

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Vid Potočnik

**Generiranje slik staršev na podlagi
slik otrok**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Luka Šajn

Ljubljana, 2023

To delo je ponujeno pod licenco *Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 2.5 Slovenija* (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavnine dela kot tudi rezultati diplomskega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu, lahko distribuira predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani creativecommons.si ali na Inštitutu za intelektualno lastnino, Strelška 1, 1000 Ljubljana.



Izvorna koda diplomskega dela, njeni rezultati in v ta namen razvita programska oprema je ponujena pod licenco GNU General Public License, različica 3 (ali novejša). To pomeni, da se lahko prosto distribuira in/ali predeluje pod njenimi pogoji. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses/>.

Besedilo je oblikovano z urejevalnikom besedil L^AT_EX.

Kandidat: Vid Potočnik

Naslov: Generiranje slik otrok na podlagi slik staršev - analiza podobnosti sorojencev

Vrsta naloge: Diplomska naloga na univerzitetnem programu prve stopnje Računalništvo in informatika

Mentor: doc. dr. Luka Šajn

Opis:

Title: Generation of parent images based on images of their children

Description:

Kazalo

Povzetek

Abstract

1	Uvod	1
1.1	Motivacija	1
1.2	Opis problema	2
1.3	Struktura naloge	3
2	Pregled področja	5
2.1	Znane metode	5
3	Metodologija	7
3.1	Generativne nasprotniške mreže	7
3.2	StyleGAN	8
3.3	Napovedni modeli	16
4	Implementacija lastne metode	23
4.1	Struktura projekta	23
4.2	Opis uporabljenih tehnologij in metod	27
4.3	Skupna implementacija	27
4.4	Lastna implementacija	30
5	Ovrednotenje rezultatov	35
5.1	Deepface knjižnica za razpoznavo obrazov	35

5.2	Ovrednotenje	36
5.3	Vizualizacija	37
6	Sklepne ugotovitve	39
6.1	Zaključek	39
6.2	Primerjava znanih metod z lastno implementacijo	39
6.3	Možne izboljšave in dopolnitve	39
	Literatura	41

Seznam uporabljenih kratic

kratica	angleško	slovensko
CA	classification accuracy	klasifikacijska točnost
SVM	support vector machine	metoda podpornih vektorjev
GAN	generative adversarial ne-twork	generativno nasprotniška mreža
LR	linear regression	linearna regresija

Povzetek

Naslov: Generiranje slik staršev na podlagi slik otrok

Avtor: Vid Potočnik

V tem diplomskem delu bom najprej opisal problem generiranja slik otroških obrazov na podlagi slik staršev in na kratko predstavil že razvite metode ChildGAN, KinshipGAN in DNA-Net. Rezultate, ki jih te metode generirajo, bom poskušal primerjati in ovrednotiti. V drugem delu diplome se bom osredotočil na implementacijo lastne metode za vizualizacijo spremembe spola pri generiranju slik otroških obrazov s pomočjo StyleGAN2 tehnologije.

Ključne besede: GAN, StyleGAN, PyTorch, ChildGAN, KinshipGAN, DNA-Net, Families In The Wild, linearna regresija, napovedni modeli, genetika.

Abstract

Title: Generation of parent images based on images of their children

Author: Vid Potočnik

In this thesis, I will first describe the problem of generating images of children's faces from their parents' images and briefly introduce the already known methods ChildGAN, KinshipGAN and DNA-Net. I will try to compare and evaluate the results of these methods. In the second part of the thesis, I will focus on the implementation of my own method for visualising gender changes when generating images of children's faces using StyleGAN2 technology.

Keywords: GAN, StyleGAN, PyTorch, ChildGAN, KinshipGAN, DNA-Net, Families In The Wild, linear regression, prediction models, genetics.

Poglavlje 1

Uvod

Sinteza človeških slik je kategorija sinteze slik, kjer poskušamo ustvariti verodostojne ali celo fotorealistične upodobitve človeških podob. Računalniško upodobitev človeške podobe je možno izvesti na več različnih načinov in v različnih formatih. V moji diplomski nalogi se bom osredotočil na slikovni format prikaza podobe, natančneje obraza. Obraze bom generiral z algoritmimi in metodami umetne inteligence, glavna med njimi bo GAN (angl. Generative Adversarial Network).

Natančneje, v diplomski nalogi poskusimo zgraditi najboljši napovedni model, ki bi na podlagi obraza otroka generiral čim boljša približka obrazov mame in očeta. Diplomsko naložbo s podobno tematiko je že naredil Nejc Šuklje [20], vendar se je on ukvarjal z generiranjem slik otrok na podlagi slik staršev, ravno obratno od moje teme. Oba sva uporabila predhodno nedokončano delo Matjaža Mava, ki je bila dobra podlaga za nadgradnjo in dopolnitev celotne metode.

1.1 Motivacija

Zakaj ljudje živimo? Kaj je cilj vsakega posameznika v življenju? V kaj vložimo veliko truda tekom življenja? Za veliko ljudi je odgovor na ta vprašanja v reprodukciji oziroma otrocih. Verjetno veliko ljudi, ki ima na-

men vzgojiti otroka, zanima kakšne fizične atribute bodo podedovali in od katerega starša bodo.

Prav zaradi tega nakaj dobrih implementacij na tematiko generiranja slik obrazov otrok na podlagi slik staršev že obstaja. Najboljše izmed njih so ChildGAN[3], KinshipGAN[16] in DNA-Net[6], v prihodnjem poglavju jih bom podrobneje predstavil.

Nisem pa našel, do začetka pisanja diplomske naloge, nobene implementacije generiranja slik obrazov staršev na podlagi slik otrok, kar je bila dobra motivacija za izbiro teme in implementacijo metode.

1.2 Opis problema

Človeški možgani lahko s pomočjo analize razlikovalnih vzorcev delov obraza prepozna sorodstvene vezi na fotografijah, kar je dokaz, kako izjemno kompleksni so. Natančne in uporabne metode za preverjanje sorodstva s pomočjo računalniške analize [14, 21, 18] obstajajo že nekaj let, saj so modeli na osnovi učenja globokih mrež pokazali impresivne sposobnosti za avtomatsko ekstrakcijo vzorcev sorodnosti iz obrazov. Te metode prekašajo sposobnosti človeškega zaznavanja pri različnih problemih sorodstvene verifikacije in identifikacije. Verifikacija je problem, predstavljen s 1-1 preslikavo danega obraza proti znani identiteti. (Ali je to ta oseba?). Identifikacija pa je problem, predstavljen s 1-n preslikavo danega obraza proti dani bazi slik obrazov (Kdo je ta oseba?). Pri ocenjevanju lastne metode bomo probali rešiti problem verifikacije. Določenemu otroku bomo poskušali generirati čim bolj skladna starša, njuni identiteti pa bomo nato verificirali z resničnimi slikami staršev tega otroka. Zato bom pri ovrednotenju namesto metod za preverjanje sorodstva opisanih zgoraj, uporabljal bolj splošne metode za identifikacijo oziroma verifikacijo obrazov kot so FaceNet [19], ArcFace [4], VGG-Face [**Parkhi15**]

1.3 Struktura naloge

Diplomska naloga je sestavljena iz šestih poglavij,

Uvod opiše osnoven problem in poda motivacijo za delo. V drugem poglavju so opisane že znane sorodne metode, ki so zgrajene na primerljivi metodologiji. Specifična metodologija lastne in skupne implementacije je opisana v tretjem poglavju. Četrto poglavje predstavi celotno implementacijo metode. Opiše strukturo zbirke slikovnih podatkov, predstavi skupni del, ki ga je pripravil Matjaž Mav in na koncu še lastno implementacijo metode za generiranje slik staršev na podlagi slik otrok. V petem poglavju so rezultati, ki sem jih generiral z metodo ovrednoteni in vizualizirani. Zadnje, šesto poglavje, pa zaključi celotno diplomsko delo in poda sklepne ugotovitve ter mogoče izboljšave in dopolnitve.

Poglavlje 2

Pregled področja

2.1 Znane metode

Metode za napovedovanje in generiranje obrazov staršev iz slik njihovih otrok, ob začetku pisanja te diplomske naloge, še ni bilo, zato je bilo vrednotenje oziroma primerjava rezultatov otežena. Obstajaja pa nekaj metod, s pomočjo katerih lahko napovedujemo obraze otrok iz slik njihovih staršev.

