Automatizirano generiranje setova portreta primjenom Stable Diffusion biblioteke

Šrempf, Michael

Master's thesis / Diplomski rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:224947

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-10**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni studij

Automatizirano generiranje setova portreta primjenom Stable Diffusion biblioteke

Diplomski rad

Michael Šrempf

Osijek, 2023.

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac D1: Obrazac za imenovanje Povjerenstva za diplomski ispit

Osijek, 18.09.2023.

Odboru za završne i diplomske ispite

Imenovanje Povjerenstva za diplomski ispit

Ime i prezime Pristupnika:	Michael Šrempf
Studij, smjer:	Diplomski sveučilišni studij Računarstvo
Mat. br. Pristupnika, godina upisa:	D-1251R, 07.10.2021.
OIB studenta:	48434917301
Mentor:	izv. prof. dr. sc. Časlav Livada
Sumentor:	,
Sumentor iz tvrtke:	
Predsjednik Povjerenstva:	prof. dr. sc. Krešimir Nenadić
Član Povjerenstva 1:	izv. prof. dr. sc. Časlav Livada
Član Povjerenstva 2:	Robert Šojo, mag. ing. comp.
Naslov diplomskog rada:	Automatizirano generiranje setova portreta primjenom Stable Diffusion biblioteke
Znanstvena grana diplomskog rada:	**Obradba informacija (zn. polje računarstvo)**
Zadatak diplomskog rada:	U radu je potrebno napraviti aplikaciju koja će primjenom Stable Diffusion biblioteke omogućiti generiranje setova portreta za potrebe treniranja modela strojnog učenja.
Prijedlog ocjene pismenog dijela ispita (diplomskog rada):	Vrlo dobar (4)
Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 2 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 2 razina
Datum prijedloga ocjene od strane mentora:	18.09.2023.

Potvrda mentora o predaji konačne verzije rada:	*Mentor elektronički potpisao predaju konačne verzije.*
	Datum:

FERIT
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

IZJAVA O ORIGINALNOSTI RADA

Osijek, 05.10.2023.

Ime i prezime studenta:	Michael Šrempf
Studij:	Diplomski sveučilišni studij Računarstvo
Mat. br. studenta, godina upisa:	D-1251R, 07.10.2021.
Turnitin podudaranje [%]:	13

Ovom izjavom izjavljujem da je rad pod nazivom**: Automatizirano generiranje setova portreta primjenom Stable Diffusion biblioteke**

izrađen pod vodstvom mentora izv. prof. dr. sc. Časlav Livada

i sumentora ,

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.
Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD 1

1.1. Zadatak rada 1

2. STROJNO UČENJE 2

2.1. Područja primjene strojnog učenja 3

2.2. Generativni modeli za generiranje slika 4

3. PROCES GENERIRANJA SLIKE STABLE DIFFUSIONOM 7

3.1. Treniranje difuzijskog modela 7

3.2. Generiranje slike pomoću difuzijskog modela 8

3.3. Generiranje slike sa Stable Diffusionom 9

4. IZRADA APLIKACIJE 11

4.1. Korišteni alati i programska okruženja 11

4.2. Dizajn i arhitektura 12

4.3. Implementacija grafičkog sučelja 12

4.4. Implementacija klase za generiranje slika 16

5. TESTIRANJE I REZULTATI 17

5.1. Različiti modeli 17

5.2. Različiti uzorkivači 18

5.3. Utjecaj veličine početne slike na generiranu sliku 21

5.4. Razvoj slike po koracima 23

5.5. Utjecaj ljestvice navođenja 25

5.6. Utjecaj sjemena 27

6. ZAKLJUČAK 29

LITERATURA 30

SAŽETAK 31

ABSTRACT 32

1. UVOD

Generativni modeli predstavljaju značajan napredak u području umjetne inteligencije omogućavajući računalima da generiraju nove podatke na temelju skupa podataka na kojemu su trenirani. Takvi modeli se danas koriste za generiranje glazbe, teksta, fotografija, videa i mnogih drugih stvari. Jedan od takvih generativnih modela je *Stable Diffusion* koji pripada latentnim modelima.

U ovom diplomskom radu će se napraviti desktop aplikacija koja će upotrebom *Stable Diffusion* biblioteke generirati portret na temelju tekstualnih smjernica i dodatnih parametara.

Pregled sadržaja diplomskog rada: U poglavlju 2. opisano je strojno učenje, područja primjene strojnog učenja i vrste generativnih modela.

U slijedećem poglavlju opisno je treniranje difuzijskog modela i generiranje slike pomoću njega i generiranje slike sa *Stable Diffusionom.*

U 4. poglavlju opisani su korišteni alati i biblioteke, dizajn i arhitektura za izradu aplikacije, te implementacija grafičkog sučelja.

Zatim, u 5. poglavlju napravljena je obrada i analiza rezultata testiranja i opisani utjecaji različitih modela, uzorkivača, početne veličine slike, utjecaj sjemena, ljestvice navođenja na generiranje slike. Prikazan je razvoj slike po koracima.

Šesto poglavlje je zaključak i sažetak.

1.1. Zadatak rada

Zadatak rada je napraviti aplikaciju koja će primjenom *Stable Diffusion* biblioteke omogućiti generiranje setova portreta. Portreti ljudi trebaju biti što realističniji. Program treba biti napisan u Python programskom jeziku. Pomoću *CustomTkinter* biblioteke će se napraviti grafičko sučelje desktop aplikacije, dok će se za upotrebu modela koristiti *Stable Diffusion* biblioteka za Python. Model će se preuzeti s *Hugging Face* web stranice na kojoj ljudi besplatno dijele trenirane modele za *Stable Diffusion*.

2. STROJNO UČENJE

U ovom poglavlju objašnjen je pojam strojno učenje i podjela strojnog učenja, navedena je primjena strojnog učenja, te podjela generativnih modela.

Strojno učenje je grana umjetne inteligencije koja se bavi razvojem algoritama koji omogućuju računalima da na setu podataka prepoznaju obrasce na temelju kojih mogu donijeti zaključke ili predikcije [1].

Strojno učenje može biti nadzirano i nenadzirano. Ono se razlikuje po tome što nadzirano učenje pri treniranju koristi skup podataka koji sadrži parove ulaza i očekivanih izlaza, dok nenadzirano ima samo ulazne podatke.

Nadzirano učenje se nadalje dijeli na generativne i diskriminativne modele. Cilj diskriminativnih modela je naučiti razlike između klasa ili kategorija podataka kako bi mogli ulazne podatke klasificirati na temelju njihovih značajki. Generativni modeli se koriste za modeliranje vjerojatnosti distribucije podataka, a njihov je cilj naučiti generirati nove podatke koji su slični podacima na kojima su trenirani.

Postoje mnogi algoritmi strojnog učenja poput linearne regresije, logistička regresija, K-najbliži susjedi, stablo odluke, neuronske mreže i drugi.

U ovome radu najviše će se govoriti o neuronskim mrežama koje su složeni modeli strojnog učenja inspirirani funkcioniranjem neurona u mozgu. Neuronska mreža uglavnom ima tri glavna sloja:

- Ulazni sloj
- Jedan ili više izlaznih slojeva
- Izlazni sloj

Tijekom treniranja svaki neuron u mreži prima težine koje se prilagođavaju kako bi se postigla željena izlazna vrijednost.

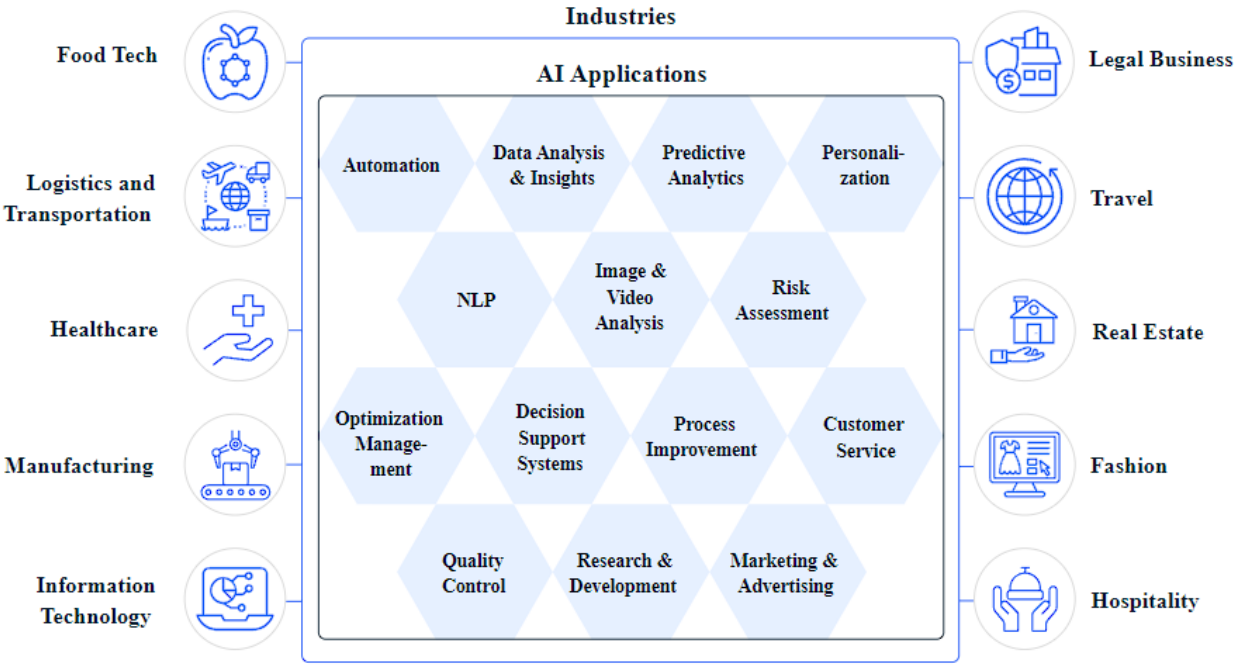
2.1. Područja primjene strojnog učenja

Strojno učenje se već danas koristi svakodnevno za donošenje bitnih odluka u medicini, trgovanju, predviđanju potrošnje energije, mešetarenju dionicama i ostalim sektorima (Slika 2.1.).

Za strojno učenje je potreban jak i brz hardver, ali razvojem tehnologije postaje sve pristupačnije i jeftinije. Već danas je moguće koristiti desktop računala za treniranje i generiranje podataka.

