**SVEUČILIŠTE U SPLITU**

**PRIRODOSLOVNO-MATEMATIČKI FAKULTET**

**UVOD U UMJETNU INTELIGENCIJU**

**SEMINARSKI RAD**

PROBLEMI PRETRAGE U UMJETNIČKOJ KREACIJI SLIKA

**PROFESOR:** Saša Mladenović **STUDENT:** Rei Krstić

## UVOD

U posljednje vrijeme se dosta susrećemo s umjetnom inteligencijom (eng. Artificial Intelligence, ili AI) u svakodnevnom životu. Specifičnije, susrećemo se s generativnom umjetnom inteligencijom. Ako ne u slobodno vrijeme, onda je barem susrećemo u svrhe učenja, te općenito u obrazovanja. U takvim slučajevima najčešće koristimo nekakav chatbot (kao što su, na primjer, ChatGPT, Gemini i Copilot). Međutim, što ako ne želimo generativnu umjetnu inteligenciju koristiti u ove „dosadnije“ svrhe (kako bi nam AI mogao reći činjenice), već za nešto interesantnije, zabavnije i kreativnije, kao što je generiranja umjetničkih djela poput glazbe, poezije, i onog „najzanimljivijeg“, slika. Zadnjih nekoliko godina se jasno mogao vidjeti drastični razvoj tehnologija potrebnih za generiranje slika umjetnom inteligencijom. Upisivanjem prompta se u roku od nekoliko sekundi sad može „stvoriti“ novo umjetničko djelo, no postavlja se pitanje na koji način AI generatori umjetnosti (poput DALL•E-a, Midjourneyja i Stable Diffusiona) uopće rade.

U ovom seminaru ćemo upravo to pobliže i pogledati. Odrediti i definirati probleme, odnosno metode, pretrage pri generiranju umjetničkih djela, ili specifičnije, slika. Uvodi se pitanje: „Kakve veze pretraga ima sa umjetničkom kreacijom slika?“ Kako bismo na ovo pitanje uopće mogli odgovoriti, prvo trebamo postaviti još jedno važno pitanje: „Što je to metoda/problem pretrage u ovom kontekstu?“ Ovom metodom mi tražimo najbolje moguće rješenje iz skupa mogućih rješenja (koji su u ovom slučaju slike).

Prvo se mora razumjeti da je način na koji ljudi slikaju, te način na koji umjetna inteligencija slika fundamentalno različit. Primjerice, želimo li nacrtati astronauta koji jaše konja na Marsu, normalan čovjek bi najvjerojatnije prvo nacrtao konja, pa čovjeka, pa pozadinu (Mars). Generativna umjetna inteligencija ne „razmišlja“ kao mi. Ona čak ne bi ni crtala, već bi pretraživala i uspoređivala matematički prostor slika (npr. latentni prostor ili prostornu distribuciju piksela). Drugim riječima, sve bi se svodilo na, laički rečeno, uspoređivanje brojeva na način da finalni rezultat (završna slika) izgleda kao ono što smo od AI-a zatražili da nam nacrta (npr. preko prompta). Više o ovome kasnije.

Nadalje, u ovome radu ćemo dodatno analizirati upravo ove probleme s kojima se susreće generativna umjetna inteligencija, te pobliže pojasniti neke ključne pojmove i ideje kroz iduća poglavlja.

## PRETRAGA U GENERATIVNOJ UMJETNIČKOJ KREACIJI SLIKA

Kao što je prethodno rečeno, načini slikanja kod čovjeka i kod umjetne inteligencije su značajno drugačiji. Dok mi, kao ljudi, analiziramo neku sliku vizualno, AI model to čini čisto „računski“. AI koristi „razne“ algoritme za navigaciju kroz matematički prostor slika koji sadrže ogroman broj mogućih kombinacija parametara (odnosno vektora) slike (primjerice, boja kose, veličina nosa, duljina šape itd.). Na kraju se sve svodi na „igru uspoređivanja“ onoga što je zadano s onim što AI trenutačno kreira.