Neposredna primerjava metod enega in drugega problema ni najbolj smiselna, saj je problem drugačen, kljub temu, da ga lahko rešimo s podobnimi metodologijami. Predstavil bom tri metode za generiranje slik otrok na podlagi slik staršev.

2.1.1 KinshipGAN

KinshipGAN [16] je najstarejša izmed treh metod opisanih tukaj, zato tudi najbolj zastarella. Pričakovano generira slike z slabšo kvaliteto od drugih dveh metod. Metoda slik ne pretvarja v latentni prostor s pomočjo nvidijske StyleGAN [11] tehnologije, vendar uporablja svoj GAN, ki je bil dobra podlaga za začetek raziskovanja o tej tehnologiji.

2.1.2 ChildGAN

Publikacija z originalnim naslovom **Heredity-aware Child Face Image Generation with Latent Space Disentanglement** [3] predstavi nov način generiranja obraznih struktur imenovan **ChildGAN**. Temelji na tehnologiji razviti pri podjetju Nvidia imenovani StyleGAN. Osnovna ideja je razplet latentnega prostora že prednaučenega generativnega modela. To omogoči natančno semantično določanje posameznih obraznih atributov in struktur. Razdalje med obraznimi atributi so uporabljeni kot psevdo-oznake za določanje najbolj pomembnih atributov (predstavljenih s semantičnimi vektorji). Avtorji s pomočjo genetskih zakonitosti združijo semantične vektorje staršev v združen latentni vektor. Ta metoda naj bi delovala bolje od drugih že znanih, ki so opisane kasneje in velja za state-of-the-art.

2.1.3 DNA-Net

Avtorji publikacije z originalnim naslovom **What will your child look like? dna-net: Age and gender aware kin face synthesizer** [6] v njej predlagajo dvo-stopenjski generativni model, ki na podlagi slik staršev skuša napovedati obraz otroka. Prva stopnja modela je generativni nasprotniški avtomatski kodirnik, pogojen s starostjo in spolom, ki mapira izgled obraza na bolj specifične obrazne attribute. Druga stopnja modela je tako imenovani DNA-Net, ki je uporabljen kot transformacija iz atributov, dobljenih iz obrazov staršev, v attribute obraza otroka, na podlagi genetskih zakonov. Omogočeno je tudi spremenjanje starosti in spola generiranih slik otrok.

Poglavlje 3

Metodologija

Metoda temelji na tehnologiji dveh konvolucijskih nevronskeih mrež, ki skupaj tvorita generativno nasprotniško mrežo (GAN), bolj specifično nvidijin StyleGAN2. S katero lahko posamezno sliko obraza predstavimo v latentnem prostoru, povedano enostavnejše, z vektorjem nižjih dimenzij od prvotne slike. Predstavitev slike v latentnem prostoru ohrani večino informacije o njej, zato lahko z obratnim procesom sliko generiramo iz predstavitve v latentnem prostoru. Latentni prostor omogoča veliko hitrejše in bolj efektivno računanje, zato lahko napovedni model zgradimo na več podatkih oziroma slikah.

3.1 Generativne nasprotniške mreže

StyleGAN je v sami osnovi generativna nasprotniška mreža (ang. General Adversarial Network - GAN) [9]. Klasicna arhitektura GAN je sestavljena iz dveh nasprotujičih konvolucijskih nevronskeih mrez (ang. Convolutional Neural Network - CNN) [**<empty citation>**] generatorja in diskriminatorja. Generator poskuša generirati umetne primere (največkrat so to slike) čim bolj podobne resničnim primerom, diskriminator pa poskuša ločiti resnične od umetno generiranih primerov. S takšnim tekmovanjem skozi učenje na veliko učnih ciklih in primerih se oba izboljšujeta. dokler ni generator zmožen generirati tako dobrih primerov, da jih skoraj ni mogoče več ločiti od re-

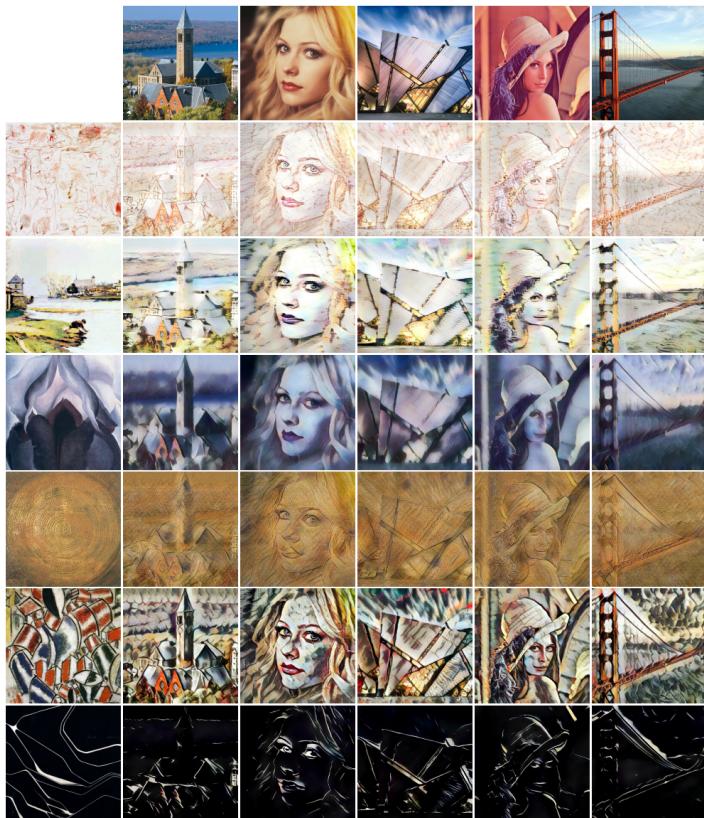
sničnih.

$$\min_G \max_D \mathcal{V}(G, D) = \mathbb{E} \mathbf{x} \sim p_{data}(\mathbf{x}) [\log D(\mathbf{x})] + \mathbb{E} \mathbf{z} \sim p_z(\mathbf{z}) [\log(1 - D(G(\mathbf{z})))] , \quad (3.1)$$

Enačba 3.1 prestavlja izgubo nasprotniških strani (ang. adversarial loss). Cilj je minimiziranje razlik med distribucijama resničnih in umetnih podatkov. Generator poskuša izgubo minimizirati, diskriminatore pa maksimizirati. Generator na vhod dobi šum z in generira umetno sliko $G(z)$. Diskriminator nato dobi na vhod realno sliko x in umetno generirano sliko $G(z)$. Med učenjem pokuša diskriminatore čim bolj ločiti med realnimi x in umetnimi $G(z)$ primeri tako, da za umetni primer vrne verjetnost, ki je blizu 0, za realni primer pa verjetnost bližje 1. V idealnem primeru za diskriminatore bi bila vrednost funkcije 1. Ravno obratno počne generator, ki poskuša minimizirati vrednost funkcije, in s tem čim bolje naučiti porazdelitev slik v učni množici, tako pa prevarati diskriminatore.

3.2 StyleGAN

Od klasičnega GANa, se StyleGAN razlikuje samo v generatorju. Karras et al. so v članku **A Style-Based Generator Architecture for Generative Adversarial Networks** [11] predstavili arhitekturo spremenjenega generatorja, ki temelji na podobnem principu generatorja predstavljenega zgoraj, vendar s spremenjeno in dopolnjeno arhitekturo. Običajni konvolucijski nevronski mreži generatorja, je bil dodan koncept iz že znane tehnologije prenosa stila med fotografijami **AdaIN** predstavljeni v **Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization** [10].