Ova pristupačnost je dovela do raznih inovacija i primjena za razne svrhe poput:

- Prevođenje teksta ili govora u stvarnom vremenu
- Prepoznavanje objekata
- Prepoznavanje i klasifikacija zvukova ili predmeta u videu
- Generiranje slike, videa, pjesme, animacije
- Detekcija tumora i istraživanje lijekova
- Služba za korisnike



Slika 2.1. Primjena umjetne inteligencije u industriji [2]

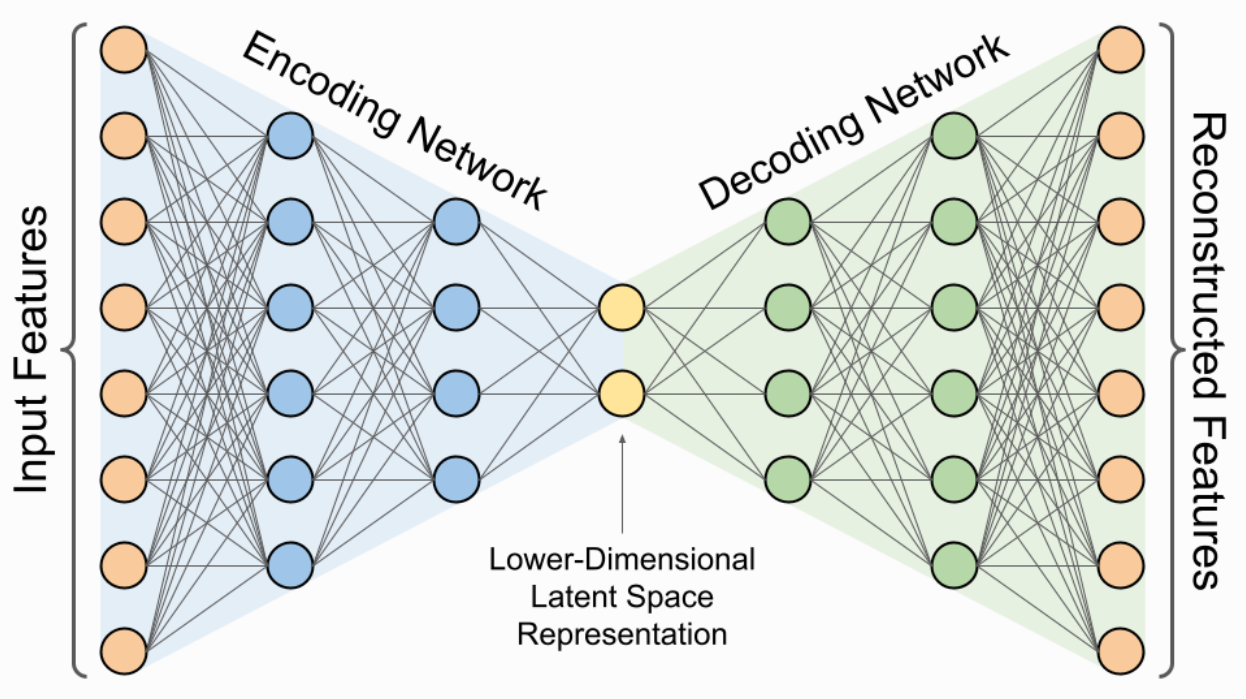
2.2. Generativni modeli za generiranje slika

Postoji tri različite skupine generativnih modela za generiranje slika. To su difuzijski modeli, generativne suparničke mreže (engl. *Generative Adversarial Networks*) i varijacijski autoenkoderi (engl. *Variational Autoencoders - VAE*).

Generativna suparnička mreža [3] je vrsta generativnog modela koji se sastoji od dva suparnička modela, generator i diskriminator. Oni se treniraju na način da se diskriminatoru daje jedna kontrolna slika, pa jedna slika koju je generator generirao. Generator pokušava iz šuma generirati sliku koja će zavarati diskriminator. To znači da je generatoru cilj generirati sliku koja je slična slikama za treniranje kako bi prevarila diskriminator. Kada diskriminator više ne može prepoznati je li generirana slika stvarna ili umjetna može se početi koristiti trenirani model.

Autoenkoder je neuronska mreža koja se koristi za kompresiju i rekonstrukciju podataka. Trenira se na način da višedimenzionalne podatke prebacuje u manje dimenzije nakon čega pokušava rekonstruirati podatak. Na kraju uspoređuje ulazni podatak s rekonstruiranim i podešava neuronsku mrežu. Slika 2.2. prikazuje autoenkoder.

Latentni prostor je smanjena reprezentacija višedimenzionalnog podatka koja sadrži važne karakteristike podatka. Strojno učenje radi s podacima u latentnom prostoru kako bi se smanjili zahtjevi na sklopovlje i ubrzao rad na podacima.



Slika 2.2. *Autoenkoder* [4]

Generiranjem nasumičnog podatka u latentnom prostoru i dekodiranjem istoga može se dobiti slika, ali dobivena slika može biti samo šum. Kako bi se mogao koristiti *autoenkoder* za generiranje slika koristi se varijacijski *autoenkoder* [5] koji proširuje *autoenkoder* dodavanjem probabilističkog aspekta. Uz latentni prostor, varijacijski *autoenkoder* uči distribuciju tih skrivenih varijabli. Odnosno, umjesto jedne točke u latentnom prostoru, varijacijski *autoenkoder* uči razdiobu tih točaka. To se postiže uvjetovanjem latentnih reprezentacija ulaznih podataka na pretpostavljenu distribuciju.

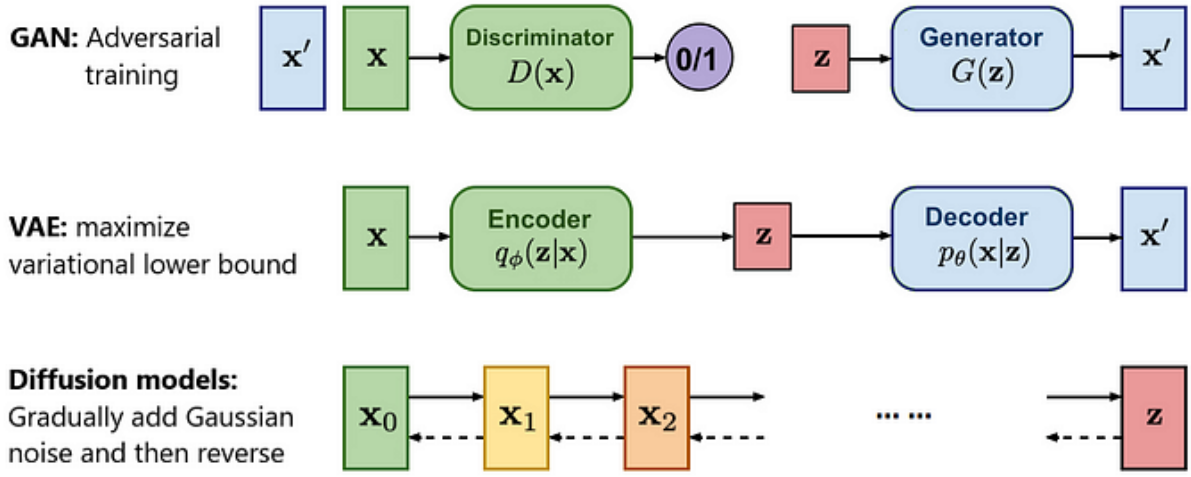
Tijekom treninga, varijacijski *autoenkoder* pokušava optimizirati dvije glavne funkcije gubitka: rekonstrukcijski gubitak i regulacijski gubitak. Rekonstrukcijski gubitak mjeri koliko dobro VAE može rekonstruirati ulazne podatke iz latentnih reprezentacija. Regulacijski gubitak mjeri koliko se latentne reprezentacije podudaraju s pretpostavljenom distribucijom.

Kada je VAE obučen, može se koristiti za generiranje novih primjera podataka tako da se uzorkuju točke iz pretpostavljene distribucije latentnog prostora i dekodiraju pomoću dekodera.

Difuzijski model je inspiriran difuzijom u fizici koja znači miješanje koncentrirane tvari prema nižoj koncentraciji. Difuzija prestaje kada nestane gradijent koncentracije. Ovom analogijom koncentrirana tvar je slika kojoj se dodaje šum sve dok se ne dobije potpuni šum.

Difuzijski model [6] je neuronska mreža koja se trenira pomoću dva procesa, difuzijom prema naprijed i obrnutom difuzijom gdje se od šuma pokušava dobiti početna slika. Prema istraživanjima [7] difuzijski modeli generiraju slike sa superiornijom kvalitetom od ostalih generativnih modela.

Treniranje difuzijskih modela i generiranje slike Stable Diffusionom bit će detaljnije objašnjeno u sljedećem poglavlju.

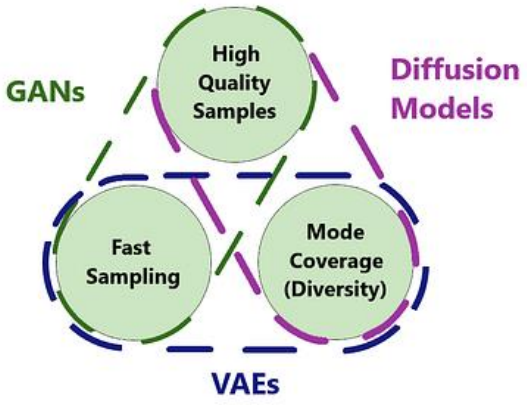


Slika 2.3. Vizualizacija procesa treniranja generativnih modela [8]

Na slici 2.3. prikazana je vizualizacija procesa treniranja generativnih modela, a svaki od ova tri generativna modela ima svoje prednosti i mane [8].

Varijacijski *autoenkoderi* imaju veliku raznolikost, ali zato što predviđa distribuciju latentnog koda mogu postojati slučajevi u kojima se dvije distribucije latentnih kodova preklapaju jedna s drugom. U tome slučaju uzima se prosjek dvaju ulaza što dovodi do zamućenih uzoraka.

Generativne suparničke mreže generiraju vrlo realistične slike, ali čest je problem što generirane slike imaju jako malo raznolikosti. Difuzijski modeli imaju veliku raznolikost i slike su realistične, no njihov nedostatak je što su sporiji od ostalih generativnih modela. Slika 2.4. grafički prikazuje prednosti i mane generativnih modela.



