VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
STUDIJŲ PROGRAMA: INFORMATIKA

Generatyviniai besivaržantys neuroniniai tinklai vaizdams iš teksto generuoti
Generative Adversarial Networks for Synthesis of Images from Text

Baigiamasis magistro darbas

Atliko: Greta Juškaitė
VU el. p.: greta.juskaite@mif.stud.vu.lt

Vadovas: Prof. Dr. Olga Kurasova
Recenzentas: Prof. Dr. Aistis Raudys

Vilnius
2023

Santrauka

Vaizdus iš teksto galima generuoti pasitelkiant GAN, bet kaip nuspręsti ar GAN sugeneruotas vaizdas yra realistiškas. Dažniausiai gauti rezultatai yra lyginami vizualiai, bet jei tokių rezultatų yra šimtai ar tūkstančiai ir juos visus reikia įvertinti. Čia atsiranda problema – kaip automatiškai įvertinti ar sugeneruotas vaizdas yra realistiškas. Kad nereikėtų vertinti sugeneruotų vaizdų vizualiai, nes skirtingi žmonės gali skirtingai įvertinti tą patį vaizdą bei vizualiai vertinant skirtingu paros metu vaizdų vertinimo rezultatai gali priklausyti nuo emocinės būklės ar požiūrio, pvz., jei nuotaika geresnė, visi sugeneruoti vaizdai gali pasirodyti žymiai geriau atvaizduojantys objektus nei buvo vakar arba du žmonės vertina tą patį sugeneruotą vaizdą tuo pačiu metu, vienam jis gali pasirodyti realistiškai/tiksliai atvaizduojantis objektą, kitam gali trūkti sugeneruotame vaizde objekto detalių, dėl to vaizdą gali aprašyti, kaip netikslų. Taip pat vizualinis vertinimas gali būti daug laiko ir resursų reikalaujantis procesas, jei yra vertinamas didelis kiekis sugeneruotų vaizdų. Dėl to šiame darbe bus bandoma sukurti papildomą procesą, kuris pagal tam tikrus suteiktus kriterijus galėtų peržiūrėti visas turimas nuotraukas ir kiekvieną iš jų galėtų įvertinti automatiškai pagal vienodus kriterijus, taip automatizuojant sugeneruotų vaizdų kokybės vertinimą neįsikišant žmogui.

Summary

It is possible to generate images from text using GAN, but how to decide whether the GAN-generated image is realistic. Usually the results are compared visually, but if there are hundreds or thousands of such results and they all need to be evaluated. The problem here is how to automatically assess whether the generated image is realistic. To avoid having to evaluate the generated images visually, as different people may evaluate the same image differently and the results of visual evaluation at different times of the day may depend on the emotional state or attitude, e.g, If the mood is better, all the generated images may appear to be a much better representation of the objects than they were yesterday, or if two people are evaluating the same generated image at the same time, one person may find it a realistic/accurate representation of the object, while the other person may find that the generated image lacks details of the object, which may lead to a description of the image as not accurate. Also, visual assessment can be a time and resource consuming process if a large number of generated images are assessed. For this reason, this thesis will attempt to develop an additional process that can view all available images according to some given criteria and automatically evaluate each of them according to the same criteria, thus automating the evaluation of the quality of the generated images without human intervention.