### Prostorna distribucija piksela:

Kad se priča o prethodno spomenutom matematičkom prostoru slika, najčešće govorimo, konkretno, o latentnom prostoru. Latentni prostor ne treba miješati sa prostornom distribucijom piksela, koji je također „vrsta“ matematičkog prostora. Pri generiranju slika umjetnom inteligencijom prostorna distribucija piksela nije najefikasniji način „mapiranja obilježja“ (u odnosu na, konkretno, latentni prostor) nekih slika s obzirom na to da se prostornom distribucijom piksela slike, kao što i sam naziv kaže, pikseli slike samo raspoređuju u prostoru (uz moguće neke dodatne značajke kao npr. RGB/dubina svakog piksela). Dok je ovaj način apsolutno potreban za „normalne“ računalom generirane slike (eng. computer generated images, ili CGI) zbog toga što taj CGI nije kreirao AI (odnosno model umjetne inteligencije) na „kreativan način“ i samim time nema potrebe za „poznavanjem“ ičeg osim rasporeda piksela na slici. Zato se ova vrsta matematičkog prostora slika najčešće koristi za, primjerice, specijalne efekte u filmovima i video igrama, te proceduralno generiranje u video igrama. Točnije, ova metoda „prostornog raspoređivanja“ se većinski koristi za generiranje slika koje se sastoje od matematičkih unaprijed definiranih objekata. Međutim, s obzirom na činjenicu da (moderni) AI „pravi“ slike na način da je potrebno „shvaćanje“ slike i njenih pojedinih obilježja u kontekstu na cjelinu umjetničke kreacije, onda prostorna distribucija nije uvijek dovoljna, odnosno potrebno nam je detaljnije poznavanje semantičkog značenja pojedinih objekata u slici.

### Latentni prostor:

Latentni prostor se vodi kao rješenje za problem koji je predstavljala prostorna distribucija piksela. To je višedimenzionalni matematički prostor, kao i prostorna distribucija piksela slika, no razlika među njima je ta da se u latentnom prostoru ne gledaju (odnosno ne „bilježe“) informacije o pojedinim pikselima na slici koju model pokušava kreirati, već se gledaju semantička obilježja „objekata“ ili karakteristika slike. Tako se mogu u cjelini, te u odnosu na cijelo djelo, gledati i „uspoređivati“ točnosti nekih obilježja sa slike (npr. duljina njuške, dužina ruku, boja kose i očiju…). Bitno je uzet u obzir i broj dimenzija prostora. Obično ima znatno manje dimenzija u ovom prostoru kombinacija brojeva koji označavaju semantička obilježja i svojstva, nego što ima u prostoru u kojem piksel označava interpolaciju. Ovo obično osigurava i bolju interpolaciju, odnosno, s obzirom da su slike komprimirane u njihova značenja, model može „lagano“ napraviti „prijelaze“ između njih.

Onda se koristi, skoro pa ista formula za udaljenost u latentnom prostoru, kao i u „pikselnom“ prostoru. Radi se o računanju Euklidske udaljenosti (samo s manjim razlikama između dvije formule):

Razlika je samo u tome što se kod prve formule (prostorna distribucija piksela) gledaju razlike piksela (,) slika (), dok se u drugoj formuli (latentni prostori) računaju konkretne razlike u obilježjima kod vektora značajki, odnosno kod latentnih vektora (), dvaju slika.

### Kratko o optimizaciji:

Naravno, kao što smo vidjeli u radu, kad se generira umjetnička kreacija AI-om, to se nikako ne radi slučajno, već se radi o uspoređivanju rezultata s onim što želimo dobiti(bez obzira na to koji model koristimo – Diffusion, GAN itd.). No ovo je dosta pojednostavljena definicija. Naime, točnije bi bilo reći da se ovdje radi optimizaciji, to jest, najboljom mogućom minimizaciji gubitka pri generiranju tražene slike (ovaj se proces ne odvija samo pri testiranju modela ili generiranju novih slika, već i pri samom treniranju modela). Za ovo možemo koristiti gradijentni spust i razne algoritme za optimizaciju (primjerice Adam), ovisno o potrebi i primjeni, te efikasno koristiti funkciju gubitka (eng. loss function, npr. MSD eng. Mean Squared Error), za, kao što i sam naziv funkcije govori, računanje „gubitak“. Odnosno, funkcija gubitka u ovom slučaju vraća koliko su AI generirane slike različite od pravih. Primjerice, tražimo li od modela da nam generira sliku mačke, onda će se prvo ta slika više puta uspoređivati s ostalim stvarnim slikama mačaka, odnosno njihovim latentnim prostorima iz dostupnog skupa podataka. Detaljnije o modelima ćemo u nastavku.