Slika 2.4. Usporedba generativnih modela [8]

3. PROCES GENERIRANJA SLIKE STABLE DIFFUSIONOM

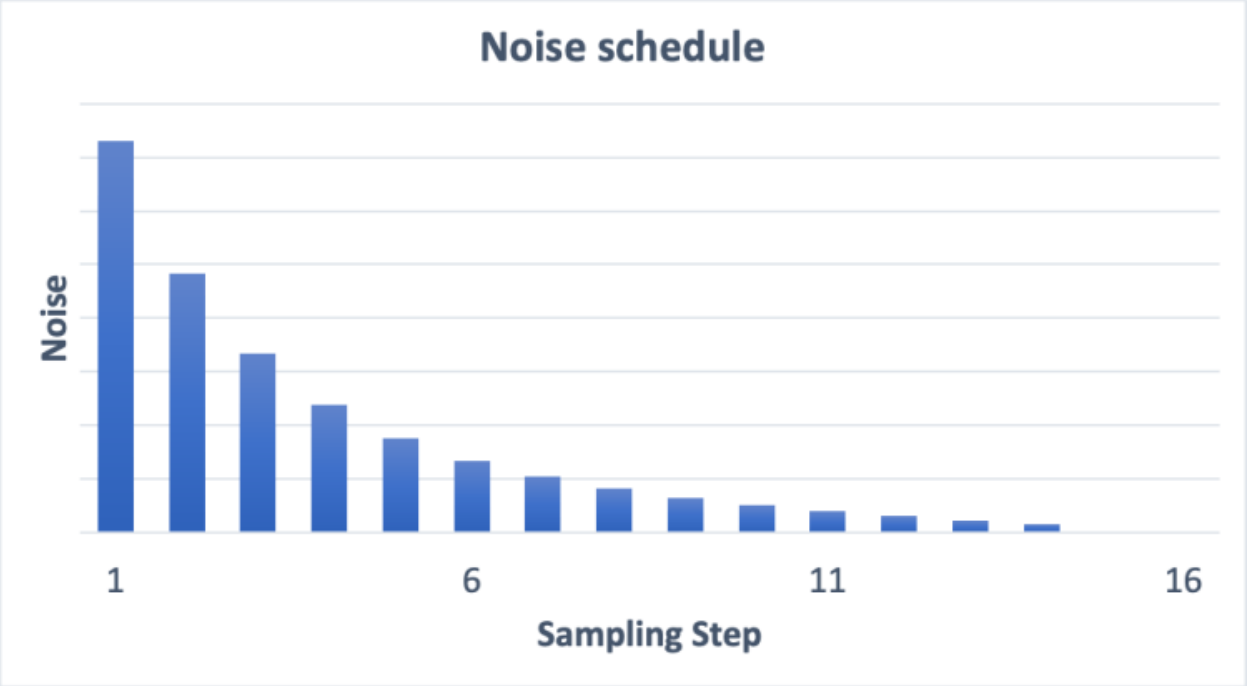
U 3. poglavlju opisano je treniranje difuzijskog modela, upotreba difuzijskog modela za generiranje slike i na koji način *Stable Diffusion* generira slike.

3.1. Treniranje difuzijskog modela

Difuzijski model se trenira procesima difuzije i obrnute difuzije unutar *T* koraka. Broj koraka *T* se bira prije treniranja i on označava unutar kojeg broja koraka će se doći od slike do potpunog šuma.

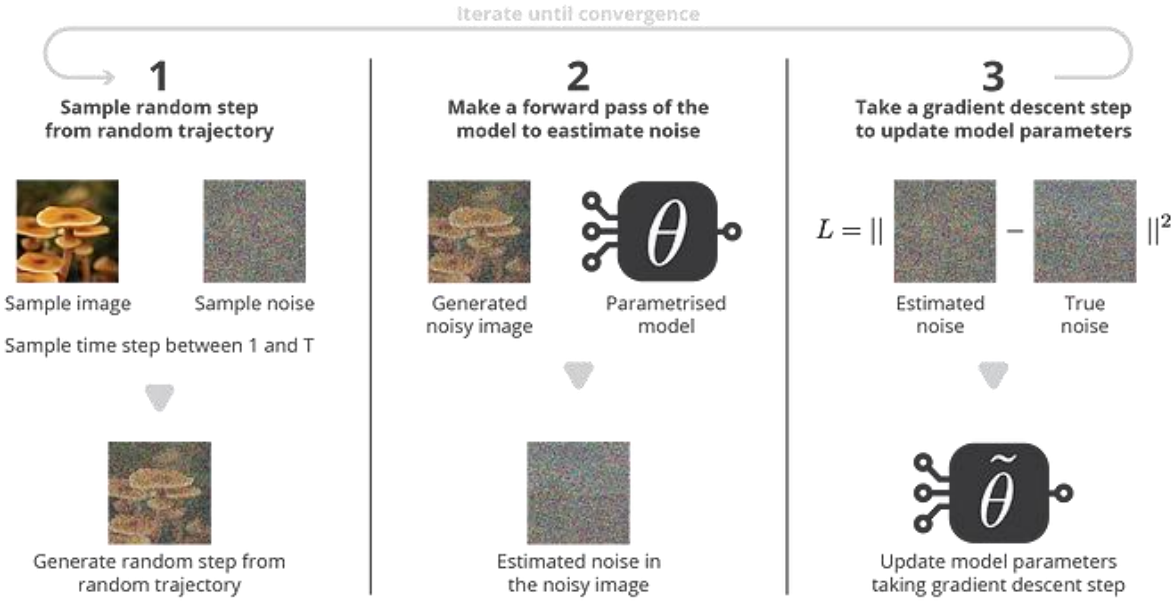
Funkcija kojom se određuje koliko će se dodati šuma se naziv uzorkivač [9] (engl. *sampler*).

U početku se koristilo 1000 koraka u kojima se postupno dodavao i oduzimao šum, takav pristup se zove linearna difuzija. Kako bi smanjili broj koraka počeli su koristiti uzorkivače koji na početku uklanjaju najviše šuma, pa svaki sljedeći korak sve manje i manje kao što je prikazano na slici 3.1. Sada postoji preko 20 uzorkivača.



Slika 3.1. Količina šuma uklonjena svakim korakom [9]

Treniranje neuronske mreže se događa tijekom procesa obrnute difuzije kada neuronska mreža pokušava predvidjeti šum na slici za korak prije. Na primjer na koraku 9 neuronska mreža pokušava predvidjeti koliko je šuma dodano slici na koraku 8 kako bi se dobila slika 9. Nakon što se dobije predviđeni šum uspoređuje se sa stvarnim šumom i podešava neuronska mreža. Proces se može vidjeti na slici 3.2.



Slika 3.2. Proces treniranja difuzijskog modela [10]

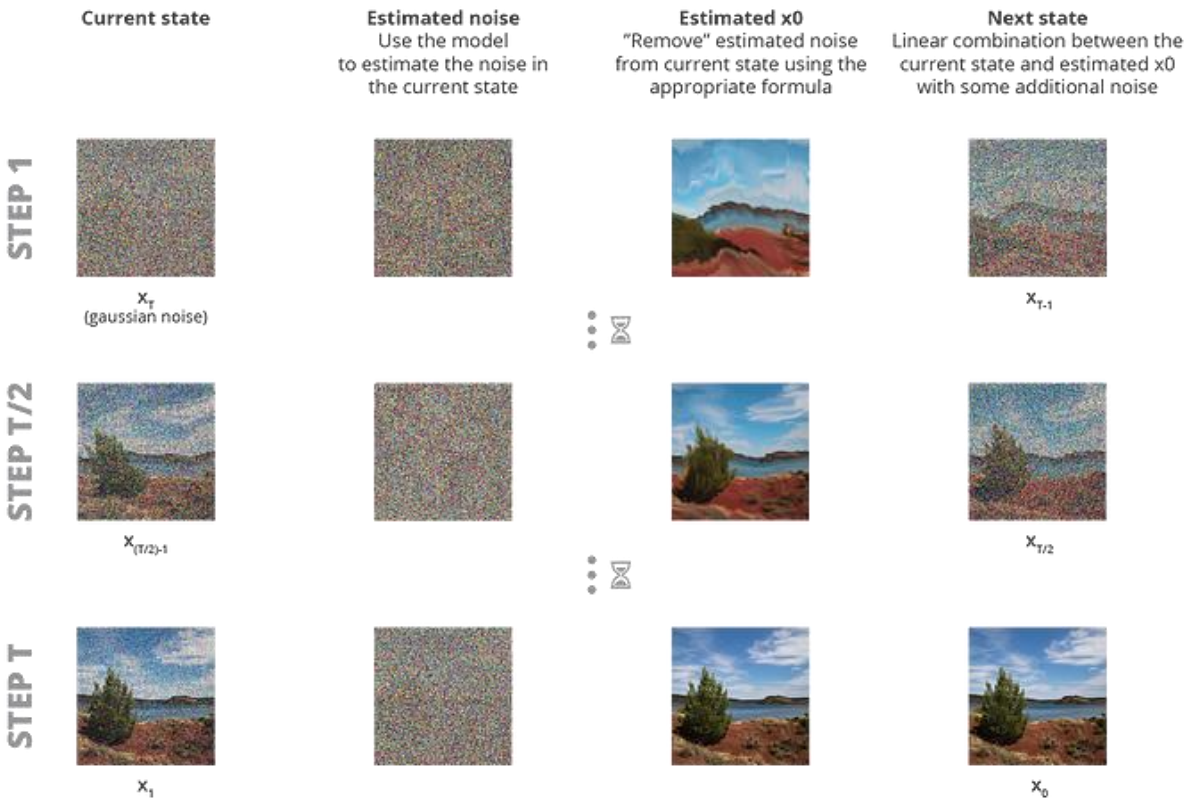
3.2. Generiranje slike pomoću difuzijskog modela

Nakon što je difuzijski model treniran spreman je za generiranje slika koristeći proces sličan procesu obrnute difuzije. Razlika je u tome što model u svakome koraku pokušava predvidjeti sav šum na slici, a ne samo između koraka *T* i *T-1*.

Koraci za generiranje slike:

1. Model predviđa sav šum na slici za korak *T*
2. Predviđeni šum oduzima se sa slike
3. Novoj slici kojoj je oduzet šum ponovno se dodaje šum za korak *T-1*
4. Ponavljaju se koraci od 1 do 3 sve dok *T* nije jednak 0

Za ovaj proces potreban je početan šum, broj koraka i uzorkivač Proces je prikazan na slici 3.3.