TURINYS

LENTELIŲ IR PAVEIKSLŲ SĄRAŠAS

LENTELĖS

PAVEIKSLAI

Įvadas

Išgirdus tokius žodžius, kaip vaizdų generavimas iš teksto pirmiausia kyla klausimas, kas tai yra? Tai būtų galima laikyti tokiu procesu, kurio metu yra skaitomos pasakos, istorijos ar kitoks tekstinis šaltinis ir mintyse kyla vaizdai pagal atitinkamus žodžius ir vystantis istorijai vaizdai keičiasi, sudaro nenutrūkstamą filmą, bet pasąmonėje matomi vaizdai visada atitinka perskaitytus tekstus. Šis gebėjimas yra natūralus žmogaus smegenų veiklos procesas, kad ne visada susimąstoma, kad šį procesą galima apibūdinti, kaip vaizdų generavimą iš teksto. Tekstinių šaltinių įsivaizdavimas yra ne tik geras pavyzdys norint apibūdinti vaizdų generavimą iš teksto, bet ir labai svarbus procesas kognityviniams įgūdžiams lavinti, tokiems, kaip dėmesio išlaikymui, gebėjimui suvokti vizualinę informaciją, informacijos įsisavinimui ir kt. [KGT01]. Remiantis žmogaus sugebėjimu įsivaizduoti buvo sukurtas kompiuterinis architektūrinis sistemos sprendimas, kuris gali susieti tekstą su atitinkamais objektais, o tai jau yra didelis žingsnis kuriant intelektualias sistemas, kurios gali veikti panašiai, kaip ir žmogaus smegenys [FHR+21].

Per pastaruosius keletą metų vaizdo apdorojimo, kompiuterinės regos technologijos labai pažengė į priekį. Viena iš šių technologijų yra vaizdo generavimo ir jau sukurtų paveikslų manipuliavimo/keitimo. Vaizdo generavimas yra svarbus ir praktiškai pritaikomas procesas žaidimuose, virtualioje realybėje, kompiuteriniame dizaine, nuotraukų redagavime ir kt. [FHR+21]. Tradiciškai išsami vaizdinė informacija apie vienokį ar kitokį daiktą/objektą buvo pateikiama išskiriant turimo objekto atributus [FEH+09], t.y., kai objektas yra atvaizduojamas, kaip mažesnių struktūrų tarpusavio junginys, kaip pavyzdys galėtų būti namas. Jei objektas yra kvadrato ar stačiakampio formos, kuris turi trikampį (stogą), toje pačioje formoje dar yra papildomai bent keli kvadratai (durys, langas), tai yra labai didelė tikimybė, kad nuotraukoje yra namas. Vėliau buvo pristatytas modelio adaptavimas naujai užduočiai nepateikiant pavyzdžių (angl. *zero-shot learning;* trump. ZSL) [YTT+14]. Kas yra labai naudinga, nes dažnu atveju iškyla problema – duomenų rinkinių *(angl. training set*) trūkumas, norint tinkamai apmokyti algoritmą atpažinti kategorijas. Bet skirtingoms kategorijoms apibūdinti yra labai daug tekstinių išteklių enciklopedijose, žodynuose, straipsniuose ir kituose interneto ištekliuose, dėl to ZSL remiasi panašumu (vaizdiniu arba semantiniu) tarp matytų ir nematytų klasių arba aprašo nematytas klases naudojantis išmoktų semantinių vaizdinių požymių žodyno terminais [ESE13].

Dar po kurio laiko buvo sukurtas sąlyginio vaizdo generavimas. Jo tikslas buvo sugeneruoti objekto vaizdą iš detalaus aprašymo [XJK+16], nors naudoti tekstą, kaip atspirties tašką yra sunku, nes tiek skirtingose kalbose, tiek vienoje kalboje tą patį vaizdą/objektą galima apibūdinti skirtingomis žodžių kombinacijomis, tuomet gali atsitikti taip, kad aprašytas objektas bus tas pats, bet sugeneruotas vaizdas jau gali neatitikti aprašyto objekto. Kad sumažinti riziką gauti netikslų

vaizdą galima papildomai pasitelkti gilųjį mokymąsi. Gilusis mokymasis gali turimą tekstą išskaidyti į žodžių bei simbolių seką ir naudojantis grandinės taisykle bandyti spėti sekantį simbolį remiantis prieš tai buvusiu simboliu. Šis procesas gali padėti nuspręsti ar generuojamas objekto vaizdas turėtų būti toks pat ar kitoks remiantis duotais aprašais [STF+16]. Aprašytas sąlyginis daugiarūšis modalumas, kai yra naudojamas ne tik vaizdo generavimas, bet ir gilusis mokymas, kuris suteikia daugiau tikslumo, yra labai natūralus generatyvinių besivaržančių neuroninių tinklų, arba trumpiau GAN, pritaikymas [GPM+14], į kurį ir bus orientuotas šis darbas.