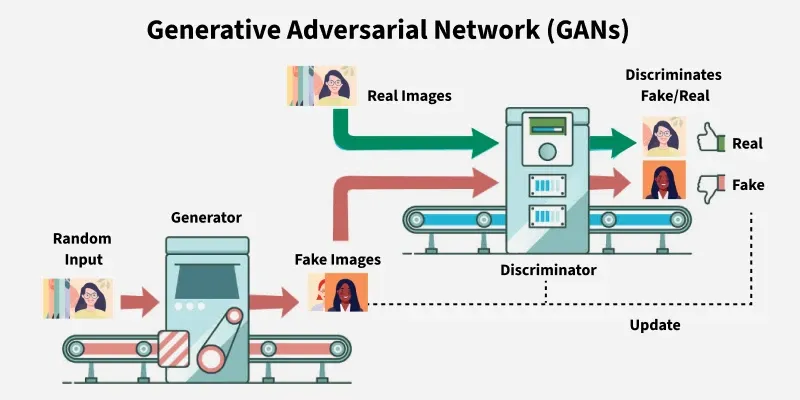
## PROBLEM PRETRAGE U GENERIRANJU SLIKA

### Kratko o GAN-ovima:

Prije pričanja o modernijim difuzijskim modelima koji se danas koriste, bilo bi dobro barem spomenuti generativne adversarijalne mreže (eng. Generative Adversarial Networks, ili kraće GAN-ove) koje se ne oslanjaju na pretragu. Unatoč tome što se danas ne koriste koliko i prije, one još uvijek imaju jako važne upotrebe i posebnosti koji ne može (trenutačno) u potpunosti nadmašiti niti difuzijski model (primjerice, kreiranje hiperrealističkih lica).

Kako uopće radi ovaj AI model i po čemu je poseban? Kao prvo, važno je napomenuti kako ovaj model nije zapravo samo jedan model, već se sastoji od dva modela. Kao što samo ime nalaže (generativne adversarijalne mreže), ovdje se radi o dvije neuronske mreže koje su adversarijalne, odnosno, „natjecateljski nastrojene“ (tijekom treniranja). Ove dvije mreže, ili modela, se sastoje od generatora, koji stvara „lažne“ slike koje pokušavaju oponašati „stvarne“ slike iz ponuđenog skupa podataka nad kojim je treniran. Također se sastoji i od diskriminatora. Diskriminator je zaslužan za „procjenu“ slika, točnije jesu li (u kontekstu kreacije slika) slike „stvarne“, odnosno iz skupa pravih slika, ili se radi o lažnim slikama, odnosno slikama koje je stvorio generator. Ukoliko diskriminator uspije točno klasificirati „lažnu“ generatorovu sliku, onda se „poboljšava“, odnosno optimizira generator. Ako se netočno klasificira slika generatora, tada se diskriminator „izgubio“ i mora se optimizirati tako da se slika bolje klasificira u idućoj iteraciji. Postupak se, laički rečeno, dalje svodi na natjecanje.

Ovo bi sve značilo da se GAN-ovi ne bi mogli jednostavno svrstati, ni u nadgledano, ni u nenadgledano učenje (slika dolje).