Slika 3.3. Proces generiranja slike [10]

3.3. Generiranje slike sa Stable Diffusionom

Postoji nekoliko različitih pristupa za generiranje slike s difuzijskim modelima. U ovome projektu bit će prikazano kako *Stable Diffusion* koristi difuzijske modele, ali postoje i drugi pristupi. Glavni konkurenti su *DALL-E 2* [11] i *Midjourney* [12].

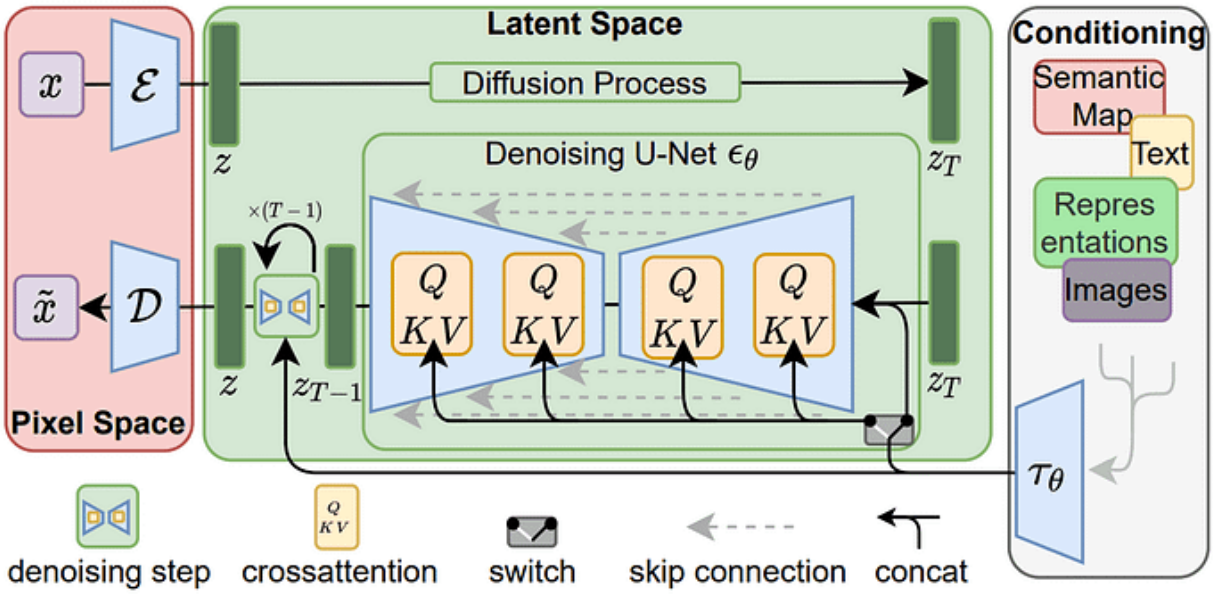
Stable Diffusion [13] spaja difuzijski model, varijacijski *autoenkoder* i transformator kako bi se generirala slika na temelju tekstualnih smjernica.

Za svaku sliku potrebno je odabrati trenirani difuzijski model, uzorkivač, broj koraka, sjeme, dimenzije slike i skalu usmjerivanja. Utjecaj različitih postavki će biti testiran i prikazan u poglavlju testiranje i rezultati.

Kako bi računalo razumjelo tekstualne smjernice potrebno je pretvoriti tekstualne smjernice u tokene. Za to se koristi transformator koji koristi *CLIP* model [14]. Taj model je neuronska mreža koja je trenirana na parovima slika i teksta.

Transformator traži riječi iz tekstualnih smjernica koje prepoznaje te ih pretvara u vektore koje će difuzijska mreža koristiti za usmjeravanje. Riječi sa istim značenjem imati će isti token.

Osim pozitivnih smjernica kojima korisnik definira što želi vidjeti na slici *Stable Diffusion* omogućuje upisivanje negativnih smjernica [15]. Nakon toga, generira se šum na temelju sjemena (engl. *seed*) ili početne vrijednosti, koji se zatim varijacijskim *autoenkoderom* smanji u latentni prostor. U latentnom prostoru slika ima četiri puta manju dimenziju. Nakon što je šum u latentnom prostoru i smjernice pretvorene u tokene difuzijski model započinje svoj proces generiranja slike. Generiranu sliku zatim varijacijski *autoenkoder* dekodira. Proces generiranja slike se može vidjeti na slici 3.4.



Slika 3.4. Proces generiranja slike sa *Stable Diffusionom* [16]

Kako bi se moglo upravljati koliko će difuzijski model pratiti smjernice uvodi se ljestvica navođenja (engl. *guidance scale*). Generiranjem bez smjernica generirati će se nasumična slika, ali ako se samo prate smjernice generirati će se slika s artefaktima. Da bi se izbjegli artefakti i dobili bolji rezultati *Stable Diffusion* generira jednu sliku bez smjernica i jednu sa smjernicama. Ovisno o postavljenoj vrijednosti ljestvice navođenja uvećavaju se razlike između te dvije slike.

4. IZRADA APLIKACIJE

U ovome poglavlju navedeni su potrebni alati i biblioteke za izradu aplikacije. Nakon toga opisan je dizajn i implementacija aplikacije.

4.1. Korišteni alati i programska okruženja

Za pisanje koda koristit će se *Visual Studio Code* tekstualni editor. Aplikacija je pisana u *Python* programskom jeziku.

Za vizualne elemente desktop aplikacije koristit će se biblioteka *CustomTkinter*. U njoj su definirani elementi poput dugmeta, polja za unos teksta, tekstualnih polja i drugih. Jedan od najvažnijih elemenata je okvir (engl. *Frame*) pomoću kojeg će se elementi grupirati.

Glavni i najbitniji dio aplikacije je dio za učitavanje modela u memoriju i generiranje slike pomoću modela. Za njega će se koristiti *Stable Diffusion* biblioteka [17] pisana u *Python* programskom jeziku.

Kako bi se mogle generirati slike potreban je i trenirani difuzijski model kojeg je moguće preuzeti na web stranici *Hugging Face* [18]. Na njoj je moguće besplatno preuzeti mnoštvo treniranih strojnih modela. Neki difuzijski modeli su trenirani da budu dobri u svemu, a neki su usmjereni na određenu temu ili grafički stil. Neka od usmjerenja su:

- Fotorealistične slike
- Portreti osoba
- Animirani stil
- Slike zgrada i gradova

U projektu će se koristiti dva *Stable Diffusion* modela kako bi se vidjele razlike u slikama koje generira opći model i model koji je treniran s ciljem da generira portrete osoba.

Opći model je naziva *stable-diffusion-2-1*, a model za portrete je *portraitplus*. Oba modela je moguće pronaći i preuzeti na web stranici *Hugging Face*. U ovome projektu koristit će se skripta iz *Stable Diffusion* biblioteke za preuzimanje modela koja pri preuzimanju pretvara *Stable Diffusion* model u *ONNX* model. Ovaj postupak je nužan zato što će se program raditi na sustavu koji koristi grafičku karticu od proizvođača *AMD* koja ne podržava tehnologiju *CUDA*. *CUDA* je softverski sloj koji daje direktan pristup instrukcijama grafičke kartice, te omogućava optimizaciju i ubrzanje pomoću paralelnog programiranja.

ONNX (engl. *Open Neural Network Exchange*) je okvir za strojno učenje otvorenog koda koji služi za pretvaranje i prijenos različitih modela kako bi se mogle iskoristiti najbolje značajke različitih alata i okvira za strojno učenje. Neki okviri za strojno učenje su bolji za treniranje dok su drugi bolji za korištenje istoga modela. ONNX također ima i svoj format modela kao i svoj izvršni okvir (engl. *runtime framework*) koji će se koristiti u ovome projektu.

4.2. Dizajn i arhitektura

Desktop aplikacija je podijeljena na tri glavna dijela.

Prvi dio je klasa *App* koja sadržava sve potrebne instance i svojstva za rad aplikacije.

Drugi dio je namijenjen za grafički prikaz i sastoji se od klasa za grafičke komponente aplikacije. To su komponente poput gumba s kojim se započinje generiranje slike ili polja za unos tekstualnih smjernica. Instance navedenih komponenata se slažu unutar *App* klase unutar koje se definira raspored komponenata.

Treći dio je klasa *ImageGenerator* koja koristi metode iz *Stable Diffusion* biblioteke kako bi se preuzeti model mogao koristiti. App klasa povezuje grafičko sučelje i *ImageGenerator* klasu, odnosno prosljeđuje postavke koje je korisnik odabrao na grafičkom sučelju metodama *ImageGeneratora*.

Kada korisnik odabere model i algoritam za uzorke može učitati model u radnu memoriju. Nakon što je model učitan može se pozvati metoda za generiranje slike. Kada generiranje slike završi spremit će se u mapu *images* s nazivom koji je korisnik definirao, te prikazati s desne strane grafičkog sučelja.

4.3. Implementacija grafičkog sučelja

Kako bi se prikazalo grafičko sučelje potrebno je postaviti dimenzije grafičkog sučelja i inicijalizirati početna svojstva (Kod 4.1.). Veličina prozora aplikacije se računa na temelju dimenzija korisničkog monitora. Koristi se responzivni dizajn što znači da će sve komponente prilagoditi svoje dimenzije u odnosu na veličinu prozora aplikacije.

```
        self.title("Stable diffusion app")
        self.overrideredirect(False)
        self.geometry("{0}x{1}-8-2"
        .format(self.winfo_screenwidth(), self.winfo_screenheight()))
        self.variable_init()
```

Kod 4.1. Postavljanje veličine grafičkog sučelja i početnih svojstava

Na vrhu grafičkog sučelja nalaze se polja za unos tekstualnih smjernica za generiranje slike (Slika 4.1.). U kodu 4.2. može se vidjeti klasa u kojoj se definira komponenta za unos tekstualnih smjernica. Navedena klasa nasljeđuje klasu okvir (engl. *frame*) iz *CustomTkinter* biblioteke kojom se grupiraju elementi.