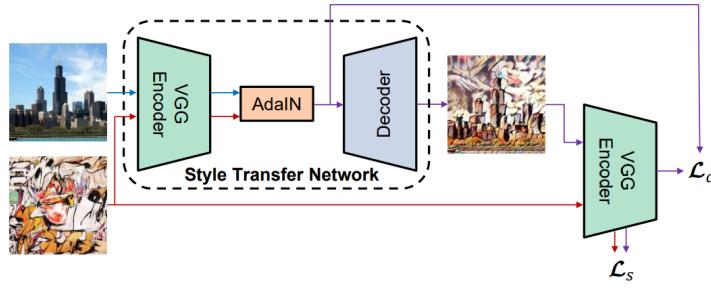


Slika 3.1: Primeri AdaIN slogovnega prenosa na slike. Vsaka vrstica si deli enak stil, vsak stolpec pa enako sliko vsebine

3.2.1 AdaIN

AdaIN (ang. Adaptive Instance Normalization) [10] je normalizacijska tehnika, uporabljena za prenos poljubnega sloga med slikami v realnem času. Algoritem sestavlja kodirnik, ki sliko z vsebino in sliko sloga preslika v latentni prostor značilk in vrne značilki x in y . Sliki predstavljeni v latentnem prostoru sta nato poslani v sloj AdaIN, kateri v vsakem kanalu slik uravna srednjo vrednost μ in standardni odklon σ značilk x slike z vsebino s srednjo vrednostjo in standardnim odklonom značilk y slike s sloganom. Na koncu naučeno dekodirno omrežje generira novo slogovno spremenjeno sliko z invertiranjem izhoda AdaIN nazaj v prostor originalne slike z vsebino.

$$\text{AdaIN}(x, y) = \sigma(y) \frac{x - \mu(x)}{\sigma(x)} + \mu(y), \quad (3.2)$$



Slika 3.2: Arhitektura AdaIN

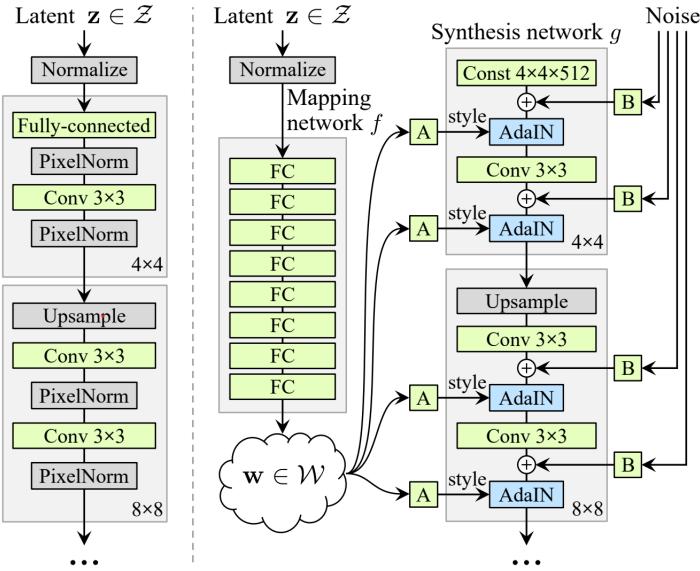
3.2.2 Arhitektura generatorja

AdaIN je osnovna oziroma glavna sprememba dodana v novo arhitekturo StyleGAN generatorja. V arhitekturi StyleGAN generatorja sta na vsakem nivoju vsebovana po dva AdaIN sloja. Arhitektura generatorja pa je okoli teh slojev primerno prilagojena, kar bom opisal v tem poglavju.

Klasičen GAN generator je večplastni perceptron (ang. multilayer perceptron) oziroma nevronška mreža, pri generiranju slik je to konvolucijska nevronška mreža [13]. Generator na vhod prejme naključni šum (ang. random noise), njegov izhod pa je generirana slika, ki je naprej podana diskriminatorju. Podobno velja za StyleGAN generator, le z določenimi dodatki in spremembami. Spremembe bom opisal v podpoglavljih Vhodni sloj, Mapirno omrežje in dodajanje sloga (ang. mapping network) ter Dodajanje šuma sinteznemu omrežju.

Generiranje slik poteka v sinteznem omrežju g , ki ga sestavlja 18 konvolucijskih slojev, ki so v 9 resolucijskih nivojih - po dve konvoluciji na enem resolucijskem nivoju. Generator omogoča povečanje dimenzije vhodne konstante iz 4x4 na 1024x1024, kar na koncu predstavlja generirano sliko. K

sinteznemu omrežju so bile v okviru StyleGAN-a dodane izboljšave, ki predstavljajo nekakšno podporno infrastrukturo generatorju.



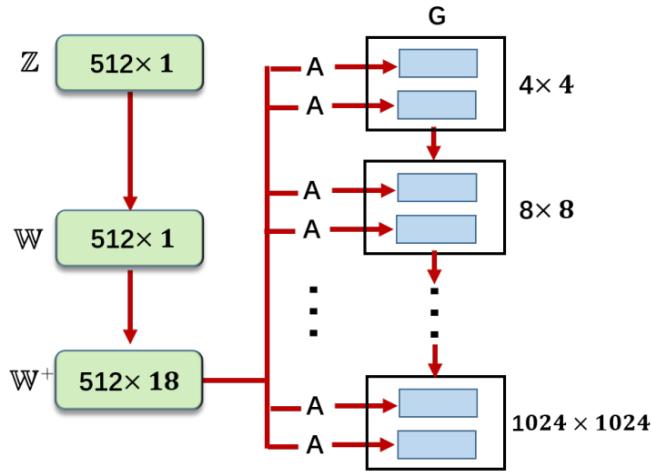
Slika 3.3: Arhitektura StyleGAN generatorja [11]

Vhodni sloj

StyleGAN generator se od klasičnega razlikuje že pri vhodnih podatkih. Prvi sloj pri klasičnem generatorju je tako imenovani vhodni sloj (ang. input layer), ki naključni šum na vhodu pretvori v uporabno obliko za nadaljnje sloje nevronске mreže. StyleGAN pa izpusti celoten naključen vhodni sloj, namesto tega pa je vhod v NN naučena konstanta določene dimenzije 4x4x512. Konstanta na vhodnem sloju pomeni, da dejanski vhodni podatki v model pridejo le preko slogov, dodanih v AdaIN operator na vsakem sloju NN, ki so odvisni od mapirnega omrežja f.

Mapirno omrežje in dodajanje sloga

Mapirno omrežje je arhitekturno in vsebinsko po pomembnosti največja sprememba od začetnega Progressive GAN-a [13]. V osnovi je to 8-plastni per-



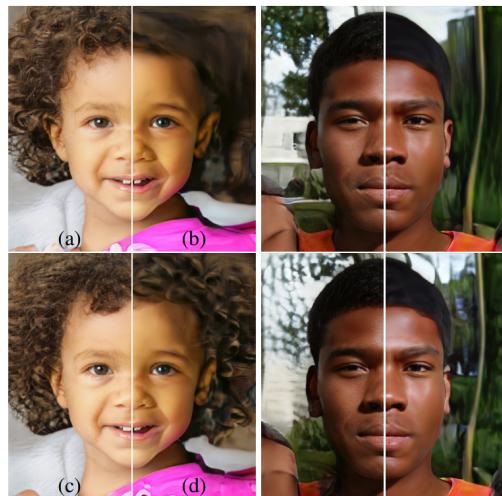
Slika 3.4: Mapirno omrežje f , ki mapira latentno kodo iz prostora Z v prostor W . W^+ je razširjen prostor W . A je afina transformacija, specializirana za vsak konvolucijski sloj sinteznega omrežja g

ceptron (ang. multilayer perceptron - MLP), drugače povedano nevronska mreža iz osmih popolno povezanih slojev. Na vhod dobi latentno kodo z dimenzije 512, ki pripada latentnemu prostoru Z . Naloga omrežja je mapiranje z v latentni prostor W . Izhod omrežja je latentna koda w enakih dimezij kot $z - 512$. Prednaučena afina transformacija A nato, za vsak izmed 18 slojev konvolucije posebej, specializira w v stile y , ki kontrolirajo AdaIN operacije po vsakem konvolucijskem slogu sinteznega omrežja g . AdaIN normalizacija deluje enako kot je bila opisana v poglavju 3.2.1, z razliko v vhodnem slogu. Vhodni slog v tem primeru ni slika sloga, vendar je to izračunan slog y na podlagi vektorja w . AdaIN operacija je torej definirana kot

$$\text{AdaIN}(x_i, y) = y_{s,i} \frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i}, \quad (3.3)$$

kjer je vsak feature map x_i obravnavan posebej. Torej srednja vrednost in standardni odklon x_i sta uravnana s srednjo vrednostjo in standardnim odklonom pripadajoče komponente sloga y .

Dodajanje šuma sinteznemu omrežju



Slika 3.5: (a) Šum dodan na vseh slojih. (b) Brez dodanega šuma. (c) Šum dodan v slojih podrobnosti (fine layers) ($64^2 - 1024^2$). (d) Šum dodan v slojih (coarse layers) ($4^2 - 32^2$)

Najbolj vidni aspekti stohastičnih podrobnosti v človeških portretih bi na primer bili točna postavitev las, lokacija kožnih por in peg, postavitev gub na obrazu in podobno. Za naponredno dodajanje stohastičnih podrobnosti se v model dodaja šum (ang. noise). Šum je eno kanalna slika nepovezanega Gaussovega šuma. Dodana je po vsakem konvolucijskem sloju sinteznega omrežja g . Katerkoli od aspektov podrobnosti je z dodajanjem šuma randomiziran brez, da to vpliva na celotno podobo obraza in identiteto človeka.