Generatyviniai besivaržantys neuroniniai tinklai sukėlė didelį susidomėjimą ir pažangių tyrimų pastangas generuoti vaizdus. Pats GAN veikimas buvo suformuluotas, kaip dviejų žaidėjų žaidimas, kuriame dalyvauja du konkuruojantys dirbtiniai neuroniniai tinklai. Generatorius yra išmokytas gaminti tikroviškus vaizdus, o diskriminatorius yra išmokytas atskirti tikrus vaizdus nuo sugeneruotų. Generatoriaus mokymo tikslas yra apgauti diskriminatorių [FHR+21], todėl sugeneruotus vaizdus būna sunku atskirti nuo realaus pasaulio vaizdų. Toks vaizdų generavimo būdas buvo pritaikytas gausybei užduočių atlikti, tokių kaip: vaizdo generavimas iš teksto, aukštos kokybės (*angl. high-resolution*) vaizdų išgavimas iš mažos skiriamosios gebos nuotraukų (*angl. low-resolution*), vaistų, kurie gali padėti išgydyti ligą, nustatymas, objektų atpažinimas, vaizdų, kuriuose yra tam tikras raštas ar struktūra, išgavimas, veido bruožų manipuliacija, anime personažų generavimas ir kt. [AKK19]. Specifiškai vaizdų generavimas iš teksto šiuo metu gali būti naudojamas iliustruoti tekstą, sumažinti skirtingų kalbų tekstų vertimo problemą. Ateityje, kai ši technika bus giliau išstudijuota, vaizdų generavimą iš teksto būtų galima pritaikyti įvairesnėse srityse, tokiose kaip [SS21]:

- namų interjerą būtų galima peržvelgti tiesiog aprašant savo mintis kompiuteryje užuot leidus laiką ieškant dizaino katalogų, ar nuotraukų, kuriose vaizduojama, kaip baldas turėtų atrodyti;
- nusikaltimo scenos galėtų būti sukurtos tik su aprašymu;
- kadangi trūksta įvairių krypčių medicininių nuotraukų, tokiems vaizdams generuoti galima būtų taikyti tą pačią techniką;
- būtų galima pritaikyti vaizdų generavimą iš teksto žmonių, kurie turi negalią, mokymui.

Generatyviniai besivaržantys neuroniniai tinklai taip pat gali turėti variantų, kurie papildomai turi pridėtų patobulinimų. Vienas iš tokių yra StackGAN, šis GAN yra taikomas generuoti realistiškus vaizdus, kurie yra panašesni į nuotraukas nei, kad į generuotus vaizdus [HTH+17], taip pat yra CycleGAN, kuris yra dažniau taikomas medicinos srityje [JYT+19] arba bandant nuspėti ateities įvykius [YM19], bet tai tikrai nėra visi GAN, kurie šiuo metu yra sukurti.

Susipažinus, kas yra vaizdų generavimas iš teksto naudojant generatyvinius besivaržančius neuroninius tinklus galima teigti, kad atlikti vaizdo sugeneravimo iš teksto užduotį galima pasinaudojant įvairius GAN, bet kaip nuspręsti, ar tikrai GAN sugeneruotas vaizdas yra realistiškas kokybiniu atžvilgiu. Straipsniuose sugeneruotų vaizdų kokybė dažniausiai yra vertinama vizualiai, bet tam, kad įvertinti visada reikia žmogaus įsikišimo ir kiekvienas žmogus skirtingai apibrėžia, kas yra realistiškai sugeneruotas vaizdas. Čia atsiranda problema – kaip automatiškai įvertinti, kuris sugeneruotas vaizdas yra geresnis/tikslesnis.

