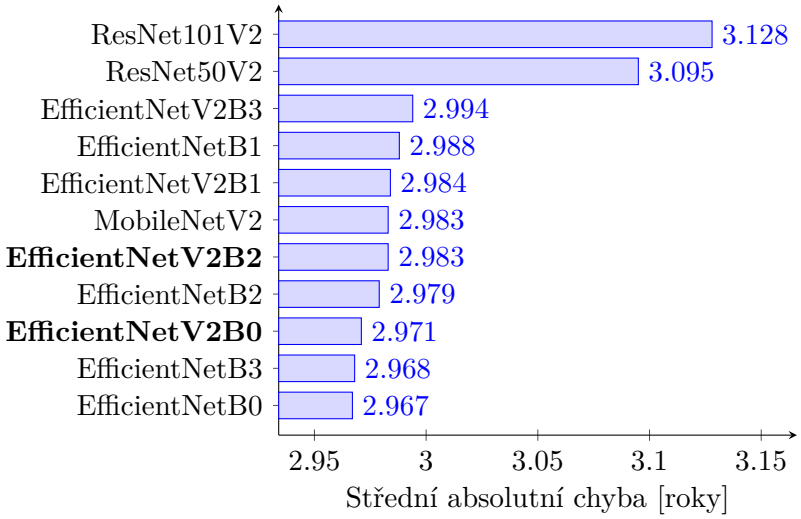
Kvůli vysokému počtu experimentů, které byly v rámci vývoje a evaluací prováděny s tímto modelem, jsem věnoval zvýšené úsilí tomu, aby byly veškeré hyperparametry snadno nastavitelné a také, aby bylo možné jednotlivé experimenty co nejefektivněji spouštět a porovnávat jimi dosažené výsledky. Jako velmi užitečný se ukázal framework *Hydra*[2], vytvořený společností *Meta*, jehož hlavním cílem je zjednodušení konfigurace komplexních aplikací, mezi které se strojové učení řadí. Využití jednoduchého kombinování nastavení z konfiguračních souborů s argumenty z příkazové řádky, a možností spustit program pro všechny možné kombinace zvolených parametrů výrazně usnadnilo hledání správných hodnot různých hyperparametrů. Pro porovnání dosažených trénovacích metrik se osvědčila

[1] https://www.tensorflow.org/
[2] https://github.com/facebookresearch/hydra

MLOps platforma *Weights & Biases*[3], která slouží ke sledování a porovnávání všech uplynulých či probíhajících trénování neuronových sítí v reálném čase.

Výběr modelu

Nejmodernějšími jsou v dnešní době modely EfficientNet, o kterých byla řeč v kapitole 2. Výhodou použití API Keras je mimo jiné také to, že nejen tyto modely, ale mnoho dalších, jsou přímo přístupné[4] bez nutnosti cokoliv implementovat, společně s natrénovanými váhami na datové sadě ImageNet. Původní EfficientNet [39] architektura byla poté ještě vylepšena, když došlo k představení *EfficientNetV2*, pomocí které autoři dosáhli ještě větší rychlosti inference a vyšší přesnosti [40]. V tomto roce byly novější modely také přidány v nové verzi Keras. Na následujícím grafu 5.1 lze vidět, jakých přesností bylo dosaženo použitím různých dostupných modelů.



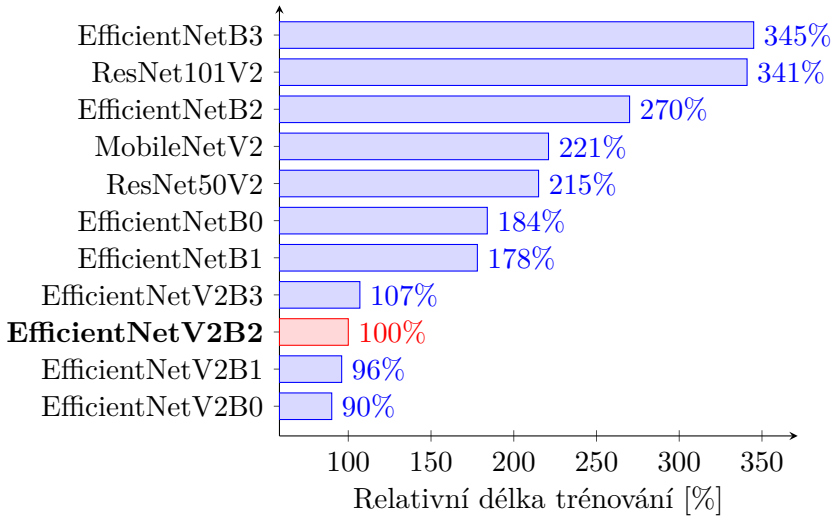
Obrázek 5.1: Velikosti chyby na validační datové sadě dosažené různými dostupnými modely.

Je zde vidět, že novější modely dominují, ale rozdíl v počtu jejich parametrů už nedělá takový rozdíl v dosažené přesnosti. V případě modelů EfficientNet platí, že velikost modelu B3 je zhruba dvojnásobná oproti modelu B0. V grafu byly vynechány modely VGG16 a VGG19, které se mi nepodařilo úspěšně natrénovat kvůli nestabilnímu trénování. Všechny modely byly pro účely porovnání trénovány pouze s barevnými snímky z datové sady v rozlišení 224×224. Abych vybral model, který budu používat, vzal jsem v potaz také rychlost jejich trénování, jelikož s nimi bude prováděno velké množství experimentů. Rozhodl jsem se pro použití modelů *EfficientNetV2B0* a *EfficientNetV2B2*, přičemž primárně jsem používal větší z těchto dvou modelů, jelikož dosahoval lepších výsledků po přidání syntetických dat. Na grafu 5.2 je vidět délka trénování relativní k modelu EfficientNetV2B2 s použitím grafické karty NVIDIA Tesla P100.

Augmentace dat

Mnoho experimentů se také týkalo nastavení augmentace dat, za cílem získání co nejkvalitnějšího baseline výsledku, před přidáním syntetických dat. Vyzkoušel jsem vliv mnoha

[3] https://wandb.ai/
[4] https://keras.io/api/applications/



Obrázek 5.2: Relativní doba trénování různých modelů v porovnání s EfficientNetV2B2.

různých typů transformací dostupných v knihovně *Albumentations*[5], se zaměřením především na zlepšení generalizace v jevech, které se hojně vyskytují v datových sadách, jako jsou různé světelné podmínky, rozmazání, komprese snímku, atd. Na základě dosažených výsledků bylo zjištěno, že i „náhodné“ kombinace transformací vedou k drastickému zlepšení validační přesnosti. Demonstrovat to lze tak, že bez použití transformací bylo na reálných datech dosaženo střední absolutní chyby 4,38 roku, a nejhorší dosažený výsledek během testování různých kombinací transformací, jejich intenzit a pravděpodobností její aplikace měl *střední absolutní chybu* (MAE) 3,69 roku. Dalším laděním bylo postupně docíleno snížení průměrné chyby na 3,40 roku. Jako nejužitečnější se ukázala následující kombinace transformací:

- **Posun**: 3 % limit, 75 % pravděpodobnost.
- **Škálování**: 5 % limit, 75 % pravděpodobnost.
- **Rotace**: 5° limit, 75 % pravděpodobnost.
- **Změna kontrastu**: 20 % limit, 50 % pravděpodobnost.
- **Změna jasu**: 20 % limit, 50 % pravděpodobnost.
- **Horizontální převrácení**: 50 % pravděpodobnost.
- **Převod do odstínů šedi**: 5 % pravděpodobnost.

Tyto transformace jsou během trénování s danou pravděpodobností aplikovány na každý skutečný i syntetický vzorek. Efekt této transformace je omezen nastaveným limitem. Na obrázku 5.3 je vidět, jakým způsobem tyto náhodné transformace různě mění stejný vstupní snímek.

[5] https://albumentations.ai/

Obrázek 5.3: Opakované aplikace transformací na stejné vstupní snímky.