3.2.3 StyleGAN2

Kreatorji StyleGAN arhitektue so 2020 to posodobili in izdali članek **Analyzing and Improving the Image Quality of StyleGAN** [12]. V članku analizirajo več njegovih značilnih artefaktov in predlagajo spremembe v arhitekturi modela in metodah učenja modela, da bi jih odpravili. Večje spremembe so na novo oblikovana normalizacija generatorja (operacija AdaIN),



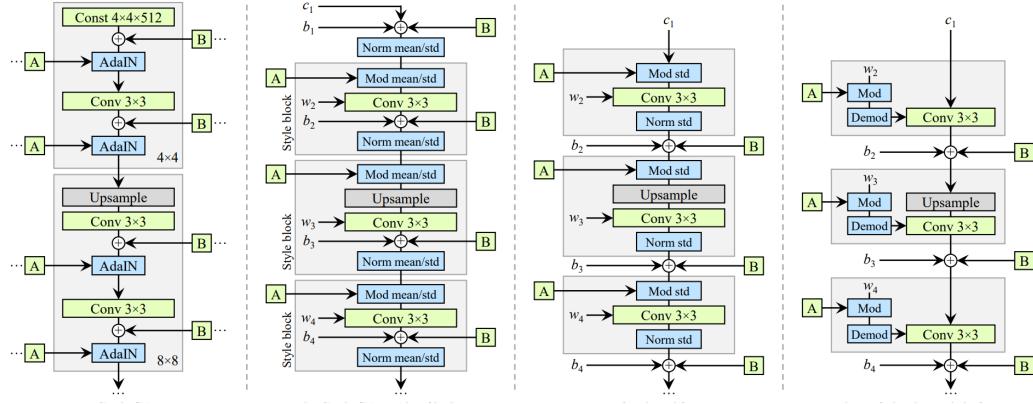
Slika 3.6: Normalizacija v operaciji AdaIN StyleGAN generatorja povzroča artefakte v obliki vodnih kapelj. Niso vedno vidni na generiranih slikah, vendar jih je mogoče opaziti na aktivacijah znotraj omrežja generatorja. Problem je sistemski, saj so artefakti prisotni na vseh slikah. StyleGAN2 predлага spremembe, ki to odpravijo.

ponovno preučijo postopno rast (ang. progressive growing) in regulirajo generator, in s tem spodbudijo dobro pogojenost pri preslikavi latentnih kod v slike.

Poleg izboljšanja kakovosti slik, regulator dolžine poti (ang. Path length regularization) prinaša dodatno korist, saj je generator dosti lažje invertirati. To omogoča zanesljivo pripisovanje latentne kode poljubni sliki. Povedano drugače, iz poljubne slike lahko pridobimo latentno kodo, ki v latentnem prostoru predstavlja dober približek originalni sliki, če bi iz te latentne kode spet generirali originalno sliko.

Spremembe v arhitekturi generatorja

- (a) Originalni generator StyleGAN, kjer A označuje naučeno afino transformacijo iz W , ki ustvari slog, B pa je operacija oddajanja šuma
- (b) Isti diagram StyleGAN kot pri (a), le da prikazuje več podrobnosti. AdaIN operacija je razbita na normalizacijo in sledečo modulacijo, obe delajoči na povprečju in standardnem odklonu za vsak feature map posebej. Anotirani so tudi naučene uteži (w), pristranskosti b in konstantni vhod c ter na novo narisana siva polja, tako da je aktiven po en slog v vsakem polju. Aktivacijska funkcija (puščajoča ReLU) se vedno uporabi takoj po dodajanju pristranskosti.



Slika 3.7: (a) Originalni generator StyleGAN (b) Originalni generator StyleGNA, podrobnejše (c) Diagram novega StyleGAN2 generatorja (d) Operacija demodulacije

(c) Diagram novega StyleGAN2 generatorja. Odstranjene so nekatere odvečne operacije na začetku, seštevanje b (pritranskosti) in B (dodajanje šuma) sta premaknjena izven aktivnega območja sloga, kjer delujeta na normaliziranih podatkih, kar se izkaže za bolj predvidljivo (njun vpliv ni več inverzno odvisen od magnitude dodanega sloga). Pri spremenjeni arhitekturi se izkaže, da je zadostno, da modulacija in normalizacija delujeta le nad standardnim odklonom za vsak feature map - povprečje je redundantno.

(d) Spremenjena arhitektura omogoča zamenjavo normalizacije primerov z operacijo "demodulacije", ki jo uporabimo nad utežimi w, povezane z vsako konvolucijsko plastjo. Demodulacija je operacija, ki je uporabljena namesto normalizacije primerov (AdaIN normalizacija). Deluje bolj direktno, na podlagi statistične analize o signalu, namesto dejanskih podatkih o feature mapih, vendar je zato tudi bolj šibka.

Projekcija slike v latentni prostor

Za manipulacijo poljubne slike v prostoru latentnih značilnosti je treba najprej poiskati ujemajočo latentno kodo w zanjo. Predlagatelji StyleGAN arhi-

tekture, tega niso implementirali, bilo pa je implementirano kasneje in predstavljeno v člankih **Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space?** [1] in **Style Generator Inversion for Image Enhancement and Animation** [5]. Te implementacije poskušajo najti posamezno optimalno latentno kodo w za vsak sloj konvolucije, kar izboljša podobnost originalni sliki. Problem takšne implementacije pa je, da omogoča preslikavo poljubne slike, ki ne bi smela imeti reprezentacije v latentnem prostoru.

Za nas je projekcija slike v latentni prostor najpomembnejša spremembra oziroma dodatek, ki ga predлага StyleGAN2 arhitektura. Pridobitev latentne kode iz poljubne slike omogoča veliko lažje operacije nad slikami. Čeprav ta preslikava ni popolna, je dovolj dobra za našo uporabo.

Predlagatelji StyleGAN2 arhitekture torej implementirajo projekcijo slike v latentni prostor na tak način, da poiščejo eno latentno kodo v prvotnem, nerazširjenem latentnem prostoru, saj te ustrezano slikam, ki bi jih lahko ustvaril generator.

Njihov implementacija se od prejšnjih razlikuje v dveh osnovnih stvareh. Prva, med optimizacijo latentni kodi dodajo zmanjšan šum, da bi čim bolj celotno preiskali latentni prostor. Druga, optimizirajo vhode stohastičnega šuma pri StyleGAN2 generatorju - jih regularizirajo, da ne prenašajo koherentnega signala ozitoma šuma. Regularizacija temelji na posodabljanju avtokorelačijskih koeficientov šuma, ki se morajo ujemati z temi Gaussovega šuma na več lestvicah.

3.3 Napovedni modeli

Cilj diplomske naloge je iz slik obrazov otrok generirati slike obrazov staršev tako, da bodo po objektivno določenih metrikah čim bolj podobni njihovim pravim staršem. V sami osnovi gre za problem napovedovanja rezultatov na podlagi pridobljenih podatkov s pomočjo prednaučenih napovednih modelov.

V prvem koraku smo uporabili StyleGAN2 projektor za pridobitev laten-

tnih kod vseh slik, na katerih bomo izvajali učenje ter kasneje predikcijo. Slikam smo tako zmanjšali dimenzijo, tako lahko veliko lažje izvajamo računske operacije in s tem učenje napovednih modelov. Dimenzija prvotnih slik je bila 1024x1024x3, dimenzija latentne kode določene slike pa je 18x512, kar je približno 34.000-kratno zmanjšanje.

3.3.1 Linearna regresija

Linearna regresija je statistična metoda, ki se uporablja za modeliranje odvisnosti med odvisno spremenljivko in eno ali več neodvisnimi spremenljivkami. Cilj linearne regresije je najti linearne razmerje med spremenljivkami, ki se lahko uporablja za napovedovanje odvisne spremenljivke na podlagi vrednosti neodvisnih spremenljivk.

Model linearne regresije (ang. Linear Regression Model), podan z enačbo (3.4), opredeljuje odvisnost med odvisno spremenljivko y in neodvisnimi spremenljivkami x_1, x_2, \dots, x_p . Model predpostavlja, da obstaja linearne razmerje med spremenljivkami in da je vsako odstopanje od linearne zveze posledica naključne napake ϵ_i . Model vključuje $p+1$ neznanih parametrov $\beta_0, \beta_1, \dots, \beta_p$, ki predstavljajo odmik in naklone regresijske premice.