Nažalost, generativne adversarijalne mreže imaju svoje probleme. Primjerice, postoji mogućnost nestabilnog treninga gdje može doći do teške neravnoteže pri treniranju između generatora i diskriminatora, te mode collapsea koji sprječava „kreativnost“ tj. raznolikost pri generiranju slika. Probleme također predstavlja smanjena fleksibilnost vezana uz mijenjanje parametara „u hodu“. Točnije, želimo li generirati neku sliku s drugim parametrima, trebali bismo ponovno trenirati GAN. Međutim, za ovaj problem postoji i lakše rješenje. Tu nastupa difuzijski model.

### Difuzijski modeli:

Difuzijski modeli predstavljaju jedan od najsuvremenijih načina generiranja umjetnina umjetnom inteligencijom (DALL•E, Midjourney itd.). Oni počinju generiranje slika sa šumom (eng. noise). Pri procesu treniranja modela se slike iz danog skupa podataka prvo pretvore u šum (iz dane slike), te nazad u sliku kroz niz iteracija koristeći U-Net (o njemu više poslije). Recimo više o samom proces generiranja slika poslije treniranja. Dakle, cilj je iz šuma (ili noisea) dobiti (kroz, najčešće, desetke ili stotine iteracija) sliku koja odgovara zadanim parametrima ili uputi, tj. zahtjevu (eng. promptu). Ovo se naziva difuzijski model jer se upravo i događa proces difuzije, odnosno iz šuma na početku se oblici i slike „pretvaraju“ u ono što nam je „potrebno“ (postupnim uklanjanjem i dodavanjem šuma). Točnije, kroz niz iteracija šum se gubi, a željeni oblik (primjerice, astronaut koji jaše konja na Marsu) se „dobiva“.

### DAE i U-Net:

Ključan dio otklanjanja šuma (eng. denoisinga) predstavlja arhitektura dizajnirana za denoising – U-Net. U-Net je zamijenio, kao efikasnije rješenje, autoenkoder za otklanjanje šuma (eng. Denoising autoencoder - DAE). To bi bila, sama po sebi, neuronska mreža za „dekodiranje“ šuma u sliku pri fazi generiranja. Ključna napomena bi bila da se ne miješaju DAE i VAE (eng. Variational Autoencoder). Dakle, VAE je također autoenkoder kao DAE, no on se koristi (većinski) samo za pretvaranje slike u latentni prostor i natrag. Međutim, DAE je jedan od ranijih pristupa problemu denoisinga, te ga je danas zamjenio U-Net (u kontekstu modernih difuzijskih modela). Ključna razlika između DAE-a i U-Neta, te razlog zašto „denoiser“ poput DAE-a je „zamijenjen“, je ta što DAE radi (poput VAE-a) kao klasični autoenkoder, te obično obavi denoising u jednom koraku (učenje rekonstrukcije iz šuma) dok U-Net uči procijeniti koliko šuma treba ukloniti u svakom koraku generacije slike. Dakle, U-Net ne radi proces generacije slike po iteracijama „sam po sebi“, već se koristi u denoising procesu moderniji difuzijskih modela koji se odvijaju u više koraka. Također je iznimno korisna U-Netova značajka skip konekcija. Skip konekcije omogućuju da se informacije iz ranijih slojeva mreže (tijekom enkodiranja) direktno prenesu na odgovarajuće kasnije slojeve (tijekom dekodiranja). Primjerice, značajke visoke razlučivosti za generiranje realističnih slika. Zbog U-Netovog „preuzimanja tržišta“, za potrebe ovog rada ćemo većinom njega dalje spominjati. Dakle, U-Net radi na način da se iz šuma prepoznaju oblici i dijelovi slike, predvidi se koji bi bio idući najefikasniji korak za „očistiti“ dio/dijelove slike koji su „onečišćeni“ šumom, te se usput postupno popunjavaju i dodavaju detalji slici i poboljšava se cjelokupna strukturu na smisleniji način. Kao što je već spomenuto, ovaj se proces odvija u više iteracija. Važno je napomenuti kako se U-Net koristi i pri treniranju u svrhu, ne samo dekodiranja, već i enkodiranja šuma.