Výpočet chyby a distribucí anotací věku

Jak již bylo zmíněno v dřívější části práce, state-of-the-art metodou pro klasifikaci věku je DLDL-v2 [9]. Pouhým nahrazením původní chybové funkce, kterou byla kategorická křížová entropie, se mi podařilo snížit velikost průměrné chyby o zhruba 5 %. Tato změna umožnila úpravy dalších parametrů. Kromě ladění poměru obou chybových funkcí dávající dohromady DLDL-v2 šlo také upravovat, jakým způsobem bude počítána směrodatná odchylka distribucí anotací věku, jelikož použitím této metody se s nimi pracuje jako s rozloženími pravděpodobností.

Vzhledem k tomu, že anotace z datové sady jsou dostupné jako konkrétní hodnoty nebo v horším případě rozsah hodnot, bylo nutné experimentovat se způsoby, jakým bude pro různé hodnoty vytvářeno rozložení pravděpodobností, na základě kterého bude poté počítána chyba predikcí. Jako základní řešení jsem proto zvolil použití směrodatné odchylky $\sigma = 1$ pro data s přesnými anotacemi, $\sigma = 2$ pro generovaná a augmentovaná data. Použití pevně zvolených směrodatných odchylek bylo doporučováno v případě, kdy skutečná rozložení nejsou známa [9]. V případě vzorků s rozsahy věku jsem σ zvyšoval podle velikosti daného rozsahu. Tento parametr jsem ještě poté podrobil dalším experimentům a podařilo se mi dosáhnout několikaprocentního zlepšení změnou výpočtu směrodatné odchylky pro oba typy vzorků následovně:

$$\sigma = 1 \ + \frac{\mu}{25}, \tag{5.1}$$

kde μ je střední hodnota výsledného rozdělení, které je v tomto případě věkem. Předpoklad byl takový, že v případě nižšího věku jsou více znatelné rozdíly ve vzhledu, než v případě starších osob. Konkrétní hodnota 25 byla poté určena na základě experimentů.

Eliminace rozsahů věku

Společně s anotacemi popisujícími kvalitu snímků mi byly společností Innovatrics poskytnuty také jimi získané odhady věku na mých datových sadách. Tyto predikce jsem nevyužil k trénování tam, kde již mám přesné anotace věku, ale využil jsem je k pokusu o redukci vlivu nejednoznačných anotací na přesnost sítě. Výpočet probíhal následovně. Pokud predikce byla v rozsahu *ground-truth* anotací, použil jsem na trénování tuto konkrétní hodnotu. Pokud se nacházela mimo daný rozsah, rozdělil jsem ho na dvě půlky a použil jsem k trénování jeho nižší, případně vyšší část, podle toho, zda predikovaný věk byl nižší nebo vyšší,

než jsou hranice rozsahu. Z této poloviny rozsahu jsem jako střed rozdělení μ vybíral náhodnou hodnotu. Touto změnou jsem docílil poměrně výrazného snížení naměřené chyby, o zhruba 10 %.

5.2 Generativní neuronové sítě

Po natrénování klasifikátoru bylo dalším cílem přidání podmíněného generování syntetických tváří. Zde bylo využito oficiální open-source implementace architektury Anycost GAN[6], která byla posléze upravena tak, aby lépe vyhovovala trénovacím datům a potřebným případům užití.

Použité knihovny a nástroje

Původní open-source řešení, na kterém jsem zakládal, využívalo druhou velmi používanou knihovnu pro strojové učení, *PyTorch*[7], ve verzi 1.7.1. Pro monitorování jednotlivých běhů a výstupů využívá nástroj *TensorBoard*[8]. Ačkoliv během integrace tohoto modelu do již existujícího projektu byla potřeba část zdrojového kódu nahrazovat, nebylo nutné sjednocovat použité knihovny a nástroje. Reprezentace tenzorů obou knihoven sice nejsou navzájem kompatibilní, ale společnou komunikaci sítí lze zajistit pomocí knihovny *NumPy*[9], která je jejich knihovnami plně podporována.

Úpravy modelu

Váhy natrénovaného modelu sítě, které byly zveřejněny společně se zdrojovým kódem, byly získány při trénování v rozlišení 1024×1024 pixelů. Taková velikost generovaných tváří není při klasifikaci věku nutná, a proto jsem využíval generátor pouze v rozlišení 256×256. Z toho důvodu jsem z původních vah odstranil váhy vrstev vyššího rozlišení, a model jsem dotrénoval na mnou vytvořené datové sadě s tímto nižším rozlišením. Pro mnou vytvořenou datovou sadu jsem také získal aktivace na předtrénované *InceptionV3* síti, které jsem poté využíval pro výpočet FID během trénování, jako ohodnocení kvality generovaných vzorků.

Důvod, proč jsem si původně zvolil architekturu Anycost GAN [27] nad StyleGAN2 [24], za které vychází, vycházel z předpokladu, že bude potřeba využít režim, kdy je omezen počet kanálů za účelem zrychlení inference. Ten se ovšem po integraci generátoru do klasifikátoru ukázal jako mylný, nakolik získání vzorků pomocí generátoru nakonec nebylo tak pomalé, jak jsem původně očekával. Již snížení rozlišení na 256×256 poskytlo významné zrychlení a ve spojení s tím, že se ke zvýšení přesnosti osvědčily relativně malé poměry syntetických dat, bylo tímto trénování zpomaleno pouze decentně. Z toho důvodu jsem nakonec z modelu odstranil možnost generování s flexibilním počtem kanálů a různými rozlišeními.

Podmíněné generování

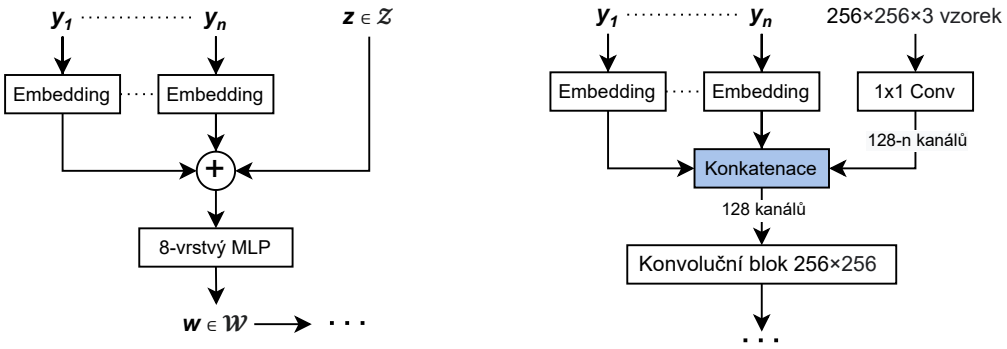
Během podmíněného generování je zapotřebí k oběma sítím přidat vstup y, který je vektorem zvolených hodnot každé z n vlastností. Pro každou síť jsem přidal n embedding vrstev, které konvertují hodnoty $y_1, \ldots, y_n$ na vektory fixní délky. V případě generátoru toto bylo realizováno ještě před získáním stylu w. Získané vektory jsou přičteny k vstupnímu šumu z

[6] https://github.com/mit-han-lab/anycost-gan/
[7] https://pytorch.org/
[8] https://www.tensorflow.org/tensorboard
[9] https://numpy.org/

a teprve poté je získána jeho projekce v latentním prostoru $\mathcal{W}$. V tomto případě je řešení velmi intuitivní, jelikož stejné úpravy přímo stylu w by nejspíše komplikovaly regularizaci, která je během trénování k „zjednodušení“ $\mathcal{W}$ prováděna. Navíc, přidáním stylu ke vstupnímu šumu probíhá učení již v plně propojené síti, která mapuje z na w.