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip} + \epsilon_i \quad (3.4)$$

Ocenjevanje z najmanjšimi kvadrati (ang. Least Squares Estimation), podano z enačbo (3.5), se uporablja za ocenjevanje vrednosti neznanih parametrov, ki minimizirajo vsoto kvadratov napak med opazovanimi vrednostmi odvisne spremenljivke in napovedanimi vrednostmi na podlagi neodvisnih spremenljivk. To se izvede z iskanjem vrednosti $\beta_0, \beta_1, \dots, \beta_p$, ki minimizirajo vsoto kvadratov ostankov $\sum_{i=1}^n (y_i - \hat{y}_i)^2$.

$$\hat{\beta} = \operatorname{argmin}_{\beta} \sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_{i1} - \beta_2 x_{i2} - \dots - \beta_p x_{ip})^2 \quad (3.5)$$

Ocenjena regresijska funkcija (ang. Estimated Regression Function), podana z enačbo (3.6), predstavlja napovedane vrednosti odvisne spremenljivke

na podlagi ocenjenih vrednosti parametrov $\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_p$ in vrednosti neodvisnih spremenljivk x_1, x_2, \dots, x_p . Ostanki (ang. Residuals), podani z enačbo (3.7), predstavljajo razlike med opazovanimi vrednostmi odvisne spremenljivke in napovedanimi vrednostmi na podlagi ocnjene regresijske funkcije.

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \dots + \hat{\beta}_p x_p \quad (3.6)$$

$$e_i = y_i - \hat{y}_i \quad (3.7)$$

Vsota kvadratov ostankov (ang. Sum of squared residuals), podana z enačbo (3.8), predstavlja celotno količino napake v regresijskem modelu. Celotna vsota kvadratov (ang. Total Sum of Squares), podana z enačbo (3.9), predstavlja skupno variacijo odvisne spremenljivke okoli njenega povprečja. Koeficient determinacije (ang. Coefficient of Determination) ali R^2 , podan z enačbo (3.10), predstavlja delež variacije odvisne spremenljivke, ki ga razloži regresijski model. Vrednost R^2 , ki je blizu 1, kaže na dobro prileganje med modelom in podatki.

$$SSE = \sum_{i=1}^n e_i^2 \quad (3.8)$$

$$SST = \sum_{i=1}^n (y_i - \bar{y})^2 \quad (3.9)$$

$$R^2 = \frac{SST - SSE}{SST} = 1 - \frac{SSE}{SST} \quad (3.10)$$

Standardna napaka ocene (ang. Standard Error of the Estimate), podana z enačbo (3.11), je mera povprečne razdalje med opazovanimi vrednostmi odvisne spremenljivke in napovedanimi vrednostmi na podlagi regresijskega modela. Interval zaupanja (ang. Confidence Interval) za β_j , podan z enačbo (3.12), predstavlja območje vrednosti, v katerem je verjetno, da bo prava vrednost β_j , s določeno stopnjo zaupanja.

$$s = \sqrt{\frac{SSE}{n - p - 1}} \quad (3.11)$$

$$\hat{\beta}_j \pm t\alpha/2, n - p - 1 \frac{s}{\sqrt{\sum_{i=1}^n (x_{ij} - \bar{x}_j)^2}} \quad (3.12)$$

kjer je $t_{\alpha/2, n-p-1}$ percentil $1 - \alpha/2$ porazdelitve t z $n - p - 1$ stopnjami prostosti.

Na splošno funkcije, ki opisujejo linearne regresije, delujejo skupaj za oceno parametrov linearnega odnosa med odvisno spremenljivko in eno ali več neodvisnimi spremenljivkami ter ocenjevanje prileganja regresijskega modela podatkom. Model se lahko uporablja za napovedovanje vrednosti odvisne spremenljivke na podlagi vrednosti neodvisnih spremenljivk, ocenjene parametre pa je mogoče interpretirati za razumevanje narave odnosa med spremenljivkami.

Matrična oblika

Vse enačbe, ki opisujejo linearno regresijo, v matrični obliki. Ta je bolj relevantna za nadaljevanje, saj se bolj enostavno odraža v kodi.

The hypothesis function:

$$h_{\beta}(\mathbf{X}) = \mathbf{X}\beta$$

The loss function:

$$J(\beta) = \frac{1}{2n}(\mathbf{X}\beta - \mathbf{y})^T(\mathbf{X}\beta - \mathbf{y})$$

The gradient of the loss function:

$$\nabla_{\beta} J(\beta) = \frac{1}{n}\mathbf{X}^T(\mathbf{X}\beta - \mathbf{y})$$

The normal equation:

$$\beta = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

The prediction:

$$\hat{\mathbf{y}} = \mathbf{X}\beta$$

The sum of squared residuals:

$$SS_{res} = (\mathbf{y} - \hat{\mathbf{y}})^T(\mathbf{y} - \hat{\mathbf{y}})$$

The total sum of squares:

$$SS_{tot} = (\mathbf{y} - \bar{\mathbf{y}})^T(\mathbf{y} - \bar{\mathbf{y}})$$

The coefficient of determination:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}$$

The standard error of the estimate:

$$SE(\epsilon) = \sqrt{\frac{1}{n-p-1} SS_{res}}$$

The confidence interval for β_j :

$$CI(\beta_j) = \beta_j \pm t_{\alpha/2, n-p-1} SE(\hat{\beta}_j)$$

where $t_{\alpha/2, n-p-1}$ is the $1-\alpha/2$ percentile of the t -distribution with $n-p-1$ degrees of freedom.

3.3.2 Ridge normalizacija

Ridge normalizacija, znana tudi kot L2 regularizacija, je tehnika, ki se uporablja za preprečevanje preprileganja v modelih linearne regresije s dodajanjem kazenskega člena v funkcijo izgube. Ta kazenski člen je sorazmeren s kvadratom magnitude koeficientov linearne regresijskega modela. S tem kazenskim členom je model spodbujen k manjšim vrednostim koeficientov, kar zmanjša kompleksnost modela in preprečuje preprileganje.

Matematično se funkcija izgube Ridge regresije lahko zapiše kot:

$$L(\mathbf{y}, \mathbf{X}, \boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}^T \boldsymbol{\beta}$$

kjer je \mathbf{y} vektor vrednosti ciljne spremenljivke, \mathbf{X} matrika vrednosti napovednih spremenljivk, $\boldsymbol{\beta}$ vektor koeficientov, ki jih je treba oceniti, n število opazovanj, λ pa je hiperparameter, ki nadzoruje jakost kazenskega člena.

Prvi člen v funkciji izgube predstavlja povprečno kvadratno napako med predvidenimi vrednostmi in dejanskimi ciljnimi vrednostmi, drugi člen pa je L2 kazenski člen, ki krči ocene koeficientov proti nič. Hiperparameter λ nadzira ravnovesje med temo dvema členoma in večje vrednosti λ vodijo v manjše ocene koeficientov.

Ocene koeficientov Ridge regresije se lahko pridobijo s pomočjo naslednje enačbe:

$$\boldsymbol{\beta} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}$$

kjer je \mathbf{I} identitetna matrika. Ta enačba je podobna enačbi za običajni najmanjši kvadrat, vendar s dodatkom člena λ .

Ridge regresija na splošno dodaja regularizacijski člen v model linearne regresije, kar pomaga zmanjšati preprileganje in izboljšati splošno učinkovitost modela.

Izbira značilk za linearno regresijo

V našem primeru je neodvisna spremenljivka latentna koda, pridobljeno iz določenega obraza otroka. Model pa napove odvisno spremenljivko, katere polovica predstavlja latentno kodo obraza mame, polovica pa latentno kodo obraza očeta. Značilke lahko sestavimo na več različnih načinov, najboljšega pa lahko določimo s poskušanjem. Dve možnosti, ki nam takoj prideta na misel sta zaporedno sestavljena značilka iz latentnih kod mame in očeta in izmenično sestavljena značilka iz latentnih kod mame in očeta. To je kasneje bolj podrobno opisano v poglavju **4.4 Lastna implementacija**

Poglavlje 4

Implementacija lastne metode

4.1 Struktura projekta

Projekt je narejen v okviru sistema za upravljanje paketov Conda [2], tako je lahko pognan s pomočjo Conde na katerem koli računalniku neodvisno od že nameščenih Python knjižnic. Conda omogoča enostavno vzpostavitev Python virtualnega okolja, znotraj katerega na začetku namestimo vse potrebne knjižnice, kodo pa potem poganjamo na Python prevajalniku iz Conda virtualnega okolja.