Slika 4.1. Komponenta za unos tekstualnih smjernica

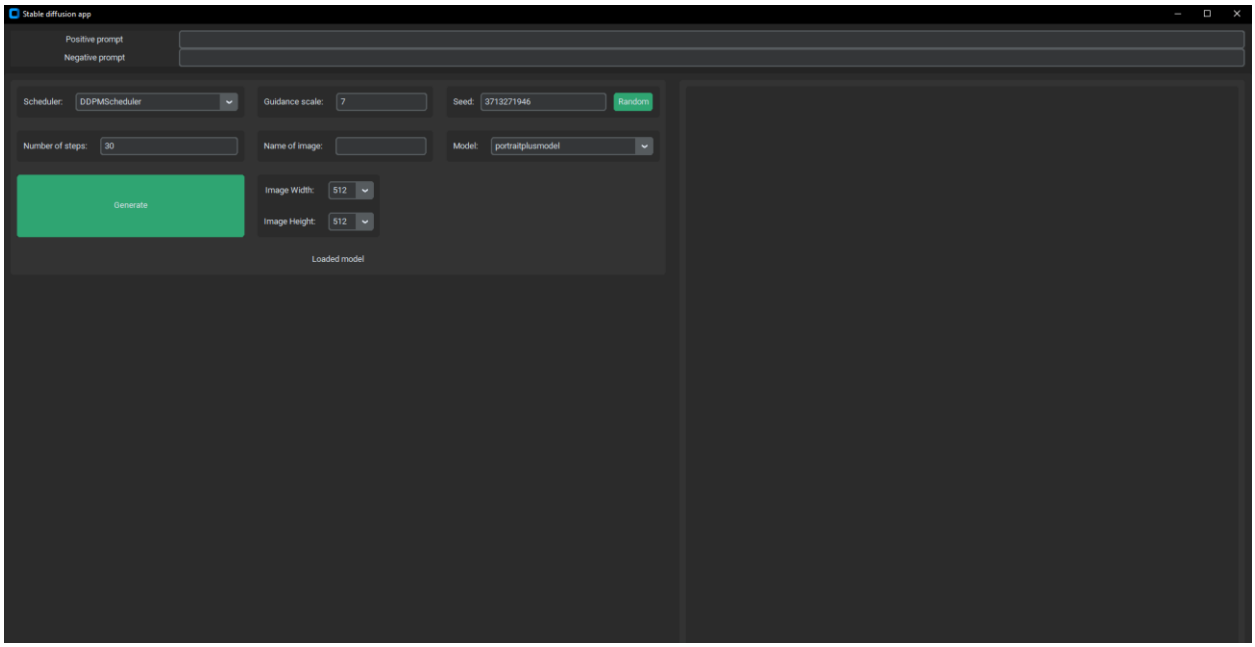
```
#Prompts
class Prompt(ctk.CTkFrame):
    def __init__(self, master, positive_promptVar : ctk.StringVar = None,
negative_promptVar : ctk.StringVar = None ):
        super().__init__(master)
        self.columnconfigure(0, weight = 1)
        self.columnconfigure(1, weight = 9)
        self.rowconfigure(0, weight = 1)
        self.rowconfigure(1, weight = 1)
        ctk.CTkLabel(self, text="Positive prompt")
        .grid(row = 0, column = 0, sticky = 'nsew')
        ctk.CTkEntry(self, textvariable = positive_promptVar)
        .grid(row = 0, column = 1, sticky = 'nsew')
        ctk.CTkLabel(self, text="Negative prompt")
        .grid(row = 1, column = 0, sticky = 'nsew')
        ctk.CTkEntry(self, textvariable = negative_promptVar)
        .grid(row = 1, column = 1, sticky = 'nsew')
```

Kod 4.2. Klasa za komponentu za unos tekstualnih smjernica

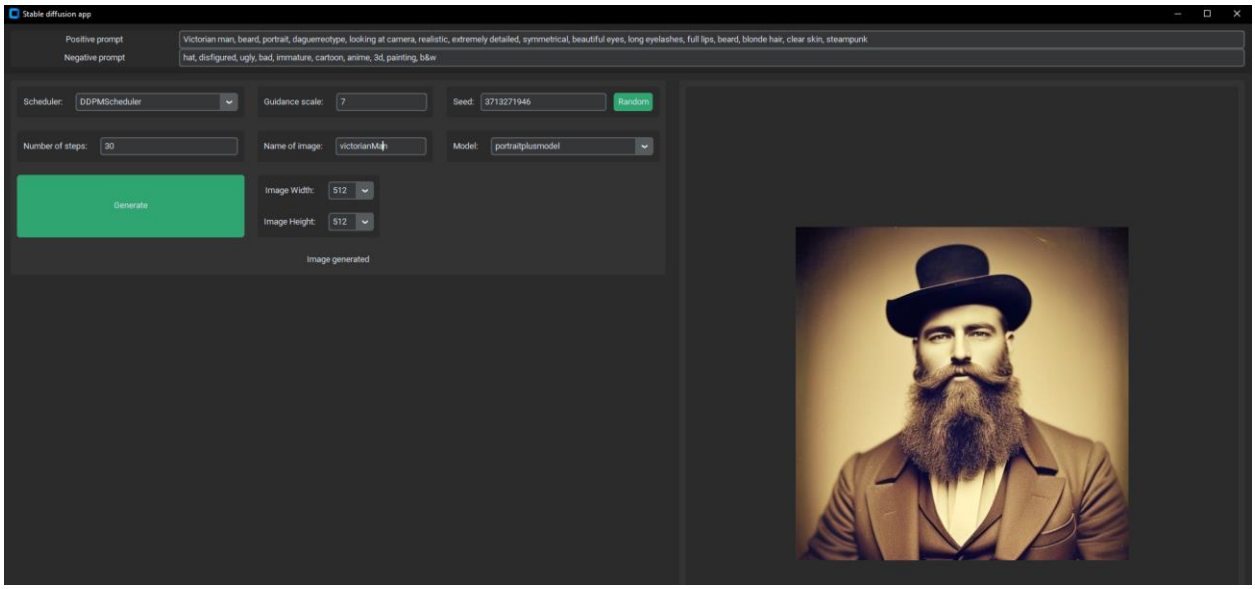
CustomTkinter podržava tri metode za raspored komponenata *grid*, *place* i *put*. Unutar svake komponente može se koristiti samo jedna vrsta metoda za raspored. U navedenoj komponenti za unos tekstualnih smjernica koristi se metoda *grid* koja radi na način da se postavi koordinatna mreža s željenim veličinama redaka i stupaca. Nakon definiranja svakog elementa potrebno je s metodom *grid* postaviti element u željeno polje.

Svakoj komponenti se prilikom inicijalizacije prosljeđuje okvir u kojem se nalazi i adrese svojstava iz *App* klase u koja se spremaju upisani podaci u toj komponenti.

Grafičko sučelje je podijeljeno na dva dijela, dio gdje korisnik unosi postavke za generiranje slike i dio za prikaz generirane slike (Slika 4.2.). U slučaju da korisnik još nije generirao sliku dio za prikaz slike bit će prazan. Sučelje nakon generiranja slike može se vidjeti na slici 4.3.

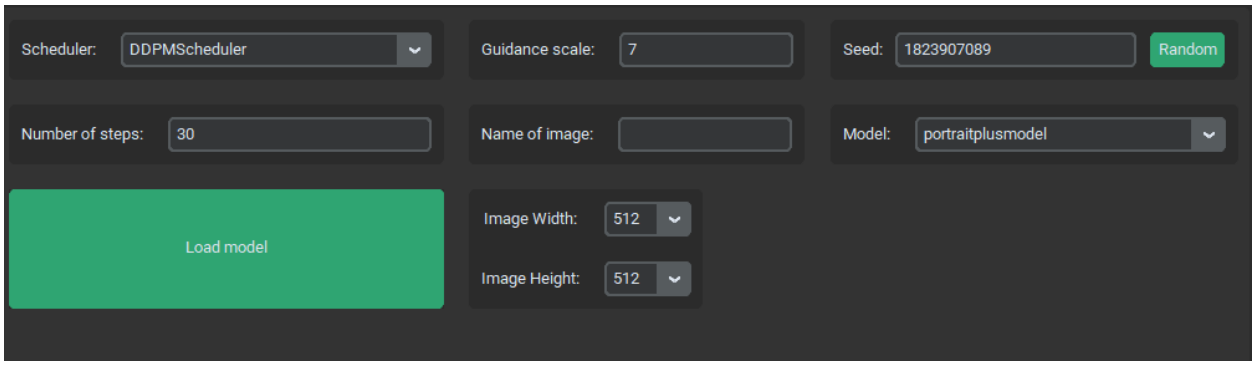


Slika 4.2. Grafičko sučelje aplikacije



Slika 4.3. Grafičko sučelje nakon generiranja slike

Uz tekstualne smjernice potrebne su i dodatne postavke koje korisnik mora definirati prije generiranja slike. One se postavljaju u odjeljku za postavke koji se sastoji od zasebnih komponenata (Slika 4.4.). Svaka komponenta sadrži logiku za određenu postavku poput ograničenja za unos brojeva ili dohvaćanje dostupnih modela koji su u mapi za modele (Kod 4.3.).



Slika 4.4. Odjeljak za postavke

```
#Select model
class ModelSelect(ctk.CTkFrame):
    def __init__ (self, master, model_name : ctk.StringVar, model_path :
ctk.StringVar):
        super().__init__(master)
        self.model_name = model_name
        self.model_path = model_path
        self.models = []
        self.get_models()
        ctk.CTkLabel(self, text = "Model:").pack(side = 'left',
        anchor = 'center', padx = 10, pady = 10)
        ctk.CTkComboBox(self, variable = model_name,
        values = [x[0] for x in self.models], width = 250)
        .pack(side = 'left', anchor = 'center', padx = 10, pady = 10)
        model_name.set(self.models[0][0])
        model_path.set(self.models[0][1])
        model_name.trace('w', self.update_model_path)

    def update_model_path(self, *args):
        for model in self.models:
            if model[0] == self.model_name.get():
                self.model_path.set(model[1])
                break

    def get_models(self):
        self.models = []
        for name in os.listdir(f"./model"):
            self.models.append([name,f"./model/{os.path.relpath(name)}"])
        if len(self.models) == 0:
            self.models = [["no model", ""]]
```

Kod 4.3. Komponenta za odabir dostupnih modela

4.4. Implementacija klase za generiranje slika

Klasa *ImageGenerator* sadrži metode koje koriste *Stable Diffusion* biblioteku za generiranje slika.