Darbo aktualumas ir naujumas – kad nereikėtų vertinti sugeneruotų vaizdų vizualiai, nes skirtingi žmonės gali skirtingai įvertinti tą patį vaizdą bei vizualiai vertinant skirtingu paros metu vaizdų vertinimo rezultatai gali priklausyti nuo emocinės būklės ar požiūrio, pvz., jei nuotaika geresnė, visi sugeneruoti vaizdai gali pasirodyti žymiai geriau atvaizduojantys objektus nei buvo vakar arba du žmonės vertina tą patį sugeneruotą vaizdą tuo pačiu metu, vienam jis gali pasirodyti realistiškai/tiksliai atvaizduojantis objektą, kitam gali trūkti sugeneruotame vaizde objekto detalių, dėl to vaizdą gali aprašyti, kaip netikslų. Taip pat vizualinis vertinimas gali būti daug laiko ir resursų reikalaujantis procesas, jei yra vertinamas didelis kiekis sugeneruotų vaizdų. Tai galėtų padaryti papildomas procesas, kuris pagal tam tikrus suteiktus kriterijus galėtų peržiūrėti visus turimus sugeneruotus vaizdus ir kiekvieną iš jų galėtų įvertinti ar kiekvienas iš sugeneruotų vaizdų yra atvaizduojami realistiškai, taip automatizuojant sugeneruotų vaizdų kokybės vertinimą neįsikišant žmogui.

Darbo tikslas – sukurti automatinį generatyvinių besivaržančių neuroninių tinklų sugeneruotų vaizdų iš teksto kokybinį vertinimo metodą. Kad įvykdyti darbo tikslą, keliami šie **uždaviniai**:

1. Analitiškai apžvelgti naujausius generatyvinių besivaržančių neuroninių tinklų vaizdams iš teksto generuoti variantus bei gautų rezultatų vertinimo metodikas.
2. Išrinkti kelis perspektyviausius variantus ir atlikti išsamesnę jų eksperimentinę analizę.
3. Pasiūlyti metodą, kuris automatiškai įvertintų sugeneruotų vaizdų kokybę.
4. Atlikti eksperimentinius sukurto metodo tyrimus.

Laukiami rezultatai: bus parinkti ir analitiškai apžvelgti naujausi generatyvinių besivaržančių neuroninių tinklų variantai bei jų taikomos metodikos rezultatams vertinti. Išrinkti keli perspektyviausi variantai ir atlikta išsamesnė jų eksperimentinė analizė. Atlikus eksperimentinę analizę bus pasiūlytas metodas, kuris galėtų automatiškai įvertinti sugeneruotų vaizdų kokybę. Galiausiai pasiūlytas metodas kokybės vertinimui bus pritaikytas išrinktiems generatyvinių besivaržančių neuroninių tinklų variantams ir naudojant juos atlikti eksperimentiniai tyrimai.

1. Generatyviniai besivaržantys neuroniniai tinklai vaizdams iš teksto generuoti

Šiame skyriuje bus apžvelgtas vaizdų generavimas iš teksto, kas yra generatyvinis modelis ir detaliau aptarta, kaip veikia generatyviniai besivaržantys neuroniniai tinklai, kokie yra naujausi GAN modeliai, naudojami vaizdams iš teksto generuoti.

1.1.Vaizdų generavimas iš teksto

Viena iš labiausiai paplitusių ir sudėtingiausių natūralios kalbos apdorojimo ir kompiuterinės regos problemų yra vaizdų antraščių generavimas: pateikus vaizdą, turi būti sukurtas tekstinis vaizdo aprašymas [HSS+21]. Vaizdo generavimas iš teksto yra atvirkštinė problema: turint teksto aprašymą, turi būti sugeneruotas tą aprašymą atitinkantis vaizdas. Žvelgiant iš aukšto abstrakcijos lygio, šios problemos nesiskiria nuo kalbos vertimo problemų – panaši semantika gali būti užkoduota dviem skirtingomis kalbomis. Vaizdai ir tekstas yra dvi skirtingos „kalbos“, skirtos koduoti susijusią informaciją. Nepaisant to, kad šiuos procesus galima laikyti panašiais, kylančios problemos yra visiškai skirtingos, nes teksto-vaizdo arba vaizdo-teksto konversijos yra labai daugiarūšės. Jei bandoma išversti paprastą sakinį, pvz., „Tai graži raudona gėlė“ į prancūzų kalbą, nėra daug sakinių, kurie galėtų būti tinkami vertimai, bet jei bandoma sukurti šio aprašymo mintį, yra daug galimų vaizdų, kurie atitiktų šį aprašymą. Nors toks daugiarūšis elgesys būdingas ir vaizdų antraštėms, problemą palengvina tai, kad kalba dažniausiai yra nuosekli. Ši struktūra išnaudojama sąlygojant naujų žodžių generavimą pagal ankstesnius (jau sugeneruotus) žodžius [CJ18].

