

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Nejc Šuklje

# **Napovedovanje obrazov otrok z GAN metodo**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM  
PRVE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Luka Šajn

Ljubljana, 2022

Copyright. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

*Besedilo je oblikovano z urejevalnikom besedil  $\text{\LaTeX}$ .*

**Kandidat:** Nejc Šuklje

**Naslov:** Napovedovanje obrazov otrok z GAN metodo

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

**Mentor:** doc. dr. Luka Šajn

**Opis:**

Napovedovanje obrazov otrok z GAN metodo na podlagi videza obeh staršev

**Title:** Predicting children faces with GAN method

**Description:**

Predicting children faces with GAN method based on looks of both parents



*Zahvalil bi se profesorjem in asistentom, ki so mi predali znanje za dokončanje predmetov, mentorju zaključnega dela, tistim, ki so mi podali mnenje o rezultatih, in vsem ostalim, ki so mi stali ob strani med študijem.*



# Kazalo

Povzetek

Abstract

<b>1</b>	<b>Uvod</b>	<b>1</b>
1.1	Motivacija . . . . .	1
1.2	Idejna rešitev . . . . .	1
1.3	Cilji in struktura dela . . . . .	2
<b>2</b>	<b>Pregled področja</b>	<b>3</b>
2.1	GAN metoda grobo . . . . .	3
2.2	Prva omemba GAN metode . . . . .	3
2.3	Sorodna dela . . . . .	4
<b>3</b>	<b>Metodologija</b>	<b>5</b>
3.1	GAN metoda podrobneje . . . . .	5
3.2	Generator in diskriminator . . . . .	5
3.3	Predstavitev slike s številkami . . . . .	8
3.4	Latentna predstavitev slik . . . . .	8
3.5	Latentni prostor . . . . .	9
<b>4</b>	<b>Izdelava programa</b>	<b>11</b>
4.1	Začetna koda . . . . .	11
4.2	Slike za učenje in testiranje . . . . .	11
4.3	Koda . . . . .	13

<b>5</b>	<b>Rezultati in analiza</b>	<b>17</b>
5.1	Primerjava-prvi del . . . . .	17
5.2	Primerjava-drugi del . . . . .	20
5.3	Diskusija . . . . .	22
<b>6</b>	<b>Sklepne ugotovitve</b>	<b>25</b>
6.1	Zaključek . . . . .	25
6.2	Možne izboljšave . . . . .	26
	<b>Literatura</b>	<b>27</b>



# Slike

3.1	Delovanje generatorja in diskriminatorja - prvi način[5]	6
3.2	Delovanje generatorja in diskriminatorja - drugi način[4]	8
3.3	Primer kompresije vektorja[10]	9
3.4	Vizualizacija latentnega prostora[10]	10
4.1	13 slik osebe, ki se razlikujejo po starosti in razpoloženju[8]	12
4.2	Končna slika	13
4.3	ID očeta, mame, njihovega otroka in še 3 za nas nepomembni ID-ji	14
4.4	priprava latentnih slik vseh trojk	14
4.5	priprava podatkov za učenje ali testiranje	15
4.6	napovedovanje podatkov	15
5.1	4 primeri slik staršev, njihovega otroka in pa generiranega otroka[8]	17
5.2	prvih 8 primerov slik staršev, njihovega otroka in pa generira- nega otroka[8]	23
5.3	drugih 8 primerov slik staršev, njihovega otroka in pa generi- ranega otroka[8]	24



# Seznam uporabljenih kratic

kratica	angleško	slovensko
GAN	generative adversarial net-work	generativna nasprotovalna mreža
DNA	deoxyribonucleic acid	deoksiribonukleinska kislina



# Povzetek

**Naslov:** Napovedovanje obrazov otrok z GAN metodo

**Avtor:** Nejc Šuklje

Diplomska naloga z naslovom "Napovedovanje slik otrok obrazov z GAN metodo" že na prvi pogled vzbudi motivacijo, ker je potencialni rezultat zanimiv. GAN metoda je tudi dokaj nova stvar, kar vse skupaj naredi še bolj zanimivo, saj ni veliko raziskav, v katerih je uporabljena. Na problem lahko pogledamo tudi s strani "kupca", torej nekoga, ki bi rad videl sliko svojega potencialnega otroka. Po mojem mnenju stvar, ki bi na trgu doživela porast, še posebej v bogatejšem sloju prebivalstva. Čeprav je raziskav na to temo malo, bodo vseeno pomagala pri izdelavi naloge. Morda toliko bolj, ker bo možno bolj podrobno predelati področje. Programerski del bo napisan v programskem jeziku python. Na koncu bomo analizirali rezultate in naredili primerjavo z dejanskimi slikami obrazov otrok in ocenili uspeh. Poiskali bomo tudi točke, ki bi se jih dalo izboljšati, da bi bil izdelek še boljši in uporabnejši.

**Ključne besede:** grebenska regresija, latentni prostor, GAN, generiranje obrazov



# Abstract

**Title:** Prediction of children's faces with the GAN method

**Author:** Nejc Šuklje

The diploma thesis entitled "Prediction of images of children's faces using the GAN method" arouses motivation at first sight, because the potential result is interesting. The GAN method is also a fairly new thing, which makes it all the more interesting because there is not much research in which it is used. We can also look at the problem from the side of the "buyer", that is someone who would like to see a picture of their potential child. In my opinion, a thing that would see an increase in the market, especially in the wealthier population. Although there is little research on this topic, they will still help in making the assignment. Perhaps even more so because it will be possible to rework the area in more detail. The programming part will be written in the python programming language. Finally, we will analyze the results and make a comparison from real images of children and evaluate the success. We will also look for points that could be improved to make the product even better and more useful.