Kada korisnik odabere dostupan model i algoritam za uzorke, te pritisne dugme za učitavanje modela poziva se *load_pipe* metoda. Ova metoda učitava model u radnu memoriju nakon čega je aplikacija spremna generirati sliku. Za učitavanje modela koristi se *OnnxStableDiffusionPipeline* iz *Stable Diffuison* biblioteke (Kod 4.4.).

```
        try:
            self.pipe = OnnxStableDiffusionPipeline.from_pretrained(
                model,
                revision="onnx",
                provider=provider,
                safety_checker=None,
                scheduler=sched_model,
            )
        except Exception as e:
            return e
```

Kod 4.4. Metoda za učitavanje modela u radnu memoriju

Metodi za generiranje slike šalju se postavke koje je korisnik odabrao nakon čega ona prosljeđuje podatke učitanome modelu u memoriji i započinje generiranje slike (Kod 4.5.). Nakon generiranja slike i prije učitavanja novog modela potrebno je očistiti podatke iz memorije koji se ne koriste pozivanjem sakupljača smeća. To je jako važno zato što sustav koristi oko 20 gigabajta memorije, pa može vrlo brzo popuniti svu memoriju nakon čega može doći do prestanka rada aplikacije.

```
    def txt_to_img(self, prompt, negative_prompt, num_inference_steps,
guidance_scale, width, height, seed, image_name):
        rng = np.random.RandomState(int(seed))
        try:
            image = self.pipe(
                prompt,
                int(height),
                int(width),
                int(num_inference_steps),
                int(guidance_scale),
                negative_prompt,
                generator = rng,
                ).images[0]
            image.save("./images/" + image_name+ ".png")
        except Exception as e:
            return e
        del image
        del rng
        gc.collect()
        return "Image generated"
```

Kod 4.5. Metoda za generiranje i spremanje slike

5. TESTIRANJE I REZULTATI

U ovome radu testirat će se utjecaj odabranih modela i ostalih postavki na konačan rezultat.

Testovi će se provesti na način da unutar svakoga testa sve postavke za generiranje slike ostaju iste osim naznačene postavke koja će se mijenjati.

Tekstualne smjernice i sjeme (engl. *seed*) će biti iste pri testiranju iste postavke, ali će se razlikovati pri testiranju različitih postavki. Na primjer tekstualna smjernica će biti ista za svaku sliku koja testira utjecaj veličine slike na rezultat, ali različita od tekstualnih smjernica za testiranje utjecaja broja koraka.

Osnovne postavke za sva ostala testiranja:

- Uzorkivač = *Euler Discrete Scheduler*
- Broj koraka = 10
- Model = *portraitplus*
- Ljestvica navođenja = 7
- Širina = 512 piksela
- Visina = 512 piksela

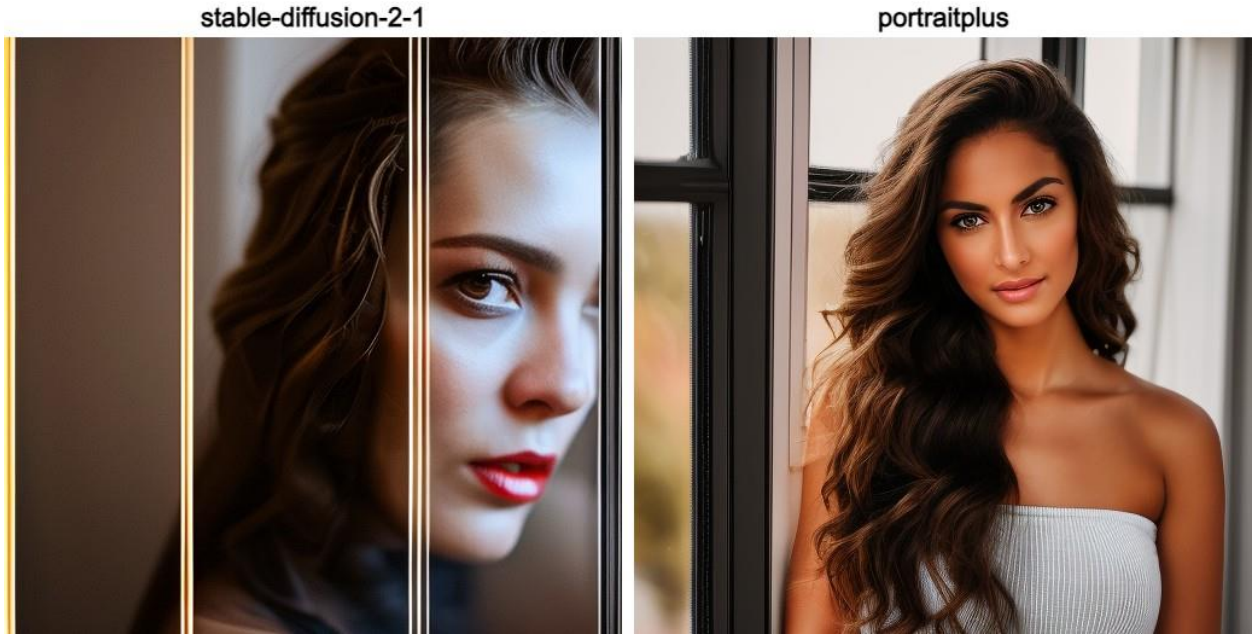
5.1. Različiti modeli

U ovome testu generirat će se dvije slike s potpuno identičnim postavkama, ali različitim modelima. Kao što je već prije navedeno neki modeli su trenirani da budu jako dobri u generiranju određenog stila ili određenih stvari.

Odabrana su dva modela *stable-diffusion-2-1* i *portraitplus* koji je treniran s ciljem da generira dobre portrete ljudi. Na slici 5.1. se može vidjeti da portret generiran s *portraitplus* modelom puno bolji od portreta generiranog s *stable-diffusion-2-1* modelom. Za generiranje slika korištene su tekstualne smjernice prikazane u tablici 5.1.

Tablica 5.1. Tekstualne smjernice za testiranje utjecaja različitih modela

Tekstualne smjernice	RAW photo, (detailed face)+, portrait of a beautiful woman posing for a picture in front of a window, 80mm, (soft golden fill light), f22, trending on ArtStation Pixiv, high detail, sharp focus, aesthetic, 8k uhd, DSLR, intricate details, soft lighting, high quality
Negativne tekstualne smjernice	(deformed iris, deformed pupils, semi-realistic, cgi, 3d, render, sketch, cartoon, drawing, anime)++, text, close up, cropped, out of frame, worst quality, low quality, jpeg artifacts, ugly, duplicate, morbid, mutilated, extra fingers, mutated hands, poorly drawn hands, poorly drawn face, mutation, deformed, blurry, dehydrated, bad anatomy, bad



Slika 5.1 Rezultati slika s različitim modelom

5.2. Različiti uzorkivači

U ovome testu primijećene su dvije grupe uzorkivača. Jedna grupa uzorkivača prikazana na slici 5.2. generira različitu sliku unatoč tome što je isto početno sjeme, dok druga grupa prikazana na slici 5.3. generira istu sliku. Za generiranje slike korištene su smjernice prikazane u tablici 5.2.

Na slici 5.3. može se primijetiti da svaki uzorkivač generira različitu sliku, ali su rezultati vrlo slični. Također može se primijetiti da nekim uzorkovačima je potrebno više koraka od drugih da bi se generirala slika bez šuma i artefakata.

Tablica 5.2. Tekstualne smjernice za testiranje utjecaja uzorkivača

Tekstualne smjernice	Close-up portrait of mix of European lady, floral printed clothing, extremely high-quality photograph, outdoor background, intricate, exquisite details and textures, warm lighting, sharp focus, high resolution, detailed eyes, 8k UHD, Nikon D850, high quality, film grain, hyper-realistic skin (detailed skin:1. 3)
Negativne tekstualne smjernice	CyberRealistic_Negative-neg, bw, worst quality, low quality, (deformed, distorted, disfigured:1.3), poorly drawn, bad anatomy, wrong anatomy, extra limb, missing limb, floating limbs, (mutated hands and fingers:1.4), disconnected limbs, mutation, mutated, ugly, disgusting, blurry, amputation, cloned, bad nail, bad eyes

Slika 5.2 Uzorkivači koji generiraju različitu sliku s istim početnim sjemenom

Slika 5.3. Uzorkivači koji generiraju istu sliku s istim početnim sjemenom

5.3. Utjecaj veličine početne slike na generiranu sliku

U ovome testu mijenjana je početna veličina slike, odnosno početnog šuma. Kao što se može primijetiti na slici 5.4. izmjenom veličine promijenio se i početni šum koji ima jako veliki utjecaj na krajnji rezultat.

Izmjenom veličine početne slike povećavaju se zahtjevi na sklopovlje. U ovome testu nije bilo moguće testirati veličinu 640x640 zbog nedovoljno memorije na grafičkoj kartici. Tekstualne smjernice korištene za generiranje slika za ovaj test prikazane su u tablici 5.3.

Tablica 5.3. Tekstualne smjernice za testiranje utjecaja početne veličine slike

Tekstualne smjernice	realistic portrait of a viking warrior, trending on artstation, sharp focus, sharp focus, studio shot, intricate detail, highly detailed
Negativne tekstualne smjernice	(deformed iris, deformed pupils, semi-realistic, cgi, 3d, render, sketch, cartoon, drawing, anime)++, text, close up, cropped, out of frame, worst quality, low quality, jpeg artifacts, ugly, duplicate, morbid, mutilated, extra fingers, mutated hands, poorly drawn hands, poorly drawn face, mutation, deformed, blurry, dehydrated, bad anatomy, bad proportions, extra limbs, cloned face, disfigured, gross proportions, malformed limbs, missing arms, missing legs, extra arms, extra legs, fused fingers, too many fingers, long neck