Vaizdų generavimas iš natūralios kalbos turės daug galimų pritaikymų ateityje, kai technologija bus paruošta komerciniam pritaikymui. Žmonės galės sukurti savo namams pritaikytus baldus, tiesiog aprašydami juos kompiuteryje, o ne praleisdami daug valandų ieškodami norimo dizaino. Taip pat ši technologija pasitarnaus ne tik buityje, bet ir įvairiose mokslinėse ir profesinėse šakose, kaip medicinoje, teisėsaugoje, statybose ir t.t.

1.2.Generatyviniai modeliai

Vaizdo generavimo iš teksto uždavinys puikiai atitinka problemos, kurią bando išspręsti generatyviniai modeliai, aprašymą. Šiuo metu geriausius rezultatus išgauna generatyvinių besivaržančių neuroninių tinklų arba sutrumpinus GAN tipo generatyviniai modeliai, bet prieš pristatant plačiau GAN, pirmiausia reikia trumpai apžvelgti, kas yra generatyvinis modelis.

Svarstymui paimkime duomenų rinkinį $S = \{x^{(1)}, \dots, x^{(m)}\}$, kuris yra sudarytas iš m pavyzdžių ir $x^{(1)}$ reikšmė yra vektorius. Konkrečiu atveju $x^{(1)}$ yra vaizdas, užkoduotas, kaip pikselių reikšmių vektorius. Duomenų rinkinys sukuriamas atrenkant vaizdus iš nežinomo

duomenų generavimo paskirstymo P_r, kur *r* reiškia realų. Generatyvinis modelis yra modelis, kuris remiantis P_r duomenų rinkiniu mokosi generuoti pavyzdžius iš skirstinio P_g. Modelio skirstinys P_g yra hipotezė apie tikrąjį duomenų pasiskirstymą P_r [CJ18].

Dauguma generatyvinių modelių aiškiai išmoksta skirstinį P_g, padidindami tikėtiną logaritminę tikimybę $E_{X \sim P_r}\ log\ log\ (P_g(x|\theta))$ modelio parametrų θ atžvilgiu. Intuityviai žiūrint, mokymasis apie didžiausią tikimybę prilygsta didesnės tikimybės masės įtraukimui aplink X regionus, kai yra daugiau pavyzdžių iš S ir mažiau regionuose, kuriuose yra mažiau pavyzdžių. Galima matyti, kad logaritminės tikimybės padidinimas yra lygiavertis Kullback-Leibler divergencijos $KL(P_r \,||\, P_g) = \int_X \quad P_r\ log\ log\ \frac{P_r}{P_g}\ dx$ sumažinimui, darant prielaidą, kad P_r ir P_g.yra tankiai. Viena iš vertingų šio metodo savybių yra ta, kad nereikia žinoti apie nežinomą P_r, nes galimybes galima apytiksliai apskaičiuoti naudojant pakankamai imčių pagal silpną didelių skaičių dėsnį [CJ18].

Generatyviniai besivaržantys neuroniniai tinklai yra dar vienas generatyvinio modelio tipas, kuriam yra taikomas kitoks požiūris, pagrįstas žaidimų teorija.