Přidání podmínek do diskriminátoru bylo náročnější a žádalo si několik experimentů. Zde již bylo několik možností, kam se dá výstup embedding vrstev přidat. Nabízí se embedding spojit s výstupem finální konvoluční vrstvy, před vstupem do plně propojených vrstev, které tvoří výslednou predikci. Výhodou je, že rozlišení v této části sítě je tak malé, že samotné embeddingy nepřidají mnoho trénovatelných parametrů. Nevýhodou je, že po pár experimentech s přidáním třídních informací na konec sítě se mi nepodařilo najít takové nastavení, které by vedlo ke stabilnímu trénování, během kterého by byl generátor nucený diskriminátorem se naučit podmíněně generovat. Zbývající možností tedy bylo přidání informací na začátku sítě. Nakonec jsem výstupy embeddingů přidal jako nové kanály druhé konvoluční vrstvy, na kterou se připojuje vstupní konvoluční vrstva s 1×1 konvolucemi, která zvyšuje dimenzionalitu vstupních dat. Při přidávání nových tříd je tuto vrstvu nutné vždy inicializovat znovu, aby se nezměnil počet vstupních kanálů druhé konvoluční vrstvy. Zmíněná místa, kde byly napojeny embedding vrstvy v obou sítích, jsou graficky znázorněná na obrázku 5.4.

Dalšími změnami spojenými s podmíněným generováním bylo upravení výpočtu *zkrácení stylu*, který je i s touto architekturou možné použít. Během něj je z mnoha vzorkovaných šumů získáván průměrný styl a poté je prováděn vážený součet s původním stylem k získání nového, méně odlehlého, stylu. Po přidání anotací bylo vhodné předělat výpočet průměrného stylu takovým způsobem, aby bral v úvahu i anotace původního stylu. Proto jsem implementoval dva alternativní způsoby výpočtu průměrného stylu. První způsob bere v úvahu jednu vybranou třídu (např. pohlaví) a při výpočtu používá náhodné hodnoty anotací všech ostatních tříd. Druhý způsob zachovává třídy úplně a používá pouze různý šum. Použití těchto způsobů vede k lepšímu zachování zvolených tříd ve výsledném snímku, na rozdíl od původního způsobu výpočtu.



(a) Generátor (b) Diskriminátor

Obrázek 5.4: Způsob napojení embeddingů tříd na oba typy neuronových sítí.

5.3 Enkodér

Abych co nejlépe využil snadnou invertovatelnost latentního prostoru $\mathcal{W}$ ve StyleGAN2 [24] generátoru, natrénoval jsem také enkodér. Ten jsem v průběhu trénování klasifikátoru věku

využíval k dalšímu typu experimentů, kde jsem část skutečných dat během trénování nahrazoval jejich syntetizovanými projekcemi do latentního prostoru generátoru.

Jako enkodér jsem využil upravený model ResNet50 [13], který na svém vstupu dostává libovolné obrázky v rozlišení 256×256 a jeho výstupem je styl w. Tento styl je poté vstupem generátoru, pomocí kterého je syntetizována tvář, jejímž cílem je co nejvíce se podobat té zdrojové. Za pomoci metod, které byly představeny v předešlé části této práce, by bylo možné provádět sémantické úpravy s užitím tohoto stylu [36]. V této práci jsem tuto další funkcionalitu již neimplementoval.

Výpočet chyby při trénování tohoto typu sítě probíhá tak, že je vypočítána střední kvadratická chyba (zkráceně MSE) a perceptual loss (pomocí předtrénovaného VGG [37] modelu) mezi původním a generátorem syntetizovaným snímkem. Využití perceptual loss značně přispívá zachování detailů z původního snímku, jelikož porovnávané hodnoty aktivací jsou více ovlivňovány rysy ze vstupních obrázků, kdežto statistické metody jako MSE porovnávají hodnoty jednotlivých pixelů. Z toho důvodu bylo nutné vybalancovat vliv jednotlivých funkcí na velikost celkové chyby a tedy nalézt kompromis mezi zachováním dostatečného množství detailů a zachováním celkového vzhledu snímku.

5.4 Trénování neuronových sítí

Zejména četné množství experimentů s klasifikátorem věku, ale také trénování velkého modelu generativních neuronových sítí si žádalo poměrně velké výpočetní prostředky. Ačkoliv se modely EfficientNetV2 osvědčily jako velmi efektivní ve srovnání s ostatními dostupnými modely, kvůli velikosti datové sady trvaly experimenty v průměru 8–12 hodin, v závislosti na jejich nastaveních a úspěšnosti. Všechny experimenty byly ukončeny teprve poté, kdy se přestala snižovat chyba na validační datové sadě po 10 průchodů celé datové sady. Celkový počet dokončených experimentů s klasifikátorem překročil hranici tří set. V případě generativních neuronových sítí bylo provedeno 15 experimentů, týkajících se především změn architektury a trénovací datové sady. Použitím vah z již dříve natrénované sítě se potřebný čas trénování výrazně zkrátil, na pouhých několik dnů. Nejmenší část času zabralo trénování enkodérů, které byly trénovány pouze pro několik posledních verzí generátoru a jednotlivé běhy zabraly 1–2 dny.

Natrénování takového množství modelů, které jsou především v případě generativních neuronových sítí velmi náročné na grafickou paměť, by nebylo možné bez použití zdrojů, které fakulta nabízí prostřednictvím vlastního výpočetního clusteru. Zde jsem mohl pomocí systému *Sun Grid Engine* (zkráceně SGE) předávat jednotlivé úlohy, které jsou potom vykonány na jednom z desítek zařízení s výkonnými grafickými kartami, kterými fakulta disponuje. Klasifikátory bylo možné trénovat na grafických kartách NVIDIA Tesla P100, ale GAN sítě bylo nutné trénovat na výkonnějších grafických kartách NVIDIA Titan RTX a NVIDIA Quadro RTX 8000.

Kapitola 6

Vyhodnocení výsledků

V této kapitole jsou představeny dosažené výsledky jednotlivých implementovaných neuronových sítí. Nejdříve se zde zabývám oběma možnostmi generování syntetických dat, poté dosaženou přesností klasifikace pouze na reálných datech. Posléze je vyhodnocen také vliv, které přidání syntetických dat mělo na přesnost klasifikace věku na mnou vytvořené datové sadě.

6.1 Generování lidských tváří

Předtrénovaný model Anycost GAN [27], který jsem využil jako výchozí bod pro *transfer learning*, byl trénovaný na datové sadě FFHQ. Z důvodu, že tato datová sada je podmnožinou mnou vytvořené datové sady, jsem věnoval zvýšenou pozornost tomu, aby nedošlo k problému, kdy se generátor neúspěšně adaptuje na změněnou distribuci trénovací sady a nadále bude generovat takové vzorky, na kterých byl originálně natrénován. Jednou možností k vynucení přetrénování sítě na novou datovou sadu bylo inicializovat některé vrstvy diskriminátoru na náhodné hodnoty. Toto ale vedlo k destabilizaci dalšího trénování, což bylo možné pozorovat na oscilujících hodnotách FID. Alternativou by samozřejmě bylo trénovat celou síť od počátku, ale tuto možnost jsem nakonec kvůli její časové náročnosti zavrhl. Trénování těchto modelů trvá v řádu týdnů až měsíců [24, 27] a použití mojí datové sady, která disponuje větším počtem vzorků z různých distribucí by tento proces ještě pravděpodobně prodloužilo.

Nakonec se ukázalo jako funkční řešení síť nejdříve trénovat na cílové datové sadě bez FFHQ, která byla přidána později. Po několika epochách trénování bylo možné sledovat, že generátor se úspěšně adaptoval na rozdílné zarovnání obličejů, které bylo aplikováno na vzorky mojí datové sady v porovnání se zarovnáním, s nímž je datová sada FFHQ distribuována. Možné bylo také v generovaných snímcích sledovat jisté biasy, které se v nových datových sadách objevují. Příkladem můžou být khaki a béžové pozadí, které jsou symbolické pro velké množství vzorků z datové sady FRGC-IJCB. Subjektivně také došlo ke změně zastoupení jednotlivých ras v generovaných snímcích, což by odpovídalo, vzhledem k použití více vyvážených datových sad. Pozorovat bylo také možné, že ačkoliv generované snímky zůstaly stejně realistické, nejsou všechny již tak ostré jako předtím, jelikož i zdrojové snímky z přidaných datových sad často postrádají vizuální kvalitu nutnou k zachycení drobných detailů tváře. Na obrázku 6.1 je možné vidět vzorky vzorkované z náhodného šumu po dotrénování nepodmíněného modelu na všech datových sadách.