**Keywords:** ridge regression, latent space, GAN, generation of faces.





# Poglavje 1

## Uvod

### 1.1 Motivacija

Zdi se mi, da naslov naloge ob prvem stiku pritegne zanimanje zaradi "koristi" oziroma povezave z realnim življenjem izven računalništva. V tem primeru je potencialna slika otroka moškega in ženske zanimiva zadeva, ki bi na koncu ob uspešnem dokončanju naloge dala najverjetneje dobre in zanimive ter uporabne rezultate. Najverjetneje obstajajo ljudje, ki bi radi videli svojega potencialnega otroka, preden se rodi oziroma odraste, seveda pa zagotovila za pravilno napoved ni. Potencialna rešitev bo narejena z uporabo GAN metode.

### 1.2 Idejna rešitev

Gan metoda je pritegnila veliko pozornosti med raziskovalci na področju računalniškega vida. Največ se uporablja v medicini za sintezo slik. Cilj je pridobiti čim večjo količino podatkov oziroma slik, ki so podobne realnim slikam. Te zgenerirane slike so nato uporabljene skupaj z realnimi za potrebe učenja. Študenti medicine lahko na njih izvajajo analize in se s tem učijo na primerih, ki zaenkrat niso resnični. Zaradi raznolikosti, ki smo jo prinesli z namišljenimi slikami, je lahko pridobljeno znanje mnogo večje[11].

V našem primeru bo idejna rešitev obraz otroka, ki bo nastal iz slik očeta in mame.

## 1.3 Cilji in struktura dela

Delo ne bo potekalo iz čistega začetka, ampak na že prej pripravljeni kodi Matjaža Mava[3]. Osnova je zastavljena kot napovedovanje otrok glede na starše z globokim učenjem(ang. deep learning). V nalogi bodo opisane uporabljene metode in pristop k problemu. Začeli bomo s predstavitvijo GAN metode, potrebnimi splošnimi informacijami za razumevanje le-te. Potem pa pridejo na vrsto postopki in metode, ki so bili uporabljeni, da smo prišli do želenega rezultata. Rezultati bodo primerno analizirani. Primerjana bo podobnost generiranih slik z realnimi. Po analizi bomo tudi predvideli potencialne izboljšave za nadaljnje delo.

## Poglavje 2

# Pregled področja

### 2.1 GAN metoda grobo

GAN metoda je relativno mlada metoda, ki se uporablja pri procesiranju slik. Uporablja se tako, da na začetku uporabimo, učno množico slik, s katero se bo program učil obraznih značilnic. Ta množica se čez čas razširja, ker program proizvaja nove slike, ki so dodane v to množico in so približek tistim, ki so bile v množici že od začetka.

### 2.2 Prva omemba GAN metode

GAN metoda je bila prvič predstavljena in opisana v članku "Generative Adversarial Nets" avtorjev Ian J. Goodfellow, Jean Pouget-Abadie in drugih raziskovalcev Univerze v Montrealu leta 2014. Članek govori o modelu, ki je sestavljen iz dveh funkcij. Prva je zadolžena, da generira podatke, druga pa, da generirane podatke opredeli kot pravilne ali pa napačne. Cilj teh dveh funkcij je, da povečamo verjetnost prepoznavne napake druge funkcije. S tem dosežemo, da je podatkov, opredeljenih kot pravih, malo, ampak so zelo podobni resničnim. Do te mere, da ne moremo ločiti med resničnimi in neresničnimi. Avtorji članka generator in diskriminator razložita na zelo razumljiv način. Generator si lahko predstavljamo kot ponarejevalce, ki po-

skušajo narediti čim bolj pristne ponaredke, ki bodo ostali neodkriti, diskriminator pa kot policijo, ki išče ponaredke in njihovo uporabo na trgu. Modela konstantno izboljšujeta svoji funkciji[1].

## 2.3 Sorodna dela

### 1. Generative adversarial network in medical imaging: A review

Gan metoda je pritegnila veliko pozornosti med raziskovalci na področju računalniškega vida. Največ se uporablja v medicini za sintezo slik in posledično za pridobivanje večje količine podatkov[11].

### 2. Emotion Detection of Autistic Children Using Image Processing

Članek govori o procesiranju slik otrok. Bolj specifično tistih, ki imajo avtizem. Zaradi bolezni je njihovo izražanje čustev različno od ljudi, ki nimajo avtizma. S procesiranjem njihovih izrazov na obrazu se poskuša določiti čustva, ki jih ta trenutek izražajo[7].

## Poglavje 3

# Metodologija

### 3.1 GAN metoda podrobneje

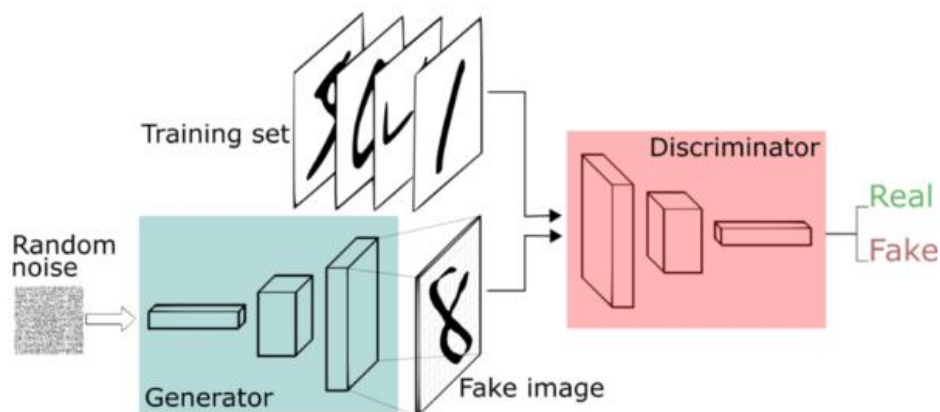
Kratika GAN pomeni "Generative Adversarial Network", v slovenščino to prevedemo kot generativna nasprotniška mreža. Metoda uporablja 2 nevronske mreži, ki delujeta ena proti drugi. Iz tega tudi izhaja beseda v imenu metode, in sicer "Adversarial". Nevronske mreži ustvarjata podatke, ki bi lahko veljali za resnične[1].