1.2.1. Generatyviniai besivaržantys neuroniniai tinklai

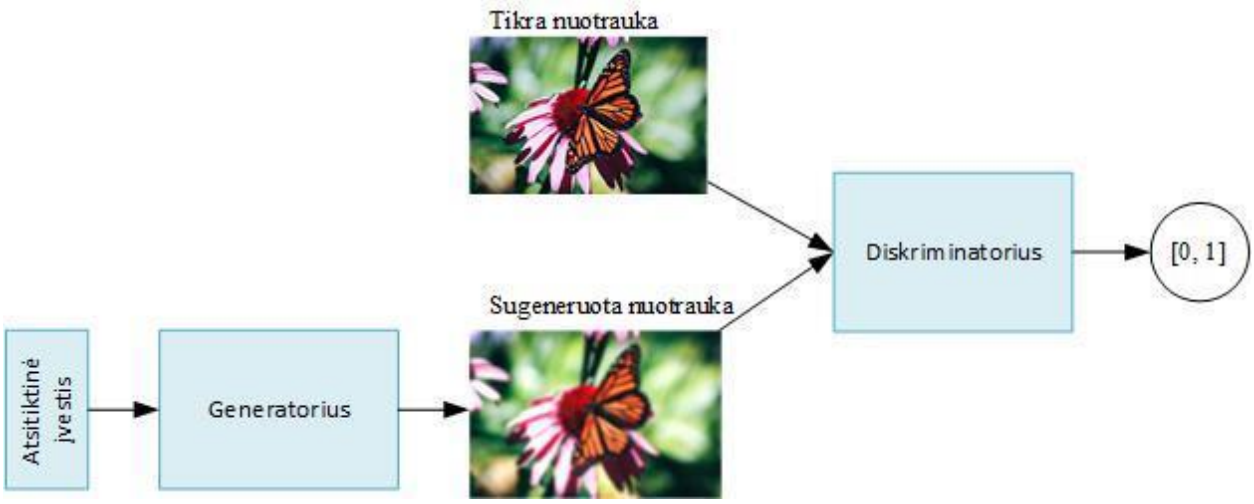
Generatyviniai besivaržantys neuroniniai tinklai (GAN) išsprendžia daugumą jau esamų generatyvinių modelių trūkumų:

- GAN generuojamų vaizdų kokybė yra geresnė nei kitų modelių.
- GAN nereikia išmokti aiškaus tankio P_g.
- GAN gali efektyviai ir lygiagrečiai generuoti vaizdus: vaizdas gali būti generuojamas lygiagrečiai, o ne po vieną pikselį.
- GAN yra lankstūs tiek nuostolio funkcijų, tiek vaizdus generuojančio tinklo topologijos atžvilgiu.
- Kai GAN susilieja, $P_g = P_r$. Ši lygybė negalioja kitų tipų modeliams, kuriuose yra šališkas optimizuojamų nuostolių įvertis.

Nepaisant to, šie patobulinimai atsiranda dėl dviejų naujų reikšmingų problemų: nestabilumo treniruočių metu ir jokių konvergencijos požymių nebuvimo. GAN mokymas yra gana stabilus konkrečioms architektūroms ir kruopščiai atrinktiems hiperparametrams, tačiau tai toli gražu nėra idealu [CJ18].

GAN sistema yra pagrįsta žaidimu, kurį žaidžia du subjektai: diskriminatorius (taip pat vadinamas kritiku) ir generatorius. Neoficialiai žaidimą galima apibūdinti taip – generatorius sukuria vaizdus, kurie turi atrodyti kuo realistiškiau, ir taip darydamas bando įtikinti diskriminatorių, kad sukurti vaizdai yra tikri. Diskriminatorius, gavęs vaizdą, siekia nustatyti, ar

vaizdas yra tikras, ar sugeneruotas. Taip nuolatos žaisdami šį žaidimą abu žaidėjai siekia tikslo tapti kuo geresniais, o tai reiškia, kad generatorius išmoks generuoti tikroviškus vaizdus [SZX+16], 1 paveiksle pateikiama, kaip šis procesas atrodo vizualiai.



1 pav. GAN veikimas

Kaip galima matyti 1 paveiksle, generatorius paėmęs atsitiktinę įvestį sukuria vaizdą. Tuomet diskriminatorius paima tikrą ir sugeneruotą vaizdus kaip įvestį ir išveda tikimybę, kiek abu vaizdai yra tikroviški. Dažna analogija yra meno klastotojas (generatorius), kuris bando padirbti paveikslus, ir meno tyrėjas (diskriminatorius), kuris bando aptikti imitacijas.