Obrázek 6.1: Snímky generované nepodmíněným generátorem trénovaným na všech datech. V prvním sloupci lze vidět způsob osvětlení fotografovaných subjektů a pozadí, které jsou typické pro datovou sadu FRGC-IJCB. Snímek vpravo dole je ukázkou vzorku vygenerovaného pravděpodobně z odlehlé hodnoty šumu.

Obrázek 6.2: Snímky generované nepodmíněným generátorem s použitím zkrácení stylu, kde průměrný styl byl získáván jako průměr 10 000 náhodných šumů a měl váhu 0,5 při váženém součtu s původním stylem.

Jak již bylo dříve zmíněno, použitím pouze jediného šumu se může stát, že z normálního rozdělení je vzorkováním získána hodnota, která je odlehlá a častěji vede k syntéze poškozených snímků (viz obrázek 6.1). Použití *zkrácení stylu*, které tomuto předchází, zvyšuje celkovou vizuální kvalitu snímků za cenu variability vzorků (viz obrázek 6.2). Tento efekt může být žádoucí pro účely vizualizace například v interaktivních ukázkách, ale pro účely rozšíření datové sady vhodný spíše není. U zkrácení stylu existují nastavitelné parametry pro počet vzorkovaných stylů, ze kterých se počítá průměr a pro podíl tohoto průměrného stylu ve váženém průměru s původním stylem, takže je možné hledat určitý kompromis mezi kvalitou a variabilitou. S těmito parametry jsem experimentoval v rámci klasifikátoru.

6.2 Podmíněné generování

U vygenerovaných obličejů, které byly k vidění výše, ovšem chyběl způsob, kterým bude možné ovlivnit výsledný vzhled. Pomocí způsobu vysvětleného v předchozí kapitole byly

Obrázek 6.3: Snímky osob obou pohlaví ve věkových kategoriích 0, 5, 10, 15 a 20 let generovaných podmíněným generátorem bez použití zkrácení stylu.

Obrázek 6.4: Snímky osob obou pohlaví ve věkových kategoriích 25, 35, 45, 55 a 70 let generovaných podmíněným generátorem bez použití zkrácení stylu.

proto do sítí přidány podmínky, které toto umožňují. Pro trénování bylo možné využít jakýchkoliv dostupných anotací jako podmínek a moje implementace umožňuje jejich snadné přidávání, ale u výsledného modelu jsem se rozhodl využít pouze anotace věku a pohlaví, které jsou z hlediska rozšíření datové sady obličejů těmi nejdůležitějšími.

Embedding vrstvy jednotlivých tříd bylo nutné inicializovat podle počtu různých hodnot, které se mohou na vstupu objevit. Pro věk jsem tedy zvolil rozsah 0–70 let, čili 71 tříd, v případě pohlaví jsou možné vstupy pouze 0 (muž) nebo 1 (žena).

V porovnání s modely starších architektur než StyleGAN2 [24], které jsem o podmíněné generování rozšiřoval v rámci bakalářské práce, bylo trénování tohoto podmíněného modelu mnohem stabilnější. Velkou výhodou je v tomto ohledu právě přidaná síť provádějící mapování $\mathcal{Z} \rightarrow \mathcal{W}$, v níž probíhá velká část učení při rozšíření modelu o třídy. Během trénování jsem využil váhy z nepodmíněných sítí trénovaných na celé datové sadě, které jsem dotrénoval s třídními informacemi.

Trénování jsem vyhodnocoval nejen pozorováním subjektivní kvality snímků, ale také pomocí dosažené Fréchet Inception Distance. Počítána byla na všech vzorcích z trénovací datové sady a stejného množství generovaných vzorků po každé epoše. Nejnižší naměřená vzdálenost byla 7,14, což je méně v porovnání s architekturami, na kterých jsem svůj

Obrázek 6.5: Snímky osob obou pohlaví ve věkových kategoriích 0, 15, 30, 50 a 70 let generovaných podmíněným generátorem s původním nepodmíněným zkrácením stylu s váhou 0,5 z 10 000 náhodných šumů. Vlivem ignorování tříd při výpočtu průměrného stylu zde dochází k změně věku generovaných osob, jak je nejvíce patrné na snímcích dětí a seniorů.

Obrázek 6.6: Snímky osob obou pohlaví ve věkových kategoriích 0, 15, 30, 50 a 70 let generovaných podmíněným generátorem s novým podmíněným zkrácením stylu s váhou 0,5 z 10 000 náhodných šumů. Mnou provedenou úpravou výpočtu průměrného stylu zde již dochází k zachování požadovaných tříd. Subjektivně tím však dochází k lehkému snížení kvality generovaných snímků.

model zakládal, kde v případě modelu Anycost GAN bylo dosaženo na datové sadě FFHQ v rozlišení 256×256 pixelů skóre FID 3,35 [27]. Důvodem je, že moje celá datová sada je v porovnání s FFHQ mnohem méně uniformní. Mnou dosažený výsledek je blíže původní architektuře s progresivním růstem, kde na FFHQ bylo dosaženo FID 7,30 [21].

Výsledky dosažené pomocí podmíněného generování jsou k vidění na obrázcích 6.3 a 6.4. Pro generování vizuálně kvalitnějších snímků na obrázcích 6.5 a 6.6 jsem využil dvě metody zkrácení stylu, původní a mnou implementovanou s úpravami pro podporu podmíněného generování, které byly popsány v předchozí kapitole. Zde byl efekt snížené variability patrný ještě více, zvlášť u kategorií méně zastoupených v trénovací datové sadě.

(a) Vzorky z validační datové sady

(b) Vzorky mimo datové sady

Obrázek 6.7: Projekce snímků do latentního prostoru generátoru pomocí enkodéru.

6.3 Augmentace reálných vzorků enkodérem

Pro podmíněný generátor použitý pro syntézu obličejů v podkapitole 6.2 jsem rovněž natrénoval enkodér. Učení probíhalo s použitím vytvořené trénovací datové sady. Výsledné projekce úspěšně napodobují vzorové snímky, ale vlivem použité chybové funkce MSE zaniká mnoho jemných detailů. Toto může například u starších osob způsobovat problém, kdy projekcí dojde ke snížení zdánlivého věku, kvůli vyhlazení vrásek. Natrénovaný model dobře generalizuje, proto funguje i na snímcích mimo trénovací datovou sadu jak ze stejné distribuce dat, tak i na *out-of-domain* snímcích, které byly získány mimo datové sady, jak je vidno na obrázku 6.7.

Během trénování klasifikátoru může být část reálných vzorků mapována do stylu w a syntetizována, a s poměrem takto augmentovaných vzorků bylo experimentováno. Zkrácení stylů při syntéze namapovaných stylů zde generátorem využíváno není.

6.4 Klasifikace věku

Před analýzou vlivu přidání syntetických dat na přesnost sítě bylo nejdříve třeba stanovit hranici, která bude posléze překonávána. Pro dosažení co nejlepšího možného výsledku jsem provedl několik experimentů za účelem hledání vhodného nastavení hyperparametrů pro modely *EfficientNetV2B0* a *EfficientNetV2B2*, jejichž výběr byl analyzovaný v předchozí kapitole.

Výběr hyperparametrů

Nastavitelné hyperparametry pro trénování na reálných datech se pojily především s optimalizátorem, kde bylo nejlepších výsledků docíleno použitím optimalizátoru *Adam*, i přes to, že k trénování modelů EfficientNet je doporučeno použít algoritmus *RMSProp* [39, 40]. Míra učení začíná na hodnotě 0,001 a na konci každé epochy trénování je její hodnota snížena o 5 %. Počet vzorků v jednotlivých minidávkách byl nastaven na 32, především kvůli tomu, aby trénování nepřinášelo paměťové problémy na žádném z dostupných výpočetních zařízení. Poměr mezi oběma chybovými funkcemi, ze kterých se skládá DLDL-v2 [9], byl nastaven na 1, tudíž *L1 loss* i *KL divergence* mají během trénování stejnou váhu.