### 3.2 Generator in diskriminator

Za boljše razumevanje metode bomo navedli konkreten primer. Opisali bomo tudi generator in diskriminator bolj podrobno kot v prejšnjem razdelku. Predpostavimo, da generativni algoritem generira nam nepoznane podatke, za katere ne poznamo niti njihovih značilnosti. Zdaj je naloga diskriminatorja, da nam te podatke kategorizira in posledično označi. Glede na značilnosti podatkov dobimo oznako. Vhodne podatke seveda po potrebi razdelimo na več vrst. Kot primer vzemimo e-pošto. Naloga diskriminatorja je, da nam označi posamezno sporočilo kot *spam* ali pa *ne spam*. Diskriminator vhodne podatke razdeli na besede v sporočilu in pa naslov sporočila (zadeva). Matematično to lahko predstavimo s pomočjo verjetnostnega zapisa.

Oznako označimo z  $z$ , značilnost oz podatke pa z  $x$ . Formula, ki povezuje te dve neznanki je  $p(y|x)$ . Formulo ubesedimo kot "verjetnost  $y$  pri  $x$ ", v našem konkretnem primeru bi se to glasilo kot "verjetnost, da je sporočilo spam glede na besede v sporočilu".

Prišli smo do zaključka, da diskriminator glede na značilnosti podatkov le-te označi. Zdaj se vrnemo k generatorju. Njega si lahko predstavljamo, da deluje obratno. Torej, da iz označbe dobi značilnosti. To pomeni, da bo zgeneriral podatke z določenimi značilnostmi, ki veljajo za neko podano označbo. Imamo še en način, kako lahko opišemo generator in diskriminator. Diskriminator pozna meje med razredi, generator pa pozna porazdelitev vsakega razreda posebej. V nadaljevanju sta dve sliki, ki opisano prikažeta grafično[4].



Slika 3.1: Delovanje generatorja in diskriminatorja - prvi način[5]

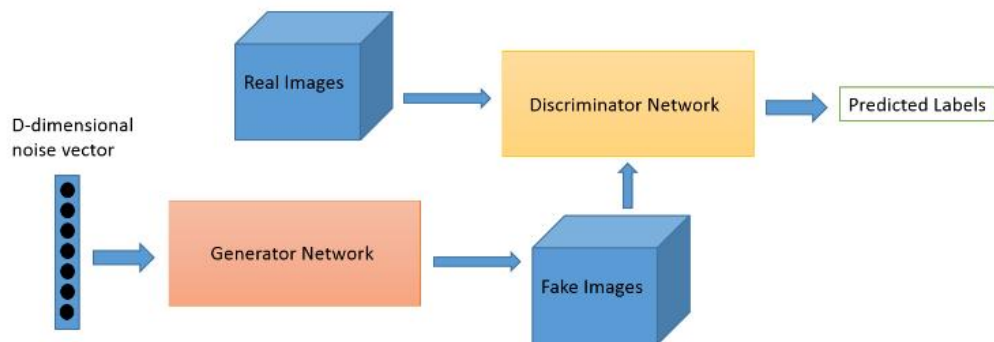
Slika 3.1 prikazuje delovanje generatorja in diskriminatorja. V diskriminator vodita dve puščici, to pomeni, da dobi dvojne vhodne podatke. Na začetku so vhodni podatki, ki pridejo v diskriminator "trening množica". To je množica resničnih podatkov, na podlagi katerih bo funkcija zgradila model. Ta model bo v prihodnosti izbral, katere slike so resnične, katere pa ne. Naš cilj bo poslati v ta model neresnično sliko, ki jo bo model označil kot resnično.

Druga puščica, ki pride v diskriminator, so neresnične slike, zgenerirane

z generatorjem. Naloga diskriminatorja je, da prepozna da so te slike neresnične. Imeti moramo dober generator, ki bo neresnične slike približal resničnim dovolj, da jih bo diskriminator glede na prej generirani model označil za resnične.

Zelo pomembna je izbira trening množice, saj je celoten sistem odvisen od modela, ki nastane. Izbrati moramo primerne podatke glede na naš cilj, ker lahko ob vmesnih napakah ugotovimo, da je za neprimerne rezultate kriv prav ta model, ki nam je bila prva točka v našem postopku[5].

Mogoče še enostavnejša predstava je lahko vidna na naslednji sliki. Razberemo lahko, da diskriminator dobljene podatke označi. Obdela resnične slike in neresnične slike ter jih primerno označi.



Slika 3.2: Delovanje generatorja in diskriminatorja - drugi način[4]

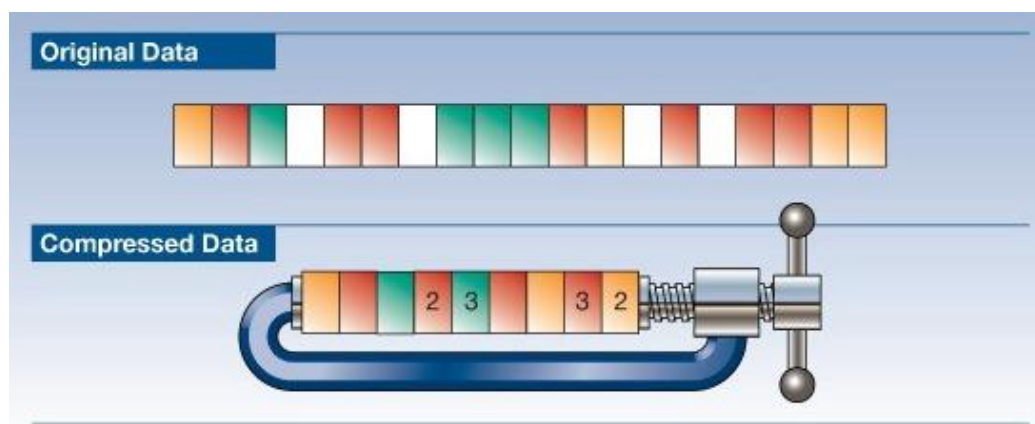
### 3.3 Predstavitev slike s številkami

Iz predhodnih odsekov dokumenta je razvidno, da bomo pri našem programu operirali s slikami. Slike moramo predstaviti vektorsko, torej s številkami, saj generator lahko za vhodne podatke vzame vektor točno določenih dimenzij.