Pri razvoju smo uporabljali sistem za nadzor različic (ang. Version control system) GitHub [8], ki je predvsem omogočil enostavnejšo integracijo dveh zunanjih modulov StyleGAN2 (v repozitoriju NVlabs/stylegan2-adaptorch) [15] in InterfaceGAN (v repozitoriju genforce/interfacegan) [7] v projekt. S pomočjo GitHubovih submodulov smo v projekt vključili obe zbirki podatkov NOKDB in PPLDB, za katera smo ustvarili ločena repozitorija za bolj efektivno razmnoževanje projekta v prihodnosti.

4.1.1 Organizacija kode

Celoten projekt zajema več repozitorijev.

4.1.2 Struktura nabora slikovnih podatkov

NOKDB [*<empty citation>*]

Za učenje napovednih modelov potrebujemo čim bolj kvalitetne podatke, saj je od njih precej ovisna kakovost naučenih modelov. Zato smo uporabili nov nabor podatkov, ki omogoča učenje modelov na podlagi sorodstev. Next Of Kin (NOK) vsebuje 3690 512×512 slik visoke ločljivosti obrazov 553 različnih ljudi z veliko etnično in starostno raznolikostjo. V nabor podatkov so vključeni tudi metapodatki o sorodstvenih vezeh, starosti, etnični pripadnosti, spolu in izrazu obraza.

Nabor podatkov vsebuje več kot 120.000 trojic oče - mama - otrok na katerih je potekalo učenje modelov. Vse slike v naboru podatkov so bile predprocesirane. Najprej so bili poiskani obrazni atributi in glede na njihovo pozicijo so bile slike obrezane in pravilno orientirane. Nato so bile vse slike projecirane v W+ latentni prostor (latentni prostor prednaučenega StyleGAN2 FFHQ modela), latentna koda (iid.npz) se nahaja poleg vsake izmed slik. Na koncu so bili dodani metapodatki o spolu, etnični pripadnosti vsaki osebi in starost in podatek o izrazu na obrazu vsaki izmeni slik.

Podani so tudi uradni učni, validacijski in testni split, kjer so trojice oče - mama - otrok razporejene tako, da se slike specifičnega otroka ne pojavitajo v več kot enem izmed njih. Spliti so definirani v datotekah nokdb-samples, poleg treh ki se uporablja za učenje in izboljševanje modela smo pripravili še dva druga splita, ki se uporablja za generiranje realnih slik staršev iz slik otrok ali sli otrok iz lik staršev. Latentne kode za te slike pridejo v NOKDB iz slikovne podatkovne zbirke PPLDB.

Značilnosti nabora podatkov	Število
persons	1116
images	3690
images per person	3.31
image sample combinations	127,719
father/mother/son person triplets	116
father/mother/daughter person triplets	133
image resolution	512x512

Pot	Opis
— nokdb-images.csv	Seznam slik z atributi
— nokdb-persons.csv	Seznam oseb z atributi
— nokdb-samples-test.csv	Trojčki oče-mama-otrok (testni)
— nokdb-samples-train.csv	Trojčki oče-mama-otrok (učni)
— nokdb-samples-validation.csv	Trojčki oče-mama-otrok (validacijski)
— nokdb-samples-all.csv	Trojčki oče-mama-otrok (vsi)
— nokdb-samples-real-c2p.csv	Trojčki oče-mama-otrok za generiranje staršev
— nokdb-samples-real-p2c.csv	Trojčki oče-mama-otrok za generiranje otrok
— nokdb-normalization.npz	W vektor normalizacija povprečje in standardni odklon za vsako dimenzijo

Poti	Opis
— pid0/	Direktorij, ki vsebuje nabor podatkov osebe ‘pid0’
—— iid0.png	Slika ‘iid0’, ki pripada osebi ‘pid0’
—— iid0.npz	StyleGAN2 latentna koda slike ‘iid0’
—— iid1.png	Slika ‘iid1’, ki pripada osebi ‘pid0’
—— iid1.npz	StyleGAN2 latentna koda slike ‘iid1’
— pid1/	
—— ...	
— ...	

PPLDB

Medtem ko je NOKDB usmerjen pretežno v učenje modelov in kvalitete teh, smo opazili potrebo po lastnem naboru podatkov, v katerem bodo hranjeni slikovni podatki iz spletne aplikacije, kjer je model uporabljen. Tu se predvsem kopijo slike, ki so lahko kasneje ali avtomatsko migrirane v NOKDB, za ponovno učenje napovednih modelov. Podobno kot NOKDB tudi PPLDB na začetku slike predprocesira. Najprej so slike obrezane in rotirane na podlagi zaznanih obraznih atributov. Nato so projecirane v StyleGAN2 FFHQ latentni prostor W^+ , latentne kode pa so shranjene, poleg podatkov o loss in gain pri projekciji, v PPLDB.

V datoteki nokdb-mapping.csv so hranjeni podatki o mapiranju oseb iz PPLDB v NOKDB. Zapisano je njihov pid iz PPLDB in njihov pid iz NOKDB. V datoteki couples.csv so hranjeni podatki o parih oseb, katerim lahko generiramo otroka. Potrebna je predhodna migracija podatkov o paru iz PPLDB v NOKDB.

Pot	Opis
— couples.csv	Seznam parov, katerim generiramo otroka
— nokdb-mapping.csv	Seznam oseb z pripisanim NOKDB pid

Pot	Opis
— pid0/	Direktorij, ki vsebuje podatke osebe ‘pid0’
—— latents/	Direktorij, ki vsebuje latentne podatke osebe ‘pid0’
——— latents.pid0_iid0/	Direktorij, ki vsebuje latentne podatke slike ‘pid0_iid0’
———— target.png	Predprocesirana slika ‘iid0’, ki pripada osebi ‘pid0’
———— proj.png	Slika, pripadajoča generirani latentni kodi
———— projected_w.npz	Generirana latentna koda v prostoru W+
———— loss.png	Graf loss in dist, glede na korak projekcije
——— latents.pid0_iid1/	Direktorij, ki vsebuje latentne podatke slike ‘pid0_iid1’
———— iid1.png	Predprocesirana slika ‘iid1’, ki pripada osebi ‘pid0’
———— iid0.png	Predprocesirana slika ‘iid0’, ki pripada osebi ‘pid0’
—— aligned/	Direktorij, ki vsebuje predprocesirane osebe ‘pid0’
——— pid0_iid0.png	Predprocesirana slika ‘iid0’, ki pripada osebi ‘pid0’
——— pid0_iid1.png	Predprocesirana slika ‘iid1’, ki pripada osebi ‘pid0’
——— pid0_iid0.png	Originalna slika ‘iid0’, ki pripada osebi ‘pid0’
——— pid0_iid1.png	Originalna slika ‘iid1’, ki pripada osebi ‘pid0’
— pid1/	Direktorij, ki vsebuje podatke osebe ‘pid1’
—— ...	
—— ...	

4.2 Opis uporabljenih tehnologij in metod

4.2.1 StyleGAN

4.2.2 Interfacegan

4.2.3 Wandb

4.3 Skupna implementacija

Matjaž Mav je pripravil osnovno implementacijo, katero sem v okviru moje diplomske naloge samo razširil in v določenih delih spremenil ter dopolnil.

Matjaž je najprej implementiral vmesnik za pridobivanje strukturiranih podatkov iz slikovne podatkovne zbirke NOKDB na več različnih načinov - datoteke v direktoriju fri-2022-diploma/src/dataset. V direktoriju fri-2022-diploma/src/utils je implementiral podporne mehanizme za izvajanje programa - vmesnik za StyleGAN2 submodul, funkcije za vizualizacijo in funkcije za pridobivanje podatkov iz GitHuba (prenos zbirke podatkov). V direktorijskih so delno dokočane implementacije različnih napovednih modelov, ki jih je Matjaž testiral.

4.3.1 Vmesnik za pridobivanje strukturiranih podatkov iz slikovne podatkovne zbirke NOKDB

Osnovna naloga vmesnika je pridobivanje podatkov - latentnih kod, iz zbirke slikovnih podatkov NOKDB, s katerimi potem operiramo pri učenju, validaciji ali testiranju modelov. Uporabljen je tudi za pridobivanje latentnih kod, ki se uporabijo za generiranje slik staršev oziroma otrok iz realnih slik.