Měření přesnosti

Vyhodnocování modelu jsem prováděl na dvou datových sadách. První z nich byla testovací část mojí datové sady, čítající 28 tisíc vzorků. Abych ověřil schopnost modelů generalizovat na out-of-domain datech, použil jsem i poskytnutou testovací datovou sadu od společnosti Innovatrics. Ta je v porovnání s mojí testovací sadou výrazně větší a obsahuje data z několika různých datových sad, které ve své práci vůbec nevyužívám, a svými distribucemi se výrazně liší od mých. Průnik těchto dvou datových sad není prázdný, ale není příliš významný, jde zhruba o 20 % dat. Model EfficientNetV2B2, který byl natrénován pouze na skutečných datech, dosáhl střední absolutní chyby **3,499 roku** na vlastní testovací sadě a **4,012 roku** na testovací sadě od Innovatrics. Menší model EfficientNetV2B0 nakonec na obou testovacích datových sadách dosáhl horších výsledků, i když na validační datové sadě dosáhl původně lehce vyšší přesnosti.

6.5 Klasifikace věku s přidanými syntetickými daty

Stanovením baseline se dostáváme do závěrečné části této práce, kterou je ověření, zda je možné dosáhnout ještě lepších výsledků použitím generativních neuronových sítí k generování unikátních dat.

Metodologie

Aby bylo trénování tak objektivní jak je možné, tak všechny hyperparametry, které jsou obecné pro všechny modely, zůstávají stejné jako v případě baseline. Výběr modelu zůstává rovněž stejný, jelikož EfficientNetV2B2 dosahoval i po přidání syntetických dat konzistentně nejlepších výsledků v porovnání s verzemi modelů s nižším i vyšším počtem parametrů. Všechny použité trénovací, validační i testovací datové sady zůstávají také nezměněné.

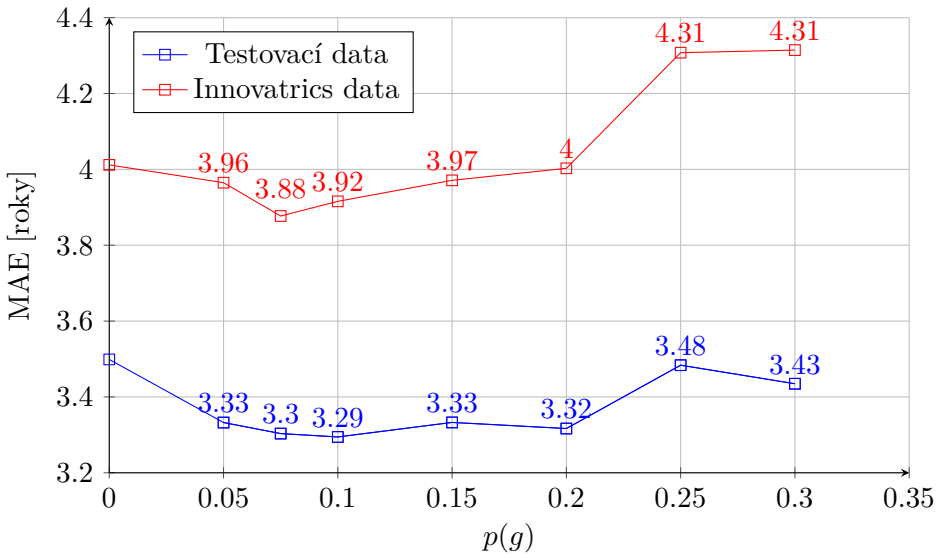
Data jsou generována dynamicky během trénování na základě potřeby, neexistuje tedy žádná vytvořená databáze takových snímků, a každá vygenerovaná tvář v průběhu trénování je tedy unikátní. Během získávání každé dávky vzorků je nejdříve na základě zvolených pravděpodobností $p(g)$, $p(e)$ určen typ dávky. S pravděpodobností $p(g)$ je typ dávky nastaven na generovanou. V tom případě jsou všechny vzorky získány pomocí náhodného vzorkování z natrénovaného generátoru. Pokud nedošlo ke zvolení generované dávky, s pravděpodobností $p(e)$ je poté nastaven typ dávky na augmentaci pomocí enkodéru. V tom případě dojde k získání dávky vzorků z datové sady, které jsou potom vstupem enkodéru, pomocí kterého jsou získány jim odpovídající styly w, pomocí kterých jsou poté syntetizovány generátorem nové vzorky, které jsou společně s původními anotacemi z datové sady následně vstupem klasifikátoru. K transformacím, jak byly popsané v kapitole 5, dochází v případě všech typů vzorků stejným způsobem. Při validaci a testování modelu nedochází k přidávání syntetických dat ani k jejich projekci do latentního prostoru generátoru.

6.5.1 Přidání pouze náhodně generovaných dat

První třídu možných experimentů tvoří pokusy, při nichž byla zvýšena pouze pravděpodobnost $p(g)$. Při generování je možné nastavovat různé implementované hyperparametry, týkající se techniky zkrácení stylu a distribuce anotací generovaných vzorků. V průběhu vyhodnocování jsem provedl mnoho experimentů s kombinacemi těchto hyperparametrů, a v rámci této sekce se na dosažené výsledky podíváme.

Experiment 1 - žádné zkrácení stylu, náhodné anotace

V této základní konfiguraci je využíváno maximální dostupné variability vzorků, kterou je generátor schopný poskytovat. Slovní spojení „náhodné anotace“ v kontextu tohoto experimentu znamená to, že anotace věku i pohlaví jsou vzorkovány náhodně z rozsahu možných hodnot, bez ohledu na jejich distribuci v trénovací datové sadě. Dosažené výsledky jsou k vidění na obrázku 6.8. V obou testovacích datových sadách se podařilo docílit několikaprocentního zlepšení oproti baseline, kde nejlepších výsledků bylo dosaženo přidáním 10% podílu generovaných dat v případě vlastní testovací sady a 7,5% podílu v případě nezávislé testovací datové sady.



Obrázek 6.8: Výsledky dosažené přidáním generovaných vzorků bez zkrácení stylu a bez zachovávání distribuce anotací.

Experiment 2 - žádné zkrácení stylu, stejná distribuce anotací

V dalším experimentu jsem nahradil náhodné anotace generovaných vzorků tak, aby výsledná distribuce anotací věku odpovídala té z trénovací datové sady. Intuice byla taková, že pokud příliš velké $p(g)$ způsobuje snížení přesnosti klasifikace, pravděpodobně kvůli velké změně distribuce trénovacích dat, tak rovnoměrná pravděpodobnost generování libovolných anotací musí tento problém ještě více prohlubovat v méně zastoupených datových sadách.

Tato změna ovšem nevedla ke zlepšení dosažené přesnosti. V případě vlastní testovací sady došlo s použitím $p(g) = 0{,}1$ ke zvýšení MAE z 3,294 roku na 3,324 roku, které bylo navíc minimem ze tří trénování modelu se stejnými nastaveními.

Experiment 3 - různé typy zkrácení stylu, náhodné anotace

Dále jsem se zabýval různými nastaveními zkrácení stylu, které jsem v dřívější části této kapitoly představoval. Kromě volby typu výpočtu jsem měnil také počet náhodných šumů, z nichž je průměr počítán. Nejdříve jsem tedy využil nejúspěšnější nastavení $p(g) = 0{,}1$ a porovnal jednotlivé typy zkrácení stylu, viz tabulka 6.1. Během experimentu byly průměrné styly počítány na základě 10 náhodných šumů. Nejlepších výsledků bylo dosaženo

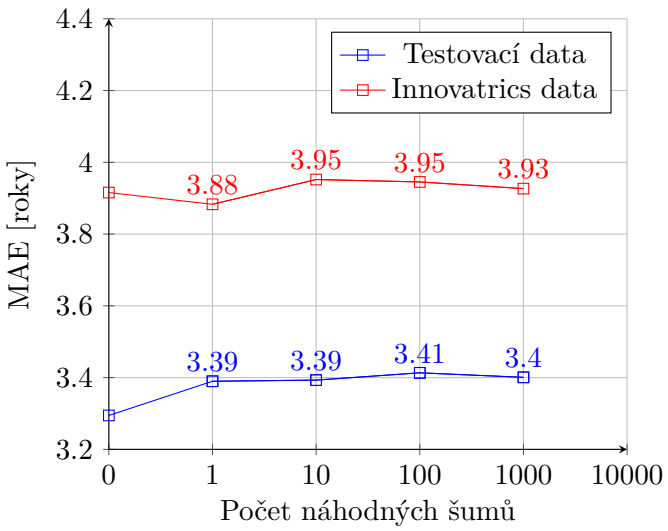
pomocí zkrácení stylu podmíněného všemi třídami. Ale stále se jedná pouze o částečný úspěch, jelikož výsledky jsou v porovnání horší, než byly před použitím zkrácení stylu.