### 3.4 Latentna predstavitev slik

Problem, ki se tu pojavi, je količina podatkov, ki jih posamezna slika vsebuje. Zmanjšati moramo dimenzije, ki jih porabimo za predstavitev slike. Dimenzija novo nastalega vektorja mora biti občutno manjša. Vektor z zmanjšanimi dimenzijami, ki ga dobimo, se imenuje latentna predstavitev slike. Na sliki primer, kaj sploh pomeni zmanjšati dimenzije nečesa. Devetnajst točk skompresiramo tako, da jih lahko prestavimo samo z devetimi[10].



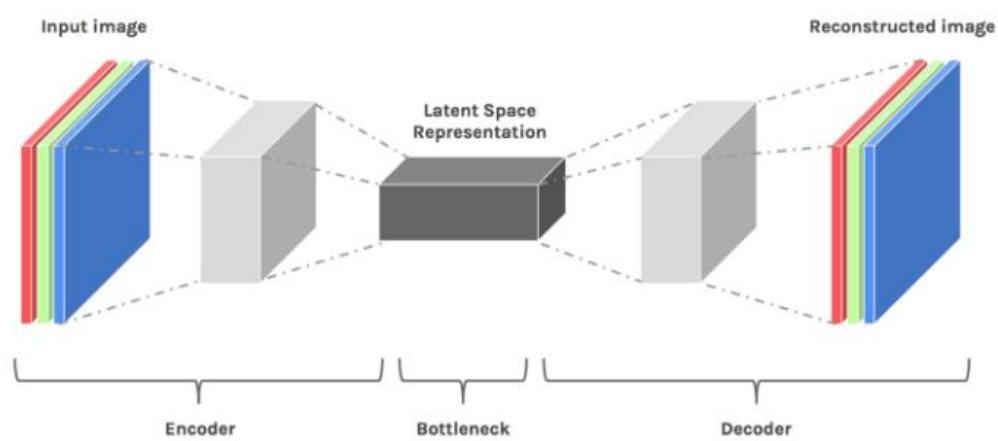


Slika 3.3: Primer kompresije vektorja[10]

### 3.5 Latentni prostor

Torej imamo za vsako sliko, ki jo bomo uporabili v našem generatorju, latentno predstavitev v obliki vektorja. Vsi ti vektorji sestavljajo oziroma oblikujejo prostor, ki mu rečemo latenten prostor. V tem prostoru so slike, torej njihovi vektorji, ki so si podobne bolj skupaj, slike, ki si pa niso, pa so bolj narazen. Dimenzija latentnega prostora je določena z velikostjo vektorja. Na primer, če imajo vektorji dolžino 3, to pomeni, da je dimenzija latentnega prostora 3. Na sliki imamo prikazano vizualizacijo latentnega prostora. Slike, ki jih skompresiramo, sestavljajo latenten prostor tako, da so ohranjeni samo pomembni podatki. To je tudi osnova, da sploh lahko kompresiramo. Seveda, da lahko iz latentnega prostora dekompresiramo kakšno sliko, potrebujemo dekodler, ki je to sposoben narediti[6][9].

Ime *latent space* oziroma latenten prostor izhaja iz besede *hidden* oziroma skrit. To pa zato, ker si težko predstavljamo prostor, ki ima kakšno dimenzijo več kot 3, kaj šele prostore dimenzije reda 100[9][10].



Slika 3.4: Vizualizacija latentnega prostora[10]

## Poglavje 4

# Izdelava programa

### 4.1 Začetna koda

Po prebiranju različnih člankov iz spleta in znanstvenih revij o GAN metodi in izdelovanju rešitev z njo sem dobil dovolj znanja, da sem začel praktično razmišljati o problemu. Začetni del kode je pripravil Matjaž Mav[3]. Moja naloga je bila že narejeno kodo dopolniti do tega, da bom dobil rezultate, ki se tičejo mojega problema.

Celotna koda je napisana v programskem jeziku python. Razlog za to so python knjižnice, ki se uporabljajo za strojno učenje (ang. Machine learning). Ena izmed najbolj pomembnih je *torch*.

### 4.2 Slike za učenje in testiranje

Skozi celoten proces izdelovanja programa in seveda tudi na koncu smo potrebovali slike obrazov. Delimo jih v 2 kategoriji: učna množica in testna množica. Ti dve množici sta bili vnaprej pripravljeni s strani že prej omenjenega Matjaža Mava[3]. Za učenje je bilo uporabljenih 217 primerov, za testiranje pa 20[8]. Za vsakega posameznika smo imeli več slik, ki so prikazovale njegovo razpoloženje. Kot primer: sreča, žalost, strah, presenečenje, začudenje.

Zraven razpoloženj je bil tudi del slik, ki so predstavljale isto razpoloženje, ampak drugačno starost posameznika ali pa kot slike. Število razpoloženj je bilo pri vsakemu različno, od 3 pa vse do 8. Z uporabo več slik posameznika smo dobili nekakšno povprečje njegovega obraza. Kategorije slik posameznikov so bile razdeljene na: moški, ženska, otrok.

Na sliki primer trinajstih slik posameznika, iz katerih smo dobili sliko, ki smo jo uporabljali v nadaljevanju za to osebo.



Slika 4.1: 13 slik osebe, ki se razlikujejo po starosti in razpoloženju[8]

Naslednja slika prikazuje sliko, ki je nastala iz predhodnih trinajstih slik.



Slika 4.2: Končna slika

Spomnimo se naše naloge, in sicer *slika obraza otroka glede na starše*. Torej potrebujemo trojko slik, mamo, očeta in otroka, ki jo bomo uporabili kot učno množico. V vnaprej pripravljeni bazi slik imamo takih trojk 207.