Iz prethodnog se da „izvući“ (općenita) formula za dodavanje šuma, odnosno forward process, koja opisuje prijelaz iz slike s manje šuma u sliku s više šuma:

Vrijednost predstavlja sliku u koraku t (s više šuma nego što ima prethodni korak), predstavlja količinu šuma dodanog u koraku t. Bitno je naglasiti da se prošli korak množi s kako bismo zadržali prošle dijelove slike. je jedinična matrica čija je svrha jednoliko dodavanje šuma na sve piksele na slici.

To bi bila osnovna ideja kreiranja šuma, no za potrebe ovog seminara bitniji nam je inverz ovog procesa, tj. otklanjanje šuma iz slike, ili reverse process. Iduća formula predstavlja modeliranu distribuciju za uklanjanje šuma:

Gornja formula je općeniti oblik reverse processa, gdje je najbolja procjena kako bi (prošla iteracija slike) mogla izgledati (za ovu je procjenu, kao što smo govorili, zaslužan je U-Net). predstavlja sigurnost u tu procjenu. Bolje rečeno, što je ova varijabla bliža nuli, to je model sigurniji u procjenu, a ako je ova varijabla relativno veća, tada je sigurnost u procjenu manja i samim tim će se morati dodati više šuma u toj iteraciji slike, to jest, nasumičnost je veća. Točnije, kontrolira nasumičnost, odnosno raznolikost pretrage kroz šum.

Formulom za , zadanom iznad, je prva koju zapravo i računamo u cijelom procesu zato što uz pomoć nje samo procjenjujemo kako bi mogla izgledati završna, „čista“ slika (odnosno slika bez šuma). Dva važnija, dosad nespomenuta, parametra ove formule su i . Prvi parametar koristimo kao procjenu „količine“ šuma kojeg bi trebalo ukloniti, dok se drugi parametar računa iz , koji se koristi i u forward processu. Ovdje se koristi na isti način, dakle označava „intenzitet“ šuma pri svakom koraku, pa samim time osigurava da postupno i stabilno uklanjamo šum pri svakoj iteraciji generiranja slike. je hiperparametar, a kao što smo vidjeli, je bitan za definiciju intenziteta šuma koji primjenjujemo (ili otklanjamo), pa ga je zato jako važno u početku „dobro“ postaviti.

No, ako je ovo samo procjena izgleda završne umjetne kreacije, kako možemo naći svaki sljedeći korak u pretrazi kroz cijeli proces uklanjanja šuma?

Ovdje je jedina nova varijabla u odnosu na prethodne formule. Ova varijabla predstavlja nasumični šum koji dolazi iz standardne normalne distribucije (Gauss), odnosno . Ova varijabla je bitna zbog fleksibilnosti i raznolikosti koju pruža (primjerice različita slika za isti prompt). No, poslije ove formule bilo bi dobro spomenuti važnost samplera, te što su oni uopće.

### Sampleri:

Sampleri su algoritmi koji određuju kojim će se putem odraditi pretraga kroz latentni prostor (koji smo pojasnili u prošlom poglavlju). Odnosno, oni kontroliraju iterativno otklanjanje šuma u prostoru (slika). Samim tim su odgovorni za kvalitetu i brzinu denoisinga (koriste se samo tijekom generiranja novih slika/testiranja) i kvaliteti završnog „umjetničkog djela“ (iako, logično, nisu jedini faktor, npr. hiperparametri, arhitektura modela itd.). Čisto matematički – sampleri imaju ogroman utjecaj na vrijednost , te su oni sami odgovorni za poboljšanje ove varijable, tj. njene „preciznosti“ i „brzine“.

Fokusirajmo se sad na specifične samplere koje ćemo poslije iskoristiti i prikazati u primjerima.

Eulerova metoda je numerička metoda za rješavanje diferencijalnih jednadžbi koja, kad se primjeni u našem kontekstu (kontekstu difuzije), koristi procjenu gradijenta šuma kako bi se iterativno otklanjao šum u slici.