Typ zkrácení stylu	MAE [roky]	
	Testovací data	Innovatrics data
Nepodmíněně	3,501	4,035
Podle věku	3,469	4,046
Podle pohlaví	3,453	4,061
Podle obou	3,393	3,952

Tabulka 6.1: Porovnání jednotlivých typů zkrácení stylu.

Experiment 4 - podmíněné zkrácení stylu z různého počtu náhodných šumů

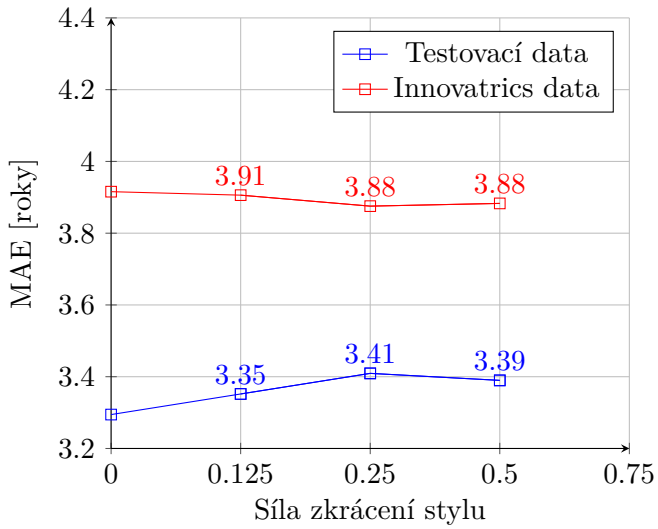
Podmíněné krácení stylu jsem tedy podrobil ještě podrobnějším experimentům, kde jsem měnil další parametr zkrácení stylu - počet průměrovaných náhodných šumů. Z výsledků na obrázku 6.9 je patrné, že vliv snížené variability vzorků nemá příliš razantní vliv na přesnost sítě. Zajímavé si je povšimnout, že výsledek na mojí vytvořené testovací datové sadě se zhoršil v porovnání s modelem s $p(g) = 0{,}1$, ale chyba klasifikací na nezávislé datové sadě se zlepšila.



Obrázek 6.9: Výsledky dosažené přidáním generovaných vzorků s podmíněným zkrácením stylu.

Experiment 5 - podmíněné krácení stylu s různou silou zkrácení

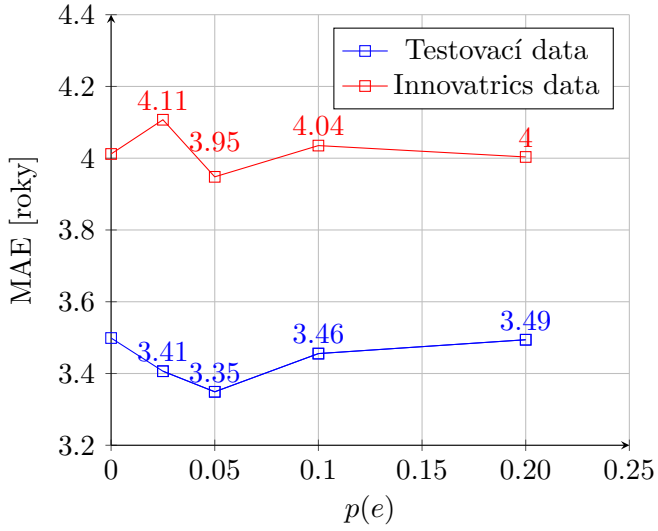
Posledním nastavitelným parametrem je *síla zkrácení*, určující poměr ve váženém součtu původního stylu a vypočítaného průměrného stylu. Dosud byla využívána v rámci všech předchozích experimentů velikost síly zkrácení 0,5. Na výsledcích z obrázku 6.10 lze vidět, že v porovnání s generováním bez zkrácení stylu může dojít k malému zlepšení generalizace, pokud takto lehce snížíme pravděpodobnost, že bude obrázek generován pouze na základě jediného stylu, který byl získaný z odlehlého šumu.



Obrázek 6.10: Vliv síly zkrácení stylu při podmíněném generování.

6.5.2 Přidání pouze enkodéru k augmentaci

Druhá třída experimentů, kterou jsem pomocí natrénovaných sítí mohl realizovat, se týkala pouze nahrazení části reálných snímků během trénování jejich projekcí do latentního prostoru generátoru, které byly vytvořeny enkodérem. Vzhledem k tomu, že jsem již neimplementoval možnost sémantických úprav takových projekcí, jediný možný experiment se týká úprav pravděpodobnosti $p(e)$. Vliv změn této proměnné je tedy možné sledovat na obrázku 6.11. Přidáním tohoto typu augmentace na 5–10 % trénovacích dat je možné sledovat mírné zlepšení generalizace oproti baseline, zejména na datové sadě pocházející ze stejné distribuce.



Obrázek 6.11: Výsledky dosažené augmentací různých částí vzorků jejich projekcí do latentního prostoru generátoru pomocí enkodéru.

6.5.3 Přidání obou typů sítí najednou

Poslední zbývající kombinací je tedy ta, při které je během trénování část vzorků z datové sady nahrazena generovanými a další část je nahrazena jejich projekcemi v latentním prostoru. Zde je možný příliš velký počet různých kombinací hyperparametrů obou sítí, proto jsem využil pouze takové hodnoty $p(g)$ a $p(e)$, které se samostatně ukázaly jako nejprospěšnější při zlepšování přesnosti klasifikace věku.

Experiment 1 - bez zkrácení stylu

Výsledky bez použití zkrácení stylu v generátoru na obou testovacích datových sadách je možné vidět v tabulce 6.2. Zde je možné si povšimnout, že v porovnání s baseline výsledkem 3,499 roku na testovacích datech došlo k poměrně výraznému zlepšení, a v případě nezávislé datové sady výsledky kromě jednoho případu zůstávají prakticky na úrovni baseline.

$p(g)$ \ $p(e)$	0,05	0,1
0,05	3,326	3,311
0,1	3,365	3,332

(a) Testovací data

$p(g)$ \ $p(e)$	0,05	0,1
0,05	4,025	4,001
0,1	4,068	3,996

(b) Innovatrics data

Tabulka 6.2: Výsledné velikosti chyby po použití obou typů sítí během trénování bez zkrácení stylu v generátoru.

Experiment 2 - se zkrácením stylu

Výsledek předchozího experimentu mě inspiroval k provedení tohoto experimentu, který probíhal podobně, s tím rozdílem, že bylo v generátoru přidání zkrácení stylu na základě jediného náhodného šumu, se silou zkrácení 0,5. Jedná se o nastavení, které se v experimentu (viz obrázek 6.9) ukázalo jako prospěšné pro zvýšení přesnosti zejména na nezávislé datové sadě s mírným zhoršením, což je opak v porovnání s předchozím experimentem. Bylo dosaženo následujících výsledků, viz obrázek 6.3. Předpoklad byl částečně správný, jelikož skutečně došlo ke zlepšení přesnosti na nezávislé datové sadě od Innovatrics a k mírnému zhoršení relativně k předchozím výsledkům na vlastní testovací sadě.