V prejšnjem poglavju smo govorili o latentni predstavitvi slik, zakaj jo potrebujemo in kako je predstavljena. Seveda rabimo latentne zapise vseh pripravljenih slik, da lahko sploh operiramo s slikami. Latente slike so bile tudi že vnaprej pripravljene, in sicer v laboratoriju za računalniški vid na Fakulteti za računalništvo in informatiko .

### 4.3 Koda

Že v prejšnjih poglavjih smo omenili, da so slike že pripravljene. Prvi korak je bil, da smo slike razporedili v trojke oče, mama in otrok. To smo naredili s pomočjo datoteke, ki je vključevala ID-je trojk.

Za nas so bili pomembni trije ID-ji, ki so predstavljali po vrsti iz leve očeta, mamo in otroka. Kot vidimo na slikah imamo veliko duplikatov, ki

```
f_pid,f_iid,m_pid,m_iid,c_pid,c_iid
984,6702,276,332,529,347
984,6702,276,332,529,348
984,6702,276,332,529,349
984,6702,276,332,529,350
984,6702,276,332,529,351
984,6702,276,332,529,352
984,6702,276,332,529,353
984,6702,276,332,529,354
984,6702,276,332,529,355
984,6702,276,332,529,356
984,6702,276,332,529,357
984,6702,276,332,529,358
984,6702,276,332,529,359
```

(a) naključen del datoteke

```
4,4889,441,595,521,558
4,4889,441,595,521,559
4,4889,441,595,521,561
4,4889,441,595,521,562
4,4889,441,595,521,564
4,4889,441,595,521,566
4,4889,441,595,521,568
641,798,672,817,207,834
641,798,672,817,207,835
641,798,672,817,207,836
641,798,672,817,207,837
```

(b) naključen del datoteke

Slika 4.3: ID očeta, mame, njihovega otroka in še 3 za nas nepomembni ID-ji

smo jih ustrezno zanemarili. Sedaj imamo pripravljene trojke od katerih potrebujemo latentne slike. Na naslednji sliki je prikaz priprave teh podatkov.

```
for i, pid in enumerate(pids):
    pid = int(pid)
    ws = []
    ww=0
    j=0
    for npz in glob(f"{self.root}/{pid}/*.npz"):
        w = np.load(npz)['w']
        if np.isnan(w).any() or np.isinf(w).any():
            w = np.nan_to_num(w)
        ww=ww+w
        j=j+1
    ww=ww/j
    ww=torch.tensor(ww)
    self.pid_map[pid] = i
    self.data.append(ww)
self.data = torch.stack(self.data, dim=0)
```

Slika 4.4: priprava latentnih slik vseh trojk

Podatke imamo pripravljene za uporabo. V naslednjem koraku jih ustrezno uredimo in jih damo v napovedovalni model. V našem primeru je to Ridge regression.

```
def load_data(split="train"):
    X = []
    y = []
    i=0
    for father, mother, child, *_ in NokMeanDataset(split=split):
        input = torch.cat([father.flatten(0), mother.flatten(0)], dim=0)
        output = child.flatten(0)
        X.append(input)
        y.append(output)
        i=i+1
    X = torch.stack(X, dim=0)
    y = torch.stack(y, dim=0)
    return X, y
```

Slika 4.5: priprava podatkov za učenje ali testiranje

```
X_train, y_train = load_data(split="train")
X_test, y_test = load_data(split="test")
regressor = Ridge(normalize=True)
regressor.fit(X_train, y_train)

y_train_hat = regressor.predict(X_train)
y_test_hat = regressor.predict(X_test)

mse_train = mean_squared_error(y_train, y_train_hat)
mse_test = mean_squared_error(y_test, y_test_hat)
print(mse_train, mse_test)

images = []
for i in range(y_test.shape[0]):
    images.append(X_test[i,:18*512].view(18, 512))
    images.append(X_test[i,18*512:].view(18, 512))
    images.append(y_test[i].view(18, 512))
    images.append(torch.from_numpy(y_test_hat[i]).to(torch.float32).view(18, 512))

images = torch.stack(images, dim=0)
```

Slika 4.6: napovedovanje podatkov

Glejmo sliko 6.4. V prvih štirih vrsticah uporabimo funkcijo definirano na sliki 6.3, da pripravimo ustrezne podatke za treniranje in testiranje ter definiramo regresijo in v njo vstavimo podatke. Model je naredil svoje, zato le še pravilno razporedimo podatke. V for zanki popolnujemo tabelo v vrstnem redu oče, mama, otrok in generiran otrok. Še zadnji korak, da dobimo dejanske slike je generiranje slik iz latentnih slik. To mi je omogočila grafična kartica NVIDIA GeForce GTX 970M. Hitrost je 924 MHz, velikost spomina pa 3GB.





## Poglavje 5

### Rezultati in analiza

#### 5.1 Primerjava-prvi del



Slika 5.1: 4 primeri slik staršev, njihovega otroka in pa generiranega otroka[8]

Gledamo sliko 7.1, vsaka vrstica je svoj primer. Prvi obraz je oče, drugi mama, tretji njun otrok, četrti pa generiran otrok z našim programom. Kot že prej omenjeno, bodo izprašani v tem delu, torej na štirih primerih, poskušali

analizirati slike v podrobnosti. To pomeni, da bodo podali mnenje o vsakemu primeru, torej vrstici, posebej in ga utemeljili.