Ovdje je funkcija koja pri svakoj iteraciji opisuje kako se mijenja šum. Ova metoda je jedna od češće korištenih i „jednostavnijih“ metoda, primarno zbog svog dobrog omjera brzine i kvalitete. Jedna manja mana Eulerove metode uzorkovanja (sampling) jest manja raznolikost, odnosno „kreativnost“ u rezultatima. Ovaj se problem djelomično „popravi“ u varijaciji ove metode (Euler Ancestral) na način da se pri svakoj iteraciji doda i nasumični šum, no nju se u ovom radu neće spominjati.

Linear multistep method, ili LMS (spada u full denoising strategiju), je metoda posebna po svojstvu po kojem je nazvana - multistep. Naime, kako bi ova metoda napravila što bolju procjenu iduće iteracije, ona ne koristi samo prošli korak kao „pomoć“ (poput Eulerove metode), nego i više prethodnih koraka kako bi metoda ostvarila što bolju preciznost, stabilnost i što je više moguće „glatku“ sliku. Pa, tako imamo:

gdje nam i služe kao težinski koeficijenti za ukomponirati prethodne korake u rezultat za idući, a predstavlja predviđeni šum (zaista kontrolira nasumičnost pretrage, ali se često aproksimira). Iako daje glatke i stabilne slike, u odnosu na Eulerovu metodu, isto tako je i sporiji (jer se, kao što smo spomenuli, prolazi kroz više koraka za cjelokupni izračun).

Zadnja sampling metoda je najbrša metoda među prethodno spomenutima. Denoising diffusion implicit models, ili DDIM (spada u accelerated denoising strategiju), je metoda samplanja koja „preskače“ korake, na način da „predvidi“ kako će „izgledati“ slika bez šuma, te izračunava korake na direktan način. Bitno je spomenuti kako se može koristiti i deterministički (ista slika s istim početnim uvjetima) i stohastički („različita“ slika s istim početnim uvjetima), što je važno za raznolikost generiranih slika.

Jedina nova varijabla koju „vrijedi“ spomenuti je koja „kontrolira“ koliko se vraćamo „čistoj“ slici (predikcija ). Ovim samplingom se ponekad može smanjiti broj koraka i do deset puta. No, nažalost, zbog preskakanja koraka u svom izračunu često se može dogoditi da je kvaliteta, preciznost i stabilnost završne umjetničke kreacije lošija u odnosu na prethodna dva spomenuta samplera (Euler i LMS).

### Primjeri:

Slika na kojoj se prikazuje pastuh, kobila, konjska oprema, Uzde

Sadržaj generiran umjetnom inteligencijom može biti netočan.

Slika iznad je rezultat generiranja slike (Python kodom) uz pomoć predtreniranog Stable Diffusion modela. Za generiranje se također se koristila guided denoising strategija, odnosno se koristio text(prompt)-to-image model. Dani prompt je bio (kao što će biti za ostatak primjera): „a photo of an astronaut riding a horse on mars“, te je za ovaj konkretan primjer specificirano pedeset (eng.) „interference steps“ tj. koraka odšumljivanja.

U sljedećem primjeru ćemo promotriti kako bi izgledao proces „čišćenja“ slike (otklanjanja šuma) koji smo spominjali u prošlim poglavljima. Za prikazati proces generiranja smo trebali redom „ručno odšumljivati“ slike. Prošlo se kroz sto koraka „odšumljivanja“, odnosno denoisinga, no bilježio se svaki dvadeseti korak:

Slika na kojoj se prikazuje umjetničko djelo, snimka zaslona, mozaik

Sadržaj generiran umjetnom inteligencijom može biti netočan.

Jasno se u gore prikazanim slikama vidi kako iz nasumičnog šuma (koji naliči šumu televizije) „izlazi“ oblik i boja završne slike.