$p(g)$ \ $p(e)$	0,05	0,1
0,05	3,389	3,352
0,1	3,387	3,485

(a) Testovací data

$p(g)$ \ $p(e)$	0,05	0,1
0,05	3,899	4,028
0,1	4,011	4,036

(b) Innovatrics data

Tabulka 6.3: Výsledné velikosti chyby po použití obou typů sítí během trénování se zkrácením stylu v generátoru.

6.5.4 Vyhodnocení experimentů

Experimenty bylo úspěšně dokázáno, že pomocí různých způsobů získání syntetických dat je možné jejich přidáním k reálným datům docílit menší chyby, jinými slovy vyšší přesnosti, na různých testovacích datových sadách. Jako nejúčinnější se ukázalo použití zhruba desetiprocentního poměru náhodně generovaných obličejů. Toto nastavení dosáhlo jasně nejvyšší přesnosti klasifikace na testovací datové sadě, s průměrnou absolutní chybou 3,294 roku. Na testovací datové sadě, která mi byla poskytnuta společností Innovatrics byly výsledky méně jednoznačné, jelikož na rozdíl od vlastních testovacích dat je zde více testována schopnost generalizovat i na snímky mimo distribuci, na kterých byly modely trénovány. Ačkoliv generování bez zkrácení stylu i zde dosáhlo velmi dobrého výsledku, spolehlivě překonávající baseline, potenciál v této oblasti ukazuje i generování s velmi mírným zkrácením stylu, kterým bylo těsně dosaženo ještě nižší chyby při klasifikaci na této datové sadě, 3,875 roku.

Zlepšení nad úroveň baseline bylo docíleno i pomocí enkodéru. Jeho použití však často vedlo k méně stabilnímu trénování, což bylo během trénování možné pozorovat zejména na oscilující hodnotě chyby na validační datové sadě. Použití obou sítí najednou přineslo zlepšení především na vlastní datové sadě, ale při vyšších poměrech syntetických dat docházelo opět k oscilacím přesnosti. Jako slibné se ukázalo použití malých poměrů jako 5 % v případě obou sítí a mírné zkrácení stylu v případě generátoru. Zde již oscilace nebyly příliš znatelné a došlo ke zlepšení generalizace modelu na nezávislé testovací datové sadě. Celkově ale přidáním enkodéru nedošlo k překonání přesností, které byly dosaženy pouze pomocí generátoru.

Kapitola 7

Závěr

Cílem této práce bylo navrhnout a natrénovat podmíněnou generativní neuronovou síť, pomocí které bude možné generovat fotorealistické lidské obličeje se zvolenými vlastnostmi, jako například věk a pohlaví. Tato syntetická data poté sloužila k regularizaci a tedy ke zpřesnění odhadů konvoluční neuronové sítě provádějící klasifikaci věku, která byla v rámci této práce také implementována. Trénování těchto typů neuronových sítí probíhalo s pomocí několika datových sad tváří, které jsem po získání profiltroval, zpracoval a sjednotil do jedné velké datové sady, čítající přes 230 tisíc vzorků. V neposlední řadě byl jako doplněk generátoru založeného na architektuře StyleGAN2 [24] navíc implementován enkodér, který využívá snadné reverzibility jeho latentního prostoru. Ten byl během trénování využíván k augmentaci části vzorků datové sady. V textové části této práce jsem analyzoval podstatu dnešních nejmodernějších přístupů generativních neuronových sítí a state-of-the-art postupů v oblasti odhadování věku, které jsem v obou případech ve své implementaci využil.

Pomocí vytvořené datové sady jsem natrénoval konvoluční neuronovou síť, na které jsem vyhodnotil přesnost odhadování věku na vlastní testovací podmnožině datové sady a na nezávislé testovací sadě, která mi byla poskytnuta společností Innovatrics. Dosáhl jsem na nich středních absolutních chyb 3,499 roku a 4,012 roku, které poté sloužily pro vyhodnocení vlivu přidání syntetických dat během trénování. Použitím natrénovaných modelů zbylých dvou typů neuronových sítí jsem provedl velké množství experimentů s různými nastaveními hyperparametrů. Na obou testovacích datových sadách se mi podařilo zlepšit dosažené výsledky pomocí jednotlivých modelů i jejich kombinace. Nejlepších výsledků bylo dosaženo užitím zhruba desetiprocentního podílu generátorem syntetizovaných tváří. Nejnižší dosažené střední absolutní chyby na datových sadách byly 3,294 roku a 3,875 roku.

I přes velké množství více než tří set experimentů, které byly v průběhu prací s klasifikátorem provedeny, se stále nabízí další možné cesty, kterými je možné se při dalším zkoumání vydat. V případě generátoru by mohlo být prozkoumáno více nastavení týkajících se krácení stylu generovaných vzorků, které během vyhodnocování ukázalo potenciál. V případě enkodéru, který byl v této práci využit pouze okrajově, by další vývoj mohl odvíjet směrem natrénování složitějšího modelu [27] či výpočtu a aplikování vektorů vedoucím k sémantickým úpravám vzorku [36]. Největší potenciál ale v této metodě vidím ale pro její užití na úlohách s více limitovanou velikostí datových sad, například v jiných oblastech biometrie. Právě variabilita datové sady a výrazná nejednoznačnost anotací věku se zde ukázaly jako značně náročné a možné pokračování práce by jistě znamenalo mnohem přísnější filtrování jak na základě kvality vzorků, tak anotací věku.

Literatura

[1] Agustsson, E., Timofte, R., Escalera, S., Baro, X., Guyon, I. et al. Apparent and Real Age Estimation in Still Images with Deep Residual Regressors on Appa-Real Database. In: *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*. 2017, s. 87–94. DOI: 10.1109/FG.2017.20.

[2] Back, J. Fine-Tuning StyleGAN2 For Cartoon Face Generation. *ArXiv e-prints*. Červen 2021, s. arXiv:2106.12445.

[3] Beckett, J. *What's a Generative Adversarial Network? Inventor Explains* [online]. Květen 2017 [cit. 2022-01-09]. Dostupné z: https://blogs.nvidia.com/blog/2017/05/17/generative-adversarial-networks/.

[4] Bond, S. *That smiling LinkedIn profile face might be a computer-generated fake* [online]. Březen 2022 [cit. 2020-04-30]. Dostupné z: https://www.npr.org/2022/03/27/1088140809/fake-linkedin-profiles.

[5] Chen, S., Zhang, C., Dong, M., Le, J. a Rao, M. Using Ranking-CNN for Age Estimation. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Červenec 2017. DOI: 10.1109/CVPR.2017.86.

[6] Cheng, J., Liu, Z., Guan, H., Wu, Z., Zhu, H. et al. Brain Age Estimation From MRI Using Cascade Networks With Ranking Loss. *IEEE Transactions on Medical Imaging*. 2021, sv. 40, č. 12, s. 3400–3412. DOI: 10.1109/TMI.2021.3085948.

[7] Elfwing, S., Uchibe, E. a Doya, K. Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning. *Neural networks : the official journal of the International Neural Network Society*. 2018, sv. 107, s. 3–11. DOI: 10.1016/j.neunet.2017.12.012.

[8] Escalera, S., Fabian, J., Pardo, P., Baró, X., Gonzàlez, J. et al. ChaLearn Looking at People 2015: Apparent Age and Cultural Event Recognition Datasets and Results. In: *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*. Prosinec 2015, s. 243–251. DOI: 10.1109/ICCVW.2015.40.

[9] Gao, B.-B., Liu, X.-X., Zhou, H.-Y., Wu, J. a Geng, X. Learning Expectation of Label Distribution for Facial Age and Attractiveness Estimation. *ArXiv e-prints*. Červenec 2020, s. arXiv:2007.01771.

[10] Gao, B.-B., Xing, C., Xie, C.-W., Wu, J. a Geng, X. Deep Label Distribution Learning With Label Ambiguity. *IEEE Transactions on Image Processing*. Duben 2017, sv. 26, s. 2825–2838. DOI: 10.1109/TIP.2017.2689998.