### 5.1.1 Mnenja izprašanih ljudi

1. Na splošno so generirani otroci zelo podobni dejanskim otrokom in bi jih zlahka zamenjali za njihove sorojence. Zdi se tudi kot da mama bolj vpliva na generiranega otroka (barva oči in oblika/položaj ustnic je skoraj enaka kot pri mami). Barva las izgleda, kot da je mešanica vseh drugih treh. Miha, 23.
2. Podobnost staršev in njihovih generiranih otrok se po mojem mnenju najbolj odraža v glavnih obraznih potezah. Takoj sem na primer opazila podobnost v obliki nosu in nasmeha. Če primerjam obraze otrok, ki jih je napovedal model, z obrazi realnih otrok, je po mojem mnenju podobnost za večino primerov dokaj velika. Največjo razliko opazim v sami obliki obraza, predvsem v delu lic in obliki brade. Nina, 21.
3. V prvi vrsti je pravi sin čisto podoben očetu, tudi generirana slika kaže izredno podobnost med njima. V drugi vrsti je hči podobna mami, med sabo si hči in generirana slika nista podobni, je pa generirana slika na hitro podobna očetu, imata enak nasmeh. Tretja vrsta: hči podobna mami in največjo podobnost vidim ravno v tej vrsti. Četrta vrsta: sin podoben očetu, generirana slika me spominja na mamo. Zaključno mnenje ja, da so računalniške slike dobro zadete. Pri nekaterih primerih je podobnost res izjemna. Andreja, 45
4. Vsi obrazi generiranih otrok so v primerjavi z obrazi staršev bolj podobni staršem kot njihovi resnični otroci. Pri medsebojni primerjavi otrok pa bi ob videnju obraza v resničnosti zaradi izredne podobnosti otroke prepoznaval kot brate ali sestre, saj so fotografije generiranih otrok videti izredno podobne z obrazi resničnih otrok. V prvi vrstici primerjave opazim podobnost oblike obraza med mamo in resničnim

otrokom, medtem ko je generiran otrok po obliki obraza zelo podoben očetu. V drugi primerjavi, kjer gre za ženskega otroka, pa opazim podobnost z mamo. Vidim, da je generiran otrok bolj podoben mami, saj večjo podobnost poudarjata bolj izraziti ličnici. Enaka primerjava in podobnost velja tudi za tretjevrstično primerjavo. Pri četrti primerjavi, kjer pa gre za moškega otroka, pa opazim podobnost resničnega otroka z materjo, medtem ko je generiran otrok bolj podoben očetu. Stanko, 51

5. V prvi vrsti je generiran otrok podoben mami, oči ima pa iste kot brat. V drugi vrsti je generiran otrok bolj podoben hčeri vendar ima mamina lica. V tretji vrsti ne vidim podobnosti med družino in generiranim otrokom. V četrti vrsti vidim največjo podobnost med vsemi primeri, saj ima oči od očeta, obraz od mame in od sina ima uho. Matic, 21.
6. Generirani otroci so v veliki meri podobni svojim staršem. Zdi se, kot da so potomci ženskega spola generirani bolj po podobi svoje matere, potomci moškega spola pa bolj zrcalijo izgled svojega očeta. Pri tem je največja podobnost izražena predvsem v obliku nosu in ust. Podobnost generiranih otrok in realnih otrok je razmeroma velika in bi jih lahko označili za sorojence. Nekoliko odstopajo v obliki obraza v predelu ličnic ter višini čela. Anonimno, 40-50.

### 5.1.2 Moje osebno mnenje

Za začetek moram reči, da sem zelo pozitivno presenečen nad podobnostjo resničnih in generiranih otrok, ker ko sem dokončno potrdil delo na tej temi, si nisem predstavljal, da je možno doseči takšno podobnost. Glede na to, da so generirani otroci zgrajeni le na podlagi izgleda staršev, ne pa tudi genskega zapisa, so v večini bolj podobni samim staršem kot njihovi resnični otroci. Ne najbolj optimalna rešitev, ampak za začetek se mi zdi, da dovolj solidna.

Na prvi pogled padeta v oči prva in tretja vrstica. Resničen in generiran otrok sta si tako zelo podobna, da sem prepričan, da bi ju večina opazovalcev

videla kot sorojenca. Gledamo prvo in četrto vrstico generiranega otroka. Zelo sta si podobna. Najverjetneje tukaj igra vlogo to, da imata podobni materi. Drugih specifičnih podobnosti ne opazim, zato bom zdaj podal še splošno mnenje.

Na prvi pogled zgleda, kot da je generiran otrok podoben staršu istega spola, ampak že po prej analiziranem primeru, ko smo rekli, da imata primerka podobni materi, se mi zdi, da to ni čisto res. Predvidevam, da me k temu zavede dolžina las in zato otrok zgleda podoben staršu istega spola. Opazka, ki nima povezave s podobnostjo: rotacija glave je pri generiranih otrocih takšna, kot je pri starših, kar ne moremo reči za resnične otroke. Pri ustih, nosu in celotnem obrazu ne vidim nekih specifičnih podobnosti enemu ali drugemu staršu. Zdi se mi pa, da so generirani otroci zelo podobni tudi resničnim otrokom, čeprav ti niso bili uporabljeni pri generiranju. S tem dobimo potrditev, da je resničen otrok podedoval veliko videza od staršev, najverjetneje več kot od starih staršev, ki niso uporabljeni v modelu.

## 5.2 Primerjava-drugi del

V tem delu je pa navodilo izprašanim popolnoma drugačno kot v prvem delu. Mnenje morajo podati čim bolj splošno brez opisovanja posameznih primerov v podrobnosti. Zato imamo tukaj tudi več primerov, da je lažje narediti splošen opis. Primeri so na strani 23 in 24, kjer imamo po 8 primerov na vsaki sliki.

### 5.2.1 Mnenja izprašanih ljudi

1. Vse generirane slike imajo do potankosti urejene frizure in zaprta usta, imajo tudi resne obraze. Kljub temu sem opazila podobnost med otrokom. Pri generiranih slikah vidim, da so v večini podobne materam. Opazila sem tudi, da so si generirani obrazi zelo podobni med sabo.