Za ovaj zadnji (i možda najvažniji) primjer bit će prikazana uspotredba tri sampling algoritma koja smo spominjali ranije (Euler, LMS i DDIM). Primjer će uključivati prikaz tri slike (s istim promptom kao i kod prošlih primjera), kao i statistike vezane uz njih, te samim tim i svaki priloženi sampler (ove statistike postoje kao rezultat izvršenog koda, no ovdje će biti zapisane u zasebnu tablicu zbog preglednosti):

Slika na kojoj se prikazuje snimka zaslona, Uzde, vanjski, Konjički sport

Sadržaj generiran umjetnom inteligencijom može biti netočan.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Vrijeme do generirane slike(s)  *(eng. Time (s))* | Broj koraka generiranja  *(eng. Interference Steps)* | SSIM u odnosu na Eulerov sampler  *(eng. SSIM vs First)* |
| *Euler* | 1.80 | 10 | N/A |
| *LMS* | 1.89 | 10 | 0.4277 |
| *DDIM* | 1.77 | 10 | 0.5801 |

Rezultati iz gornje tablice skoro pa u potpunosti odgovaraju teoriji koju smo pokrili vezanu uz ova tri sampling algoritma. Primjerice, LMS je najsporiji jer gleda više koraka „unazad“ kako bi se riješio šuma, no zato njegova slika „ljepše“ izgleda. Zato je DDIM imao najbrže vrijeme, no, vidljivo je lošija kvaliteta slike. Samo se sad možda pitate što predstavlja treći podatak koji pratimo. Pa, to bi bio SSIM (eng. Structural Similarity Index Measure). Radi se o metrici koja prati razliku između slika (npr. osvijetljenje, kontrast itd.). Na ovom su se primjeru uspoređivale ostale slike (LMS, DDIM) s prvom generiranom (Euler, zato nema SSIM podatka za Eulerov sampling). Slike su sličnije je što je broj bliže jedinici. Analogno, što je broj bliže nuli, to su slike više različite.

## Izazovi i ograničenja pretrage u AI umjetnosti

### Kombinatorička eksplozija:

Iako se naziv ovog problema čini strašnim, on samo predstavlja izazov u generiranju slika s relativno dužim tesktualnim uputama (promptovima), odnosno s većim (traženim) brojem parametara. Reklo bi se da je broj mogućih kombinacija estetike, stilova, oblika i karakteristika s kojim umjetna inteligencija raspolaže „beskonačan“. Kao što smo spomenuli, iako je prostorna distribucija „veća“, latentni prostor kroz koji se pretražuje je svakako ogroman za pretraživati. Pa naravno da će AI modelu (npr. difuzijskom) biti lakše „nacrtati“ samo traženog konja, umjesto astronauta koji jaše konja na Marsu. U slučaju prompta koji smo mi zatražili u našim primjerima, AI je morao pretražiti ekstremno velik prostor kombinacija ovih parametara (npr. konj, Mars, astronaut), dok će, na primjer, za prompt koji traži samo konja prostor pretrage latentnog prostora biti logično dosta „uži“.

### Balansiranje između kreativnosti i prepoznatljivosti:

Pošto AI ne može u potpunosti „shvatiti“ kontekst slike koju kreira, ovaj izazov onda predstavlja dosta velik problem. Naime, ako se ograniči model u svojoj kreativnosti, onda bi, primjerice, onaj naš „astronaut koji jaše konja na Marsu“ izgledao dosta dosadno i beživotno (u kontekstu slika). S druge strane, ako model pretjera s inovativnošću ili kreativnošću, onda bi astronaut i konj mogli dobiti, na primjer, po dodatnu glavu, svaki. Ovo se (kod difuzijskih modela) regulira na način da se doda dodatni parametar, (eng.) guidance scale, koji služi kao „regulator“ ovog problema. Što je guidance scale viši, to će generirana slika biti realističnija, odnosno manje inovativna/kreativna. Što je ovaj scale manji, to će slike biti sve kreativnije, odnosno generirane slike će biti sve apstraktnije. Ovaj se parametar da „naštimati“ ovisno o našim potrebama, odnosno očekivanjima sa strane modela.

### Ograničenja računalne snage i vremena generiranja:

Upravo se zbog ovog razloga u odjeljku ovog rada sa primjerima koristio predtrenirani difuzijski model. Pogotovo je kod difuzijskih modela slučaj da će se željena slika generirati kroz velik broj iteracija, no kao, što je spomenuto ranije, ovaj proces je dosta računalno i vremenski „skup“ (pogotovo kod treniranja modela).