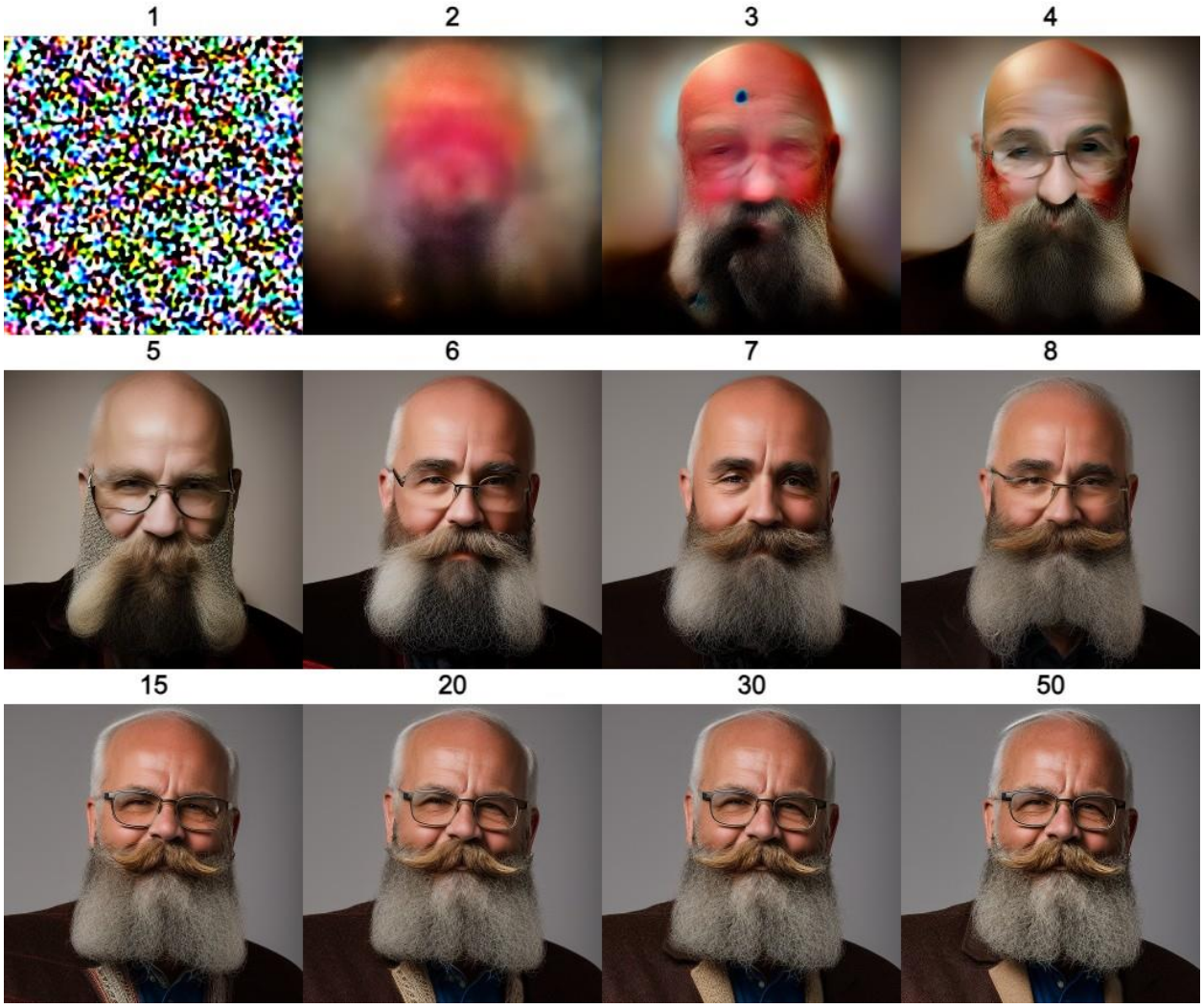
Slika 5.4 Utjecaj dimenzija početnog šuma na generiranu sliku

5.4. Razvoj slike po koracima

Ovim testom na slici 5.5. prikazuje se generiranje slike korak po korak. Može se primijetiti da se najviše šuma uklanja na početku. Razlike između slike generirane s 30 koraka i 50 koraka su vrlo male. Može se zaključiti da je najbolji pristup generirati slike s desetak koraka, a nakon što je rezultat zadovoljavajući ponovno generirati sliku s više koraka. Tekstualne smjernice korištene u ovome testu su prikazane u tablici 5.4.

Tablica 5.4. Tekstualne smjernice za usporedbu razvoja slike po koracima

Tekstualne smjernice	photo of a old man with beard, 80mm, f22, trending on ArtStation Pixiv, high detail, sharp focus, aesthetic, 8k uhd, DSLR, intricate details, soft lighting, high quality
Negativne tekstualne smjernice	(deformed iris, deformed pupils, semi-realistic, cgi, 3d, render, sketch, cartoon, drawing, anime)++, text, close up, cropped, out of frame, worst quality, low quality, jpeg artifacts, ugly, duplicate, morbid, mutilated, extra fingers, mutated hands, poorly drawn hands, poorly drawn face, mutation, deformed, blurry, dehydrated, bad anatomy, bad proportions, extra limbs, cloned face, disfigured, gross proportions, malformed limbs, missing arms, missing legs, extra arms, extra legs, fused fingers, too many fingers, long neck



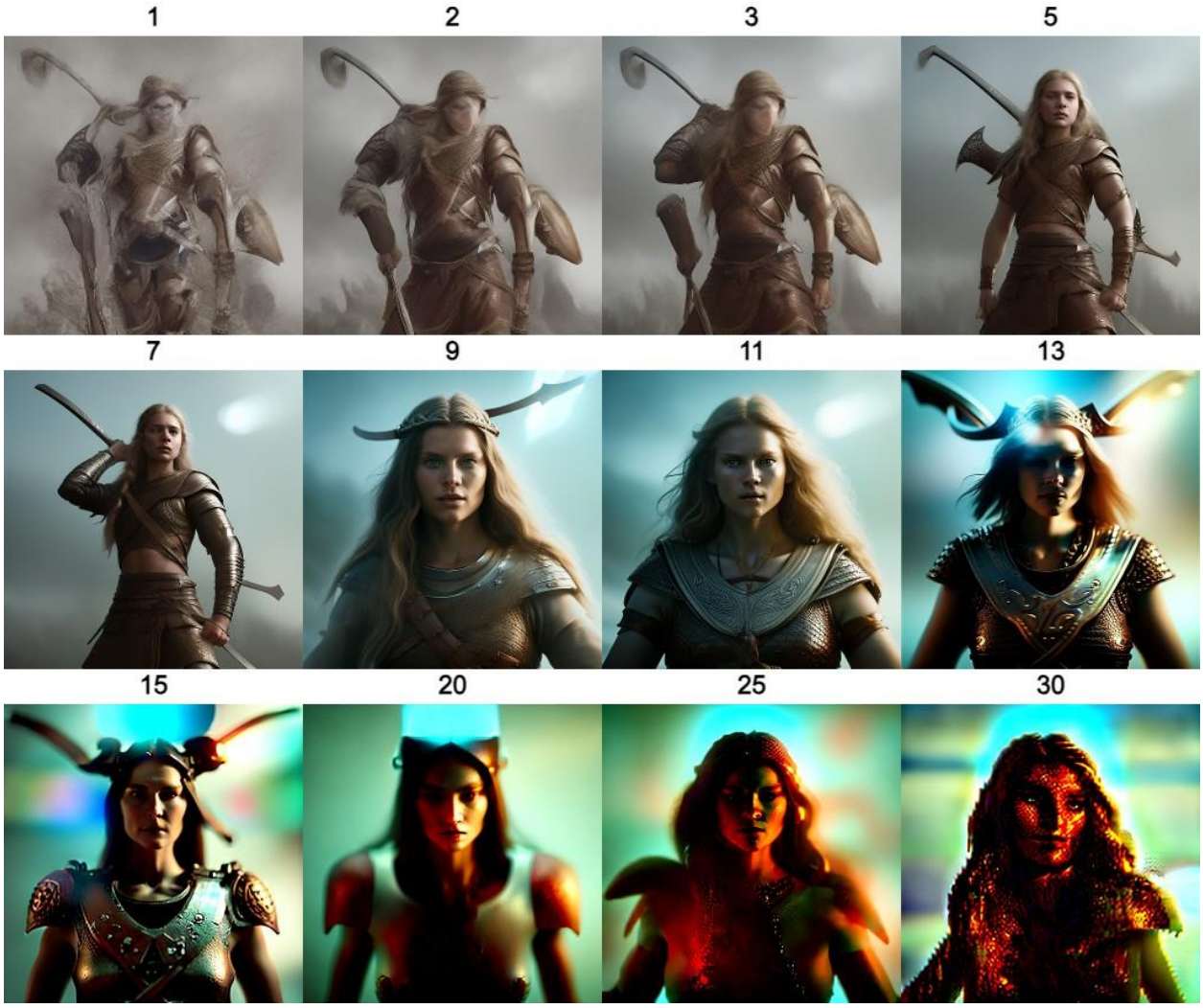
Slika 5.5 Utjecaj broja koraka na generiranu sliku

5.5. Utjecaj ljestvice navođenja

Ovim testom prikazan je utjecaj ljestvice navođenja na slici 5.6. Može se primijetiti da su generirane slike pri vrlo maloj ljestvici navođenja poput dima i imaju mali kontrast. Slike s velikom ljestvicom navođenja imaju veliki kontrast i počinju dobivati artefakte. Preporučena ljestvica navođenja je od 7 do 11 za kreativne slike, dok vrijednosti od 12 do 20 za oštrije slike koje dobro prate smjernice. Tekstualne smjernice korištene u ovome testu su prikazane u tablici 5.5.

Tablica 5.5. Tekstualne smjernice za testiranje utjecaja ljestvice navođenja

Tekstualne smjernice	A finely detailed depiction of a stunning Viking warrior woman, illuminated with cinematic lighting, reflections, and ray tracing effects. Created using Octane Render, the artwork is in sharp focus and boasts a seamless 8K resolution. Crafted as a digital painting, it's gaining popularity on ArtStation. Crafted collaboratively by Seb McKinnon and Greg Rutkowski.
Negativne tekstualne smjernice	(deformed iris, deformed pupils, semi-realistic, cgi, 3d, render, sketch, cartoon, drawing, anime)++, text, close up, cropped, out of frame, worst quality, low quality, jpeg artifacts, ugly, duplicate, morbid, mutilated, extra fingers, mutated hands, poorly drawn hands, poorly drawn face, mutation, deformed, blurry, dehydrated, bad anatomy, bad proportions, extra limbs, cloned face, disfigured, gross proportions, malformed limbs, missing arms, missing legs, extra arms, extra legs, fused fingers, too many fingers, long neck



Slika 5.6 Utjecaj ljestvice navođenja na generiranje slike

Slika 5.7. je generirana sa ljestvicom navođenja postavljenom na 20, ali je povećan broj koraka za generiranje s 10 na 30. Povećanjem broja koraka nestali su artefakti, te se može zaključiti da za veću ljestvicu navođenja je potreban veći broj koraka kako bi dobili zadovoljavajući rezultat.