Šią tikimybę taip pat galima sumodeliuoti matematiškai. Tegul S yra imčių $x^{(i)}$ pavyzdinių duomenų rinkinys, kuris priklauso kompaktinei metrinei aibei X, tokiai kaip vaizdų erdvė $[-1,1]^n$. Diskriminatorius išmoksta parametrinę funkciją $D_\omega : X \to [0,1]$, kuri kaip įvestį paima vaizdą x ir išveda tikimybę, kurią priskiria vaizdui, kad jis yra tikras. Tegul Z yra atsitiktinio vektoriaus Z1 diapazonas su paprastu ir fiksuotu skirstiniu, pvz., $p_{Z1} = N(0, I)$. Generatorius išmoksta parametrinę funkciją $G\theta : Z \to X$, kuri atsitiktinio vektoriaus Z būsenas susieja su atsitiktinio vektoriaus X būsenomis. $X \sim P_g$ būsenos atitinka generatoriaus sukurtus vaizdus. Taigi generatorius išmoksta atvaizduoti triukšmo vektorių [CJ18].

Lengviausias būdas apibrėžti ir analizuoti šį žaidimą yra nulinės sumos žaidimas, kuriame D_ω ir $G\theta$ yra dviejų žaidėjų strategijos. Tokį žaidimą galima apibūdinti vertės funkcija $V(D, G)$, kuri šiuo atveju reiškia diskriminatoriaus atlygį. Diskriminatorius nori maksimaliai padidinti V, o generatorius nori jį sumažinti. V aprašytas atlygis turi būti proporcinga D gebėjimui atskirti tikrus ir netikrus pavyzdžius [SZX+16].

$$V(D, G) = EE_{X \sim P_r}\big[log\big(D(x)\big)\big] + E_{Z \sim p_Z}[log(1 - D(G(z)))] \quad (1)$$

1 lygtyje pateikiama vertės funkcija, kuri iš pradžių buvo pasiūlyta [GPM+14] straipsnyje. *V(D, G)* tampa didesnis, kai diskriminatorius gali atskirti tikrus ir netikrus pavyzdžius. Iš kitos

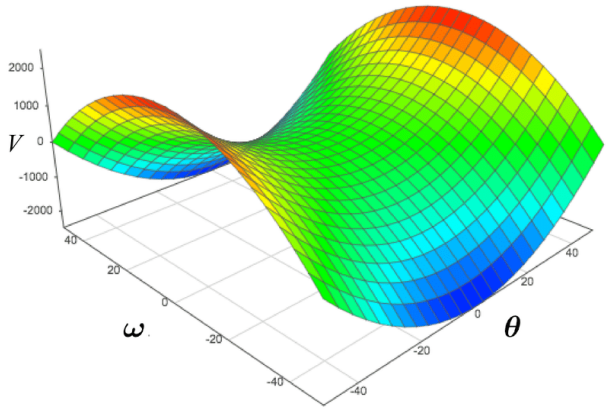
pusės, V tampa žemesnis, kai generatorius veikia gerai, o diskriminatorius negali tinkamai atskirti realių ir sugeneruotų pavyzdžių.

Skirtumas nuo didžiausios tikimybės modelių yra tas, kad generatoriaus sugeneruotų imčių numatomas pasiskirstymas P_g, nustatomas pagal tam tikrą θ reikšmę. P_g negalima aiškiai įvertinti. Diskriminatorius priverčia generatorių priartinti P_g prie P_r.

Nulinės sumos žaidimas yra tas pats, kaip minimalu-maksimalu vertinimas, todėl optimalius generatoriaus parametrus galima apibūdinti taip, kaip yra pateikiama 2 lygtyje [CJ18].

$$\theta^* = \frac{argmin}{\theta}\frac{max}{\omega} V(D, G) \tag{2}$$

Minimalu-maksimalu optimizavimo sprendimas yra Nešo pusiausvyra (2 pav.)



2 pav. Nešo pusiausvyra [CJ18]

Nešo pusiausvyra atitinka balno, kuris yra pateikiamas 2 paveiksle, tašką erdvėje, kai V yra mažiausias θ ir didžiausias ω atžvilgiu. Nė vienas iš tinklų nėra suinteresuotas keisti savo parametrų [CJ18]. Nešo teorema aprašo neparametrinių funkcijų pusiausvyros žadinančias savybes. Teorema patvirtina, kad žaidžiant dviem žaidėjams šį žaidimą, generatoriaus kuriami vaizdai taps vis tikroviškesni [CJ18].