Andreja, 45

2. Pri veliko izmed primerov se po mojem mnenju generiran potomec in realni otrok precej razlikujeta in ju na prvi pogled ne bi ocenila kot sorojenca. V teh primerih je tudi podobnost s starši manjša. Pri primerih, kjer opazim večjo podobnost generiranih otrok s starši, je tudi ujemanje s sorojenci večje. Podobnosti opazim predvsem v obliki nosu, ličnic in ustnic. Nina, 21.
3. Vidim veliko podobnosti med osebami. Najbolj sta si podobni Mama in generirana oseba, saj se to vidi pri večini družin. Očetom pa generirani otroci niso podobni, oziroma so jim podobni zelo malo in jih nebi kategoriziral kot sorodnike. Matic, 20
4. Pri ogledu drugega sklopa, skupaj večjega števila fotografij, opazim, da zadnji stolpec fotografij izstopa. Opaziti je, da bi fotografije lahko bile generiran izdelek, saj imajo osebe na fotografijah zelo podobne frizure. Pri vseh frizurah so lasje odmaknjeni izza ušesa. Vse osebe izkazujejo rahel nasmešek, enake oblike brez, da bi kateri koli nasmeh razprl ustnice. Poleg tega pa so ti generirani otroci v zelo veliki meri podobni staršem in tudi njihovimi biološkimi otrokom, kar je po eni strani smiselno, čeprav realni otroci niso bili uporabljeni za generacijo generiranih. Stanko, 51

### 5.2.2 Moje osebno mnenje

V tem poglavju bom poskušal slike ovrednotiti kot celoto brez izpostavljanja podrobnosti pri posameznem primeru, saj je bilo to narejeno že v prejšnjem razdelku. Če gledamo 4. stolpec, torej generirane otroke in jih primerjamo med sabo, nas obrazi nekaterih od njih spominjajo na isto osebo. Zdi se mi, da je podobnost največja pri obliki brade in pri očeh. Nekateri obrazi moškega spola imajo dolge lase, kar ni nič narobe, ampak te lasje ponekod zgledajo kot, da so ženski.

Premaknemo se na primerjavo obraza generiranega in resničnega otroka. Starost se pri večini ujema, imamo pa 1 primer, kjer je malce težje narediti

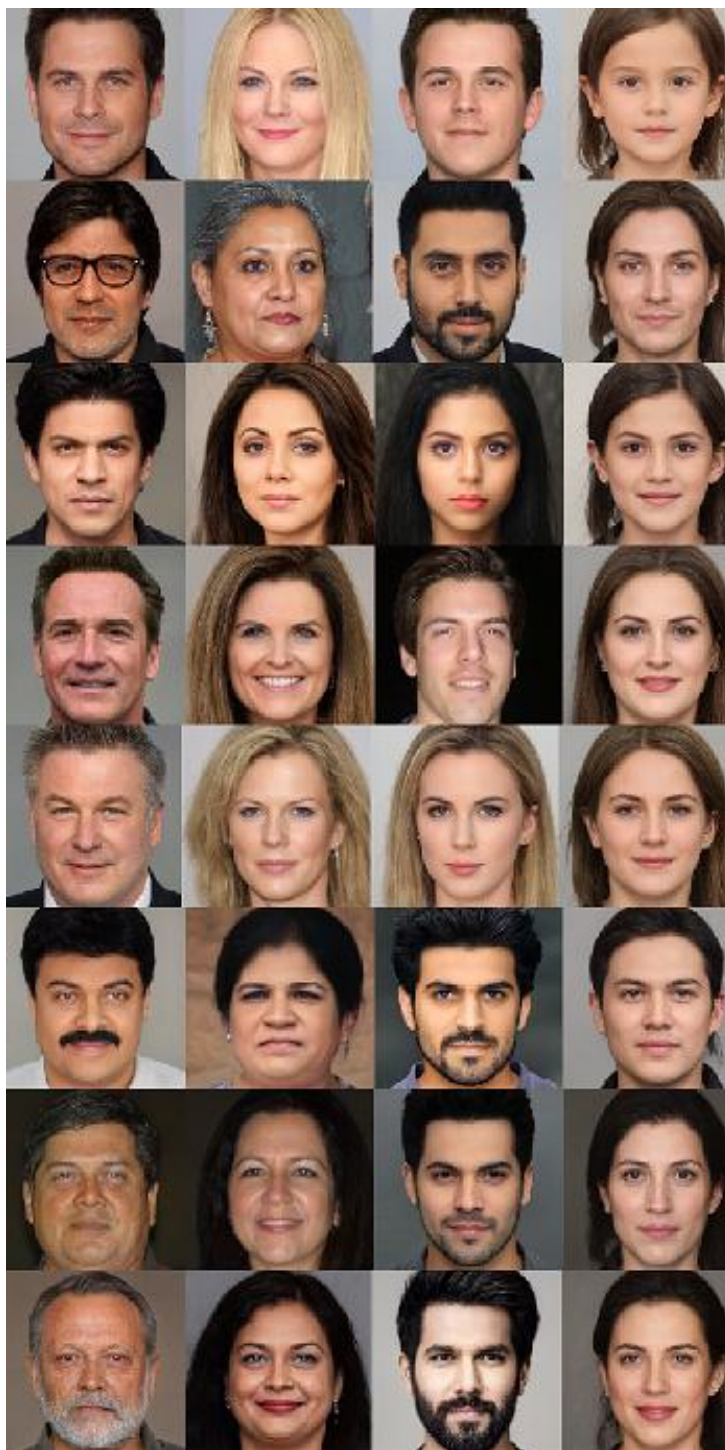
primerjavo zaradi prevelike razlike v starosti, je pa podobnost kljub temu vidna. V prejšnjem razdelku je bilo nekako vidno, da je otrok moškega spola podoben ponavadi mami, otrok ženskega spola pa očetu. Zdaj, ko pregledujemo več slik, sem dobil občutek, da je veliko generiranih otrok bolj podobnih mami ne glede na spol.

Kot sem že povedal v prejšnji primerjavi, so generirani otroci bolj podobni staršem, kot pa resničnemu otroku, kar je logično saj smo uporabili samo slike staršev ne pa njihovega DNA-ja. Zato bi generirane otroke po večini pripisal njihovim staršem, ne pa vedno označil, kot brata ali sestro resničnemu otroku.