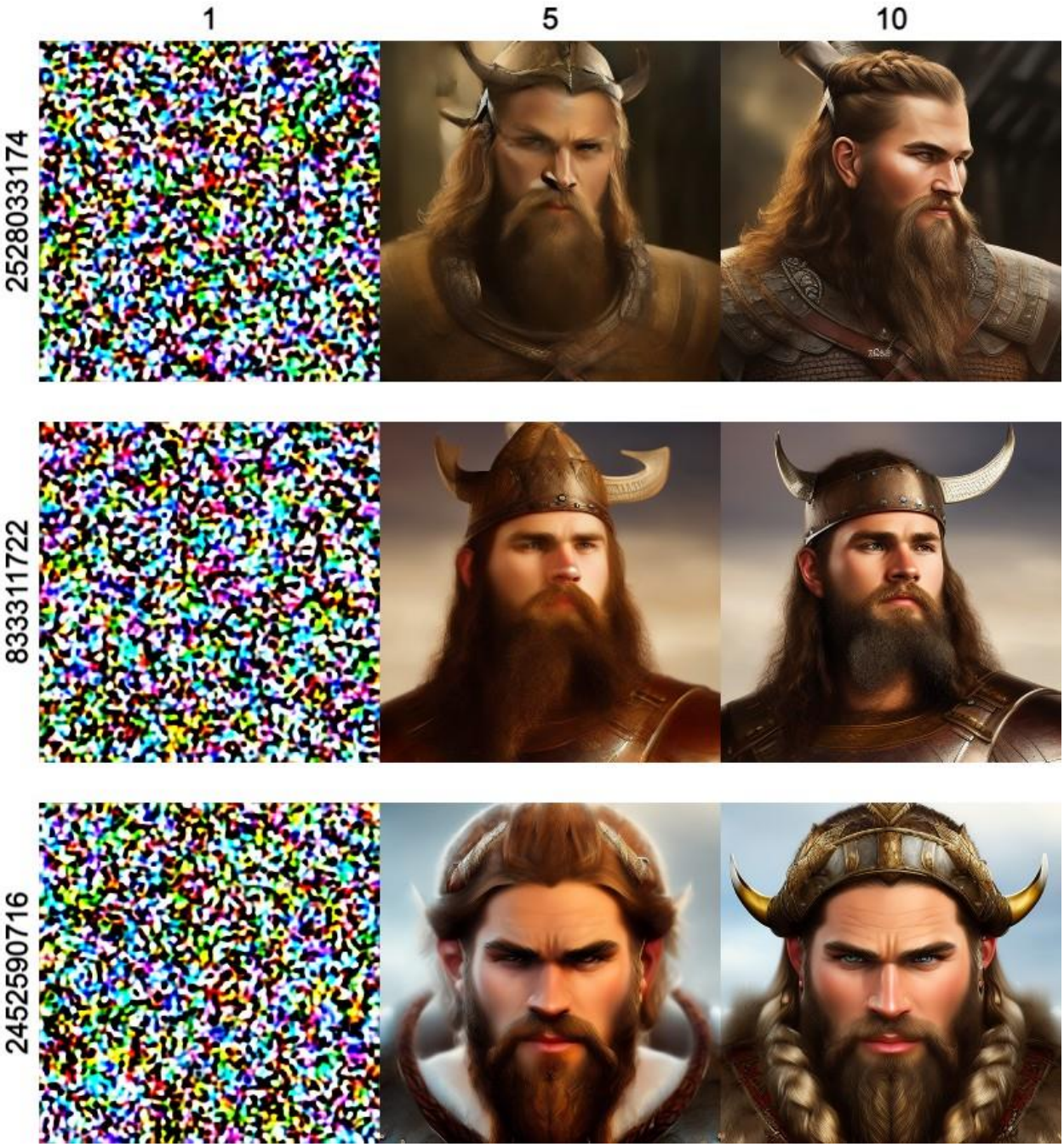
Slika 5.7 Slika generirana sa ljestvicom navođenja postavljenom na 20 i 30 koraka

5.6. Utjecaj sjemena

U ovome testu mijenjano je početno sjeme koje utječe na početni šum. Na slici 5.8. može se vidjeti utjecaj sjemena i broja koraka na generiranu sliku. Kao što se moglo vidjeti u testu gdje je mijenjana veličina slike izmjenom početnog šuma generira se potpuno drugačija slika. Tekstualne smjernice korištene u ovome testu su prikazane u tablici 5.6.

Tablica 5.6. Tekstualne smjernice za testiranje utjecaja početnog sjemena

Tekstualne smjernice	realistic portrait of a viking warrior, trending on artstation, sharp focus, sharp focus, studio shot, intricate detail, highly detailed
Negativne tekstualne smjernice	(deformed iris, deformed pupils, semi-realistic, cgi, 3d, render, sketch, cartoon, drawing, anime)++, text, close up, cropped, out of frame, worst quality, low quality, jpeg artifacts, ugly, duplicate, morbid, mutilated, extra fingers, mutated hands, poorly drawn hands, poorly drawn face, mutation, deformed, blurry, dehydrated, bad anatomy, bad proportions, extra limbs, cloned face, disfigured, gross proportions, malformed limbs, missing arms, missing legs, extra arms, extra legs, fused fingers, too many fingers, long neck



Slika 5.8 Utjecaj sjemena i broja koraka na generaciju slike

6. ZAKLJUČAK

U ovome radu istražene su vrste generativnih modela, kako se treniraju, te kako *Stable Diffusion* koristi razne neuronske mreže i generativne modele kako bi generirao slike. Izradom ovog projekta postignuta je funkcionalna desktop aplikacija za generiranje slika koja je poslužila za testiranje utjecaja pojedinih postavki na krajnji rezultat generirane slike. Testiranjem se utvrdilo da izmjenom bilo koje postavke možemo generirati potpuno drugačiju sliku.

Tijekom testiranja povećanja rezolucije slike došlo je do poteškoća zato što 8 gigabajta memorije na grafičkoj kartici nije bilo dovoljno za generiranje slike visine i širine 640 piksela. Može se zaključiti da je proces generiranja slike jako zahtjevan i potrebno je puno resursa. *Stable Diffusion* je otvorenog koda i razvojem novih boljih grafičkih kartica sve više ljudi ga počinje koristiti i unaprjeđivati.

LITERATURA

[1] "What is machine learning," IBM, [Online]. Available: https://www.ibm.com/topics/machine-learning. [Accessed 21 8 2023].

[2] A. Takyar, "AI USE CASES & APPLICATIONS ACROSS MAJOR INDUSTRIES," [Online]. Available: https://www.leewayhertz.com/ai-use-cases-and-applications/.

[3] J. Rocca, "Understanding Generative Adversarial Networks (GANs)," 7 1 2019. [Online]. Available: https://towardsdatascience.com/understanding-generative-adversarial-networks-gans-cd6e4651a29.

[4] R. O'Connor, "Introduction to Variational Autoencoders Using Keras," 3 1 2022. [Online]. Available: https://www.assemblyai.com/blog/introduction-to-variational-autoencoders-using-keras/.

[5] J. Rocca, "Understanding Variational Autoencoders (VAEs)," 24 9 2019. [Online]. Available: https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73.

[6] S. A. N. Karagiannakos, "Diffusion models: toward state-of-the-art image generation," 2022. [Online]. Available: https://theaisummer.com/diffusion-models/#:~:text=Stable%20diffusion%3A%20Latent%20diffusion%20models,more%20detail%2C%20Rombach%20et%20al. [Accessed 20 8 20223].

[7] P. Dhariwal and A. Nichol, "Diffusion Models Beat GANs on Image Synthesis," 1 6 2021. [Online]. Available: https://arxiv.org/pdf/2105.05233.pdf. [Accessed 20 8 2023].

[8] A. Gainetdinov, "Diffusion Models vs. GANs vs. VAEs: Comparison of Deep Generative Models," 12 5 2023. [Online]. Available: https://pub.towardsai.net/diffusion-models-vs-gans-vs-vaes-comparison-of-deep-generative-models-67ab93e0d9ae#:~:text=Unlike%20VAEs%20and%20GANs%2C%20which,modes%20of%20the%20training%20dataset..

[9] "Stable Diffusion Samplers: A Comprehensive Guide," 10 6 2023. [Online]. Available: https://stable-diffusion-art.com/samplers/.

[10] J. Rocca, "Understanding Diffusion Probabilistic Models," 5 12 2022. [Online]. Available: https://towardsdatascience.com/understanding-diffusion-probabilistic-models-dpms-1940329d6048.

[11] "DALL-E 2," [Online]. Available: https://openai.com/dall-e-2.

[12] "Midjourney," [Online]. Available: https://docs.midjourney.com/.

[13] "How does Stable Diffusion work," 23 7 2023. [Online]. Available: https://stable-diffusion-art.com/how-stable-diffusion-work/.

[14] "Hugging Face CLIP," [Online]. Available: https://huggingface.co/docs/transformers/model_doc/clip.

[15] "How does negative prompt work?," 8 1 2023. [Online]. Available: https://stable-diffusion-art.com/how-negative-prompt-work/.

[16] A. Saxena, "Understanding Diffusion & Stable Diffusion in AI," 18 4 2023. [Online]. Available: https://medium.com/tech-blogs-by-nest-digital/understanding-diffusion-stable-diffusion-in-ai-113f32c977f3.

[17] "Stable Diffusion Github," [Online]. Available: https://github.com/CompVis/stable-diffusion. [Accessed 5 5 2023].

[18] "Hugging Face," [Online]. Available: https://huggingface.co/.

SAŽETAK

Cilj rada je bio napraviti program za generiranje portreta koristeći *Stable Diffuison* biblioteku. U radu su istražene vrste generativnih modela, kako se treniraju generativni modeli, te njihove prednosti i mane. U radu također prikazan način na koji *Stable Diffusion* koristi difuzijske modele za generiranje slika. Na osnovu stečenih znanja napravljena je desktop aplikacija za generiranje portreta pomoću koje je testirano kako različiti parametri utječu na generiranje slike.

Ključne riječi: generativni modeli, generiranje slike, Stable Diffusion

ABSTRACT

Automated generation of portraits using Stable Diffusion library

The aim of this project is to create a program for generating portraits using Stable Diffusion library. This paper explores the types of generative models, how generative models are trained and their advantages and disadvantages. The paper also shows the way in which Stable Diffusion uses diffusion models to generate images. Based on the acquired knowledge, a desktop application for generating portraits was created, which was used to test how different parameters affect image generation.

Keywords: generative models, generating images, Stable Diffusion