Vmesnik je implementiran kot Python razred, kateremu ob inicializaciji podamo določene parametre, od katerih je odvisno kateri podatki se pridobijo iz podatkovne zbirke NOKDB. Osnovna funkcija razreda je dovajanje podatkov (latentnih vektorjev) v razredno spremenljivko data, odvisno od podanega splita - train/validation/test oziroma P2C ali C2P. Informacije o tem kateri latentni vektorji (katera starša pripadata kateremu otroku in obratno) se naložijo v spremenljivko data pa vmesnik pridobi iz datotek nokdb-samples, odvisno od podanega splita.

Funkcija, v kateri je implementirano nalaganje vektorjev je zasnovana tako, da latentne kode posameznega človeka združi v eno - naredi povprečje vseh latentnih kod, ki se potem uporabi v nadaljnem napovedovanju (učenje/validacija/testiranje oziroma generiranje rezultatov). Takšen način se je izkazal za najbolj učinkovitega pri končnem generiranju rezultatov. Poleg opisanega vmesnika, ki pridobi s povprečjem združeno latentno kodo posameznika, smo implementirali tudi druge, vendar se niso izkazali za tako natančne pri nadalnjem napovedovanju.



Slika 4.1: Slike, ki predstavljajo eno osebo v zbirki podatkov NOKDB

4.3.2 Vmesnik za StyleGAN2 submodul

Uporaba StyleGAN2 arhitekture, je omogočena skozi vmesnik implementiran v datoteki `fri-2022-diploma/src/utils/stylegan.py`. V okviru vmesnika sta za uporabo podani dve osnovni StyleGAN2 operaciji `project` in `generate`, z različnimi možnostmi podajanja argumentov. Vmesnik izvaja python skripte definirane v submodulu repozitorija `stylegan2-ada-pytorch` [15] preko Docker zabojnika nad sliko `stylegan2-ada-pytorch:latest`, tako je zasnovan robustno, saj ni neposredno odvisen od okolja, v katerem ga poganjamo.

4.3.3 Napovedni modeli

Matjaž Mav je v okviru svoje naloge implementiral in testiral več napovednih modelov, vendar so bili ti narejeni za generacijo sli otrok na podlagi slik



Slika 4.2: Povprečna slika osebe, generirana iz povprečne latentne kode vseh slik te osebe v zbirki podatkov NOKDB

staršev, jaz pa v svoji nalogi potrebujem napovedni model za napovedovanje staršev iz otrok.

4.4 Lastna implementacija

Osnovno implementacijo sem sam razširil predvsem z novim modelom, ki iz slik otrok generira slike staršev, za kar sem moral prilagoditi tudi ostale podporne mehanizme. Drugi dodatek pa je implementacija in integracija nove podatkovne zbirke PPLDB, ki omogoča vnos novih oseb in razmerij med osebami v podatkovno zbirko NOKDB. Zadnja stvar, ki sem jo naredil je ovrednotenje generiranih slik in vizualizacija teh rezultatov.

4.4.1 PPLDB

Za implementacijo nove zbirke podatkov in njenih podpornih mehanizmov smo se odločili zaradi potrebe po širši uporabi sistema. PPLDB omogoča dodajanje novih oseb in njenih slik v celoten sistem. V nadaljevanju bomo glede na tok podatkov ene osebe opisali funkcionalnosti sistema PPLDB.

Slike osebe shranimo v nov direktorij poimenovan po osebi v direktoriju fri-2022-diploma/src/dataset/ppldb, npr.
fri-2022-diploma/src/dataset/ppldb/vid/slika1.jpg, Ob naslednjem zagonu pro-

grama se sproži podatkovni cevovod, ki bo omogočil generiranje slik staršev. Najprej se požene funkcija, ki dodani osebi pripiše identifikacijsko številko PPLDB in slike prestavi v nov direktorij, ter jih hkrati preimenuje v standar-dizirano ime "ime_iid" npr.

fri-2022-diploma/src/dataset/ppldb/people/ppldb_pid/vid_1.jpg. Takoj za tem poteče predprocesiranje slik, kjer so obrezane in rotirane na podlagi zaznanih obraznih atributov. To je ključnega pomena za nadaljno uporabo slik pri generiranju latentnih kod, kar je naslednji korak. Vsaka slika, ki je bila uspešno predprocesirana je nato projecirana v latentni prostor s pomočjo StyleGAN2 funkcije project, ki jo kličemo preko vmesnika za StyleGAN2 submodul. Rezultat projekcije so latentne kode, graf spremenjanja rezultata funkcije izgube in razdalje skozi iteracije projekcije in slika, ki prikazuje generirano latentno kodo v originalnem nelatentnem prostoru.

Osnovna funkcija zbirke podatkov PPLDB je le shranjevanje latentnih kod, ki pripadajo realnim slikam. Njena uporabnost pa je v tem, da dovaja podatke v zbirko podatkov NOKDB preko katere kasneje poteka pridobivanje podatkov za generiranje slik staršev.

4.4.2 Generacija slik staršev

Celoten proces generacije slik staršev s pomočjo že naučenega modela je razdeljen na več manj obsežnih operacij. Najprej, poteče operacija opisana v prejšnjem podpoglavlju - populacija PPLDB baze s latentnimi kodami slik in migracija teh latentnih kod in ostalih podatkov v podatkovno zbirko NOKDB.

Nato se preko vmesnika, opisanega v poglavju **Vmesnik za pridobivanje strukturiranih podatkov iz slikovne podatkovne zbirke NOKDB** pridobijo podatki, ki jih nato posljemo v določen model. Podatki so povprečne latentne kode, izračunane iz latentnih kod vseh slik določene osebe.

Prednaučen model linearne regresije potem generira latentne kode najbolj verjetnih staršev za podane latentne kode.

Zadnji korak pred ovrednotenjem rezultatov je generacija slik iz generi-

ranih latentnih kod. Končne slike generiramo s pomočjo modula opisanega v poglavju **Vmesnik za StyleGAN2 submodul**. Znotraj modula kličemo StyleGAN2 funkcijo generate, ki podanemu nizu latentnih kod generira pri-padajoče slike. Preden poteče ovrednotenje rezultatov slike še zložimo v kolaž, da je pregledovanje rezultatov bolj pregledno.

4.4.3 Rezultati

Predstavili bomo rezultate pridobljene na dva različna načina, glede na vhodne podatke. Prvi način so rezultati pridobljeni le s pomočjo podatkovne zbirke NODKB, kjer je model naučen in testiran le na podatkih iz te zbirke (ločeni podatki za učenje in testiranje modela). Pri drugem načinu pa bomo predstavili rezultate pridobljene s pomočjo modela naučenega na vseh NOKDB podatkih, testiranje pa poteka na lastnih oziroma zunanjih slikah, ki smo jih v NOKDB podatkovni zbirko vnesli preko sistema PPLDB. V poglavju Ovrednotenje bomo rezultate kvalitativno ovrednotili s pomočjo tehnik razpoznavanja obrazov, v tem poglavju pa bomo le podali slikovne rezultate in jih vizualno opisali ter razložili možne napake v generiranju oziroma opredelili vizualne artefakte.

NOKDB rezultati

Ti rezultati so bili pridobljeni z enakim načinom kot poteka ovrednotenje, opisano kanseje. V tem poglavju bom le vizualno predstavil pridobljene slike. Na prvi pogled delujejo generalno rezultati precej dobri, brez veliko vidnih artefaktov, vendar se tudi ti občasno pojavljajo. Artefakti se po navadi pojavijo v primerih, kjer poskušamo generirati starše odraslemu človeku. Razlog za to je v tem, da modela nismo učili na slikah, kjer bi bili otroci že starejši (ali pa je bilo teh primerov v učni množici premalo). Temu bi se lahko izognili, če bi ob generiranju preverjali starost otroka in ustrezno prilagodili sliko (oziroma latentno kodo) tako, da bi mu zmanjšali starost brez sprememb identitete. To bi lahko implementirali s pomočjo submodula InterfaceGAN.

Če vizualno primerjamo generirane starše realnim, ugotovimo da model bolje napoveduje mamo kot očeta, česar nismo mogli spremeniti kljub drugačni postavitvi značilk pri vhodu v linearo regresijo. Za ugotovitev zakaj je to tako, bi potrebovali več časa.