### 5.3 Diskusija

Za zaključek analize bi naredil še primerjavo ljudi, ki so podali svoje mnenje. Njihova starost je variirala od 20 do 50 let. Ob prebiranju mnenj sem ugotovil, da imajo različne starostne skupine drugačno razmišljanje in, da drugače opazajo podrobnosti in celostno sliko. Mlajša starostna skupina, torej okrog 20 let, ni opazila manjših podrobnosti, je pa mnenje, ki je bilo podano, bilo zelo razumljivo in smiselno sestavljeno. Starejša starostna skupina, 40-50 let, pa je pokazala čisto nasprotje. Opazili so veliko manjših podrobnosti, ampak je pa končno mnenje bilo napisano zelo zmedeno.

Z uporabo različnih starostnih skupin sem dobil različna mnenja, kar je pa bil tudi cilj, dobiti čim več različnih opažanj.



Slika 5.2: prvih 8 primerov slik staršev, njihovega otroka in pa generiranega otroka[8]



Slika 5.3: drugih 8 primerov slik staršev, njihovega otroka in pa generiranega otroka[8]



## Poglavje 6

# Sklepne ugotovitve

### 6.1 Zaključek

S končnimi rezultati sem zelo zadovoljen, ker si na začetku nisem predstavljal, da je možno doseči podobnost v tolikšni meri. Dobre rezultate potrjujejo tudi mnenja zunanjih opazovalcev, ki so predstavljena v prejšnjem poglavju. Izprašani dajejo občutek, da so zelo presenečeni nad podobnostjo, kar si lahko tudi preberemo. Poleg mnenj v pisni obliki sem z njimi tudi govoril, tam pa je bila presenečenost še bolj izražena. Predvidevam, da je presenečenost posledica tega, da nimajo predznanj o tej temi in se jim zdi vsaka podobnost izjemna.

Kot že prej omenjeno, v prejšnjih poglavjih je generacija otrok temeljila zgolj na izgledu staršev. To je najverjetneje tudi razlog, da je generiran otrok bolj podoben staršem kot pa njihov dejanski otrok. Ta otrok je dobil del DNA-ja tudi od starih staršev in mogoče še njihovih staršev, generiran pa je zgrajen le iz njegovih staršev. Link do repozitorija za nadaljevanje dela na projektu[2].

## 6.2 Možne izboljšave

Za nadaljnje delo bi predlagal, da se za generacijo otrok ne uporabi enak delež očeta in matere. Predvidevam, da bi takšni rezultati bili še bolj zanimivi, ker bi bil otrok moškega spola še bolj podoben očetu in bi lahko iskali po obrazu, kaj je pa tisto, kar je podedoval po mami. Glede na to, da je generiranje temeljilo le na ozgledu staršev bi bilo smiselno v generiranje zraven staršev vključiti tudi stare starše, amapk seveda z manjšim deležom.

# Literatura

- [1] Ian J. Goodfellow in sod. *Generative Adversarial Networks*. 2014. DOI: 10.48550/ARXIV.1406.2661. URL: <https://arxiv.org/abs/1406.2661>.
- [2] Matjaž Mav. *Link do repozitorija za nadaljnje delo*. URL: [https://drive.google.com/drive/folders/1hm4QF9BkXA556Y0oP6A0G8-Ud\\_kV4h9b](https://drive.google.com/drive/folders/1hm4QF9BkXA556Y0oP6A0G8-Ud_kV4h9b).
- [3] Matjaž Mav. *Link do začetnega repozitorija*. URL: [https://drive.google.com/drive/folders/1ljkfBx1eSvu93r-0-whZJrdl2\\_OuUzB](https://drive.google.com/drive/folders/1ljkfBx1eSvu93r-0-whZJrdl2_OuUzB).
- [4] Chris Nicholson. *A Beginner's Guide to Generative Adversarial Networks (GANs)*. 2020. URL: <https://wiki.pathmind.com/generative-adversarial-network-gan> (pridobljeno 27. 10. 2022).
- [5] Chris Nicholson. *Keywords to know before you start reading papers on GANs*. 2021. URL: <https://towardsdatascience.com/keywords-to-know-before-you-start-reading-papers-on-gans-8a08a665b40c> (pridobljeno 29. 10. 2022).
- [6] Manning publications. *concept latent space in category GAN*. 2022. URL: <https://livebook.manning.com/concept/gan/latent-space> (pridobljeno 29. 10. 2022).
- [7] Pooja Rani. "Emotion Detection of Autistic Children Using Image Processing". V: *2019 Fifth International Conference on Image Infor-*

- mation Processing (ICIIP)*. Nov. 2019, str. 532–535. DOI: 10.1109/ICIIP47207.2019.8985706.
- [8] Joseph P Robinson in sod. “Families in Wild Multimedia: A Multimodal Database for Recognizing Kinship”. V: *IEEE Transactions on Multimedia* 24 (2021), str. 3582–3594.
- [9] Thalles Silva. *An intuitive introduction to Generative Adversarial Networks (GANs)*. 2018. URL: <https://www.freecodecamp.org/news/an-intuitive-introduction-to-generative-adversarial-networks-gans-7a2264a81394> (pridobljeno 29.10.2022).
- [10] Ekin Tiu. *Understanding Latent Space in Machine Learning*. 2020. URL: <https://towardsdatascience.com/understanding-latent-space-in-machine-learning-de5a7c687d8d> (pridobljeno 29.10.2022).
- [11] Xin Yi, Ekta Walia in Paul Babyn. “Generative adversarial network in medical imaging: A review”. V: *Medical Image Analysis* 58 (2019), str. 101552. ISSN: 1361-8415. DOI: <https://doi.org/10.1016/j.media.2019.101552>. URL: <https://www.sciencedirect.com/science/article/pii/S1361841518308430>.