[11] Goodfellow, I., Pouget Abadie, J., Mirza, M., Xu, B., Warde Farley, D. et al. Generative Adversarial Nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D. a Weinberger, K. Q., ed. *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014, s. 2672–2680.

[12] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. a Courville, A. C. Improved Training of Wasserstein GANs. Curran Associates, Inc. 2017, sv. 30. Dostupné z: https://proceedings.neurips.cc/paper/2017/file/892c3b1c6dccd52936e27cbd0ff683d6-Paper.pdf.

[13] He, K., Zhang, X., Ren, S. a Sun, J. Deep Residual Learning for Image Recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Červen 2016. DOI: 10.1109/CVPR.2016.90.

[14] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B. a Hochreiter, S. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In: Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R. et al., ed. *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017, sv. 30. Dostupné z: https://proceedings.neurips.cc/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf.

[15] Huang, Y., Cheng, Y., Bapna, A., Firat, O., Chen, D. et al. GPipe: Efficient Training of Giant Neural Networks using Pipeline Parallelism. Curran Associates, Inc. 2019, sv. 32. Dostupné z: https://proceedings.neurips.cc/paper/2019/file/093f65e080a295f8076b1c5722a46aa2-Paper.pdf.

[16] Ioffe, S. a Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: Bach, F. a Blei, D., ed. *Proceedings of the 32nd International Conference on Machine Learning*. Lille, France: PMLR, 07.–09. červenec 2015, sv. 37, s. 448–456. Proceedings of Machine Learning Research. Dostupné z: https://proceedings.mlr.press/v37/ioffe15.html.

[17] Isola, P., Zhu, J.-Y., Zhou, T. a Efros, A. A. Image-To-Image Translation With Conditional Adversarial Networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Červenec 2017. DOI: 10.1109/ICCV.2017.244.

[18] Jo, Y. a Park, J. SC-FEGAN: Face Editing Generative Adversarial Network With User's Sketch and Color. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Říjen 2019, s. 1745–1753. DOI: 10.1109/ICCV.2019.00183.

[19] Karkkainen, K. a Joo, J. FairFace: Face Attribute Dataset for Balanced Race, Gender, and Age for Bias Measurement and Mitigation. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2021, s. 1548–1558. DOI: 10.1109/WACV48630.2021.00159.

[20] Karnewar, A. a Wang, O. MSG-GAN: Multi-Scale Gradients for Generative Adversarial Networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Červen 2020. DOI: 10.1109/cvpr42600.2020.00782.

[21] Karras, T., Aila, T., Laine, S. a Lehtinen, J. Progressive Growing of GANs for Improved Quality, Stability, and Variation. *ArXiv e-prints*. Říjen 2017, s. arXiv:1710.10196.

[22] Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J. et al. Alias-Free Generative Adversarial Networks. In: *Proc. NeurIPS*. 2021.

[23] Karras, T., Laine, S. a Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Červen 2019. DOI: 10.1109/CVPR.2019.00453.

[24] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J. et al. Analyzing and Improving the Image Quality of StyleGAN. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Červen 2020. DOI: 10.1109/CVPR42600.2020.00813.

[25] Kingma, D. P. a Ba, J. Adam: A Method for Stochastic Optimization. *ArXiv e-prints*. Prosinec 2014, s. arXiv:1412.6980.

[26] Krizhevsky, A., Sutskever, I. a Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. In: Pereira, F., Burges, C., Bottou, L. a Weinberger, K., ed. *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2012, sv. 25. DOI: 10.1145/3065386. Dostupné z: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[27] Lin, J., Zhang, R., Ganz, F., Han, S. a Zhu, J.-Y. Anycost GANs for Interactive Image Synthesis and Editing. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021. DOI: 10.1109/CVPR46437.2021.01474.

[28] Mirza, M. a Osindero, S. Conditional Generative Adversarial Nets. *ArXiv e-prints*. Listopad 2014, s. arXiv:1411.1784.

[29] Nightingale, S. J. a Farid, H. AI-synthesized faces are indistinguishable from real faces and more trustworthy. *Proceedings of the National Academy of Sciences*. 2022, sv. 119, č. 8, s. e2120481119. DOI: 10.1073/pnas.2120481119.

[30] Radford, A., Metz, L. a Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *ArXiv e-prints*. Listopad 2015, s. arXiv:1511.06434.

[31] Ramachandran, P., Zoph, B. a Le, Q. V. Searching for Activation Functions. *ArXiv e-prints*. Říjen 2017, s. arXiv:1710.05941.

[32] Ricanek, K. a Tesafaye, T. MORPH: a longitudinal image database of normal adult age-progression. In: *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*. 2006, s. 341–345. DOI: 10.1109/FGR.2006.78.

[33] Rothe, R., Timofte, R. a Gool, L. V. DEX: Deep EXpectation of Apparent Age from a Single Image. *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*. 2015, s. 252–257. DOI: 10.1109/ICCVW.2015.41.

[34] Sauer, A., Schwarz, K. a Geiger, A. StyleGAN-XL: Scaling StyleGAN to Large Diverse Datasets. *ArXiv e-prints*. Únor 2022, s. arXiv:2202.00273.

[35] Schmidhuber, J. *Making the World Differentiable: On Using Self-Supervised Fully Recurrent Neural Networks for Dynamic Reinforcement Learning and Planning in Non-Stationary Environments*. 1990.

[36] Shen, Y., Gu, J., Tang, X. a Zhou, B. Interpreting the Latent Space of GANs for Semantic Face Editing. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Červen 2020. DOI: 10.1109/CVPR42600.2020.00926.

[37] Simonyan, K. a Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *ArXiv e-prints*. Září 2014, s. arXiv:1409.1556.

[38] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. a Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*. 2014, sv. 15, č. 56, s. 1929–1958.

[39] Tan, M. a Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In: Chaudhuri, K. a Salakhutdinov, R., ed. *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 09.–15. červen 2019, sv. 97, s. 6105–6114. Proceedings of Machine Learning Research. Dostupné z: https://proceedings.mlr.press/v97/tan19a.html.

[40] Tan, M. a Le, Q. EfficientNetV2: Smaller Models and Faster Training. In: Meila, M. a Zhang, T., ed. *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 18.–24. červenec 2021, sv. 139, s. 10096–10106. Proceedings of Machine Learning Research. Dostupné z: https://proceedings.mlr.press/v139/tan21a.html.

[41] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L. et al. Attention is All you Need. In: Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R. et al., ed. *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017, sv. 30.

[42] Yin, Y., Jiang, S., Robinson, J. P. a Fu, Y. Dual-Attention GAN for Large-Pose Face Frontalization. In: *2020 15th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2020)*. 2020, s. 249–256. DOI: 10.1109/FG47880.2020.00004.

[43] Yu, N., Skripniuk, V., Abdelnabi, S. a Fritz, M. Artificial Fingerprinting for Generative Models: Rooting Deepfake Attribution in Training Data. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Říjen 2021, s. 14448–14457. DOI: 10.1109/ICCV48922.2021.01418.

[44] Zazo, R., Sankar Nidadavolu, P., Chen, N., Gonzalez Rodriguez, J. a Dehak, N. Age Estimation in Short Speech Utterances Based on LSTM Recurrent Neural Networks. *IEEE Access*. 2018, sv. 6, s. 22524–22530. DOI: 10.1109/ACCESS.2018.2816163.

[45] Zhang, K., Zhang, Z., Li, Z. a Qiao, Y. Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. *IEEE Signal Processing Letters*. Říjen 2016, sv. 23, č. 10, s. 1499–1503. DOI: 10.1109/LSP.2016.2603342.

[46] Zhang, Y. a Davison, B. D. Adversarial Regression Learning for Bone Age Estimation. In: Feragen, A., Sommer, S., Schnabel, J. a Nielsen, M., ed. *Information Processing in Medical Imaging*. Cham: Springer International Publishing, 2021, s. 742–754. ISBN 978-3-030-78191-0.

[47] Zhu, J.-Y., Park, T., Isola, P. a Efros, A. A. Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Říjen 2017.