Lastni rezultati

Poglavlje 5

Ovrednotenje rezultatov

Da bi ovrednotili podobnost identitet generiranih slik staršev z realnimi slikami bomo uporabili različne modele za razpoznavo obrazov. Takšnih modelov je kar nekaj, mi bomo uporabljali tri precej splošno razširjene FaceNet [19], ArcFace [4] in VGG-Face [17].

5.1 Deepface knjižnica za razpoznavo obrazov

Deepface je python knjižnica oziroma ogrodje, ki omogoča hitro in enostavno uporabo state-of-the-art modelov za razpoznavo obrazov, prepoznavo obrazov, podobnost obrazov in analizo obraznih atributov. Vsebuje implementacije VGG-Face, Facenet, Facenet512, OpenFace, DeepFace, DeepID, ArcFace, Dlib in SFace modelov. Uporabili smo jo, ker je zelo enostavna za uporabo in smo jo zlahka vključili v lastno implementacijo razreda, namejenega ovrednotenju rezultatov. Knjižico smo uporabili le za eno funkcijo, ki iz določene slike obraza pridobi skrčeno predstavitev oziroma nek vektor, ki predstavlja atrIBUTE obraza (ang. embedding).

5.2 Ovrednotenje

Po uporabi Deepface knjižnice za pridobitev skrčene predstavitve obraza generirane slike in obraza prave slike enakega primera, ju primerjamo s pomočjo neke razdalje. Razdalja uporabljeni v implementaciji je kosinusna, saj je najbolj primerna za vektorje takšnih dimenzij. Pri ovrednotenju smo izbrali strategijo pozitivnih parov, za razliko od drugih podobnih implementacij, ki uporabljajo strategijo pozitivnih in negativnih parov. To pomeni, da smo primerjali le realno sliko starša z generirano sliko starša (strategija pozitivnih parov). Nismo pa tudi primerjali razlike realne slike starša s naključno generirano sliko starša (strategija negativnih parov), saj menimo, da ta razlika ni tako pomembna kot podobnost pozitivnega para.

Napovedni model smo v primeru generiranja slik obrazov staršev, na podlagi slik obrazov otrok, ovrednotili tako, da smo primerjali realne slike obrazov staršev z generiranimi slikami obrazov staršev. To smo za vsak primer naredili s tremi modeli prepozname obrazov ArcFace, Facenet in VGG-Face.

Da smo dobili čim bolj spodobne rezultate smo ovrednotenje opravili večkrat, po podobnem principu kot poteka prečna validacija modela (ang. cross validation). To pomeni, da smo model vrednotili v več iteracijah in na koncu vzyeli povprečje rezultatov vseh iteracij, tako so rezultati bolj reprezentativni, kot če bi opravili le eno iteracijo ovrednotenja. Najprej smo razdelili vse otroke na k množic, tako smo po principu k-fold cross validationa v vsaki izmed k iteracij pridobili učno in testno množico podatkov, kjer se je testna množica otrok menjala v vsaki iteraciji. Podatke za učenje in testiranje (kombinacije slik trojice oče-mama-otrok) smo iz zbirke podatkov pridobili preko informacije o tem ali otrok spada v učno ali testno množico podatkov. S pridobljenimi podatki smo pognali učenje modela in testiranje modela na testni množici podatkov. Po vsaki iteraciji smo hranili rezultate primerjav očetov in mam posebej, ter na koncu naredili povprečje vseh iteracij. Iz rezultatov podobnosti smo zgradili histograma, ki prikazujeta števila vzorcev, ki pripadajo določeni gostoti podobnosti med realnim in generiranim parom.

Ker modela za generiranje slik obrazov staršev na podlagi slik obrazov otrok do tega trenutka še ni, našega modela ne moremo neposredno primerjati oziroma ovrednotiti. Lahko primerjamo naše vrednosti s vrednostmi modelov za generiranje slik obrazov otrok iz slik staršev, vendar ovrednotenje ne bo validno.

5.3 Vizualizacija

Za lažjo predstavo rezultatov smo iz dobljenih kosinusnih podobnosti zgradili histograme. To smo naredili tako, da smo za vsak modeli prepozname obrazov, zgradili histogram napovedi kosinusnih podobnosti in histogram popolnih kosinusnih podobnosti. Za kvantitativno predstavitev rezultatov smo izračunali tudi razliko med napovednim in popolnim histogramom s razdaljo earth mover's distance, ter povprečje kosinusnih podobnosti primerov.

Poglavlje 6

Sklepne ugotovitve

6.1 Zaključek

6.2 Primerjava znanih metod z lastno implementacijo

6.3 Možne izboljšave in dopolnitve

Literatura

- [1] Rameen Abdal, Yipeng Qin in Peter Wonka. “Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space?” V: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (okt. 2019). DOI: 10.1109/iccv.2019.00453. URL: <http://dx.doi.org/10.1109/ICCV.2019.00453>.
- [2] Inc. Anaconda. *Conda*. 12.04.2023. 2017. URL: <https://docs.conda.io/en/latest/>.
- [3] Xiao Cui in sod. “Heredity-aware Child Face Image Generation with Latent Space Disentanglement”. V: *arXiv preprint arXiv:2108.11080* (2021).
- [4] Jiankang Deng in sod. “ArcFace: Additive Angular Margin Loss for Deep Face Recognition”. V: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, str. 4685–4694. DOI: 10.1109/CVPR.2019.00482.
- [5] Aviv Gabbay in Yedid Hoshen. *Style Generator Inversion for Image Enhancement and Animation*. 2019. arXiv: 1906.11880 [cs.CV].
- [6] Peng Gao in sod. “What Will Your Child Look Like? DNA-Net: Age and Gender Aware Kin Face Synthesizer”. V: *ICME*. 2021.
- [7] genforce. *InterfaceGAN*. 12.04.2023. 2023. URL: <https://github.com/genforce/interfacegan>.
- [8] Inc. GitHub. *GitHub*. 12.04.2023. 2023. URL: <https://github.com/>.

- [9] Ian Goodfellow in sod. “Generative adversarial nets”. V: *Advances in neural information processing systems* 27 (2014).
- [10] Xun Huang in Serge Belongie. “Arbitrary style transfer in real-time with adaptive instance normalization”. V: *Proceedings of the IEEE international conference on computer vision*. 2017, str. 1501–1510.
- [11] Tero Karras, Samuli Laine in Timo Aila. “A style-based generator architecture for generative adversarial networks”. V: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, str. 4401–4410.
- [12] Tero Karras in sod. “Analyzing and Improving the Image Quality of StyleGAN”. V: *Proc. CVPR*. 2020.
- [13] Tero Karras in sod. “Progressive growing of gans for improved quality, stability, and variation”. V: *arXiv preprint arXiv:1710.10196* (2017).
- [14] Yong Li in sod. “KinNet: Fine-to-Coarse Deep Metric Learning for Kinship Verification”. V: *Proceedings of the 2017 Workshop on Recognizing Families In the Wild* (2017).
- [15] NVlabs. *NVlabs/stylegan2-ada-pytorch*. 12.04.2023. 2023. URL: <https://github.com/dvschultz/stylegan2-ada-pytorch>.
- [16] Savas Özkan in Akin Ozkan. “Kinshipgan: Synthesizing of Kinship Faces from Family Photos by Regularizing a Deep Face Network”. V: *2018 25th IEEE International Conference on Image Processing (ICIP)* (2018), str. 2142–2146.
- [17] Omkar M. Parkhi, Andrea Vedaldi in Andrew Zisserman. “Deep Face Recognition”. V: *British Machine Vision Conference*. 2015.
- [18] Joseph Robinson in sod. “Visual Kinship Recognition of Families in the Wild”. V: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP (apr. 2018), str. 1–1. DOI: 10.1109/TPAMI.2018.2826549.

- [19] Florian Schroff, Dmitry Kalenichenko in James Philbin. “Facenet: A unified embedding for face recognition and clustering”. V: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, str. 815–823.
- [20] NEJC ŠUKLJE. “Napovedovanje obrazov otrok z GAN metodo”. Doktorska disertacija. Univerza v Ljubljani, 2022. URL: <https://repozitorij.uni-lj.si/IzpisGradiva.php?lang=slv&id=143466>.
- [21] Kaihao Zhang in sod. “Kinship Verification with Deep Convolutional Neural Networks”. V: *Proceedings of the British Machine Vision Conference (BMVC)*. Ur. Mark W. Jones Xianghua Xie in Gary K. L. Tam. BMVA Press, sep. 2015, str. 148.1–148.12. ISBN: 1-901725-53-7. DOI: 10.5244/C.29.148. URL: <https://dx.doi.org/10.5244/C.29.148>.