Nešo teorema (neparametrinė) skirta GAN žaidimui yra realizuojama, kai:

1. *Diskriminatoriaus strategija yra* $D = \frac{P_r}{P_r + P_g}$.
2. *Generatoriaus strategija G yra* $P_g = P_r$

Remiantis minimalu-maksimalu vertinimu, kurios lygtis yra žymima 2 numeriu, galima tikėtis, kad diskriminatorius treniruojamas iki optimalumo fiksuotam generatoriui, vėliau treniruojamas generatorius ir procesas kartojasi. Praktiškai tinklai treniruojami alternatyviai, žingsnis po žingsnio [CJ18].

1.3.Teksto įterpimas

Kad būtų galima naudoti tekstinį vaizdų aprašymą bet kuriame generatyviniame modelyje, jie turi būti vektorizuoti. Šios vektorizacijos paprastai vadinamos teksto įterpimais [CJ18]. Teksto įterpimas yra klasė, su tam tikromis technikomis, kuri pavaizduoja atskirus žodžius, kaip realios vertės vektorius iš anksto nustatytoje vektorių erdvėje. Kiekvienas žodis susietas su vienu vektoriumi, o vektoriaus reikšmės yra išmokstamos, kaip neuroniniame tinkle, todėl ši technika dažnai įtraukiama į gilaus mokymosi sritį [Gol17]. Žodžių vaizdavimas išmokstamas remiantis žodžių vartosena, tai leidžia panašiai vartojamiems žodžiams turėti panašius vaizdinius, natūraliai užfiksuojant jų reikšmę. Nebent būtų aiškiai atvaizduojama, kad skirtingi žodžiai gali turėti skirtingą vaizdavimą, neatsižvelgiant į tai, kaip jie naudojami [Zel15].

Dažnai žodžiams susieti su vaizdais yra pasirenkamas *Skip-Thought Vectors* metodas, kuris yra kalbos modelis naudojamas susieti panašios sintaksės ir semantikos sakinius su panašiais vektoriais. [CJ18].

1.4.Naujausi GAN naudojami vaizdų generavimui iš teksto

Apžvelgus, kas yra ir kaip veikia vaizdų generavimas naudojant GAN bei kaip yra įterpiamas tekstas į generatyvinius modelius reikia išrinkti tuos modelius, kurie bus apžvelgiami detaliau ir bandoma sužinoti, kaip yra vertinami skirtingų sugeneruotų modelių rezultatai.

Norint išsirinkti tinkamus generatyvinius besivaržančius neuroninius tinklus vaizdams iš teksto generuoti reikia apibrėžti reikalavimus, pagal kuriuos bus renkami GAN. Kadangi nagrinėjami turėtų būti naujausi GAN modeliai turi būti atsižvelgiama į modelio naujumą, taip pat šaltinių kiekį, nes turint tik vieną šaltinį bus sudėtinga atlikti gilesnę modelio analizę. Taip pat svarbu yra patikrinti ar modelis turi prieinamą programinį kodą, kuriuo būtų galima remtis atliekant eksperimentinę analizę. Remiantis šiais trimis kriterijais buvo atlikta paieška ir rasti penki modeliai atitinkantys kriterijus, t.y:

- DF-GAN;
- Semantic-Spatial Aware GAN;
- DM-GAN;
- Obj-GAN;
- VQGAN-CLIP;

Šie GAN bus toliau nagrinėjami antrame skyriuje, kuriame bus išsamiau apžvelgti paminėtieji modeliai bei apžvelgta kiekvieno modelio vertinimo logika.

2. Naujausių generatyvinių besivaržančių neuroninių tinklų vaizdams iš teksto generuoti apžvalga

Kaip ir buvo minėta, šiame skyriuje bus apžvelgiami Semantic-Spatial Aware GAN, DF-GAN, DM-GAN, Obj-GAN bei VQGAN modeliai bei tai, kaip yra vertinami rezultatai pasinaudojus vienu ar kitu generatyviniu modeliu.

2.1.DF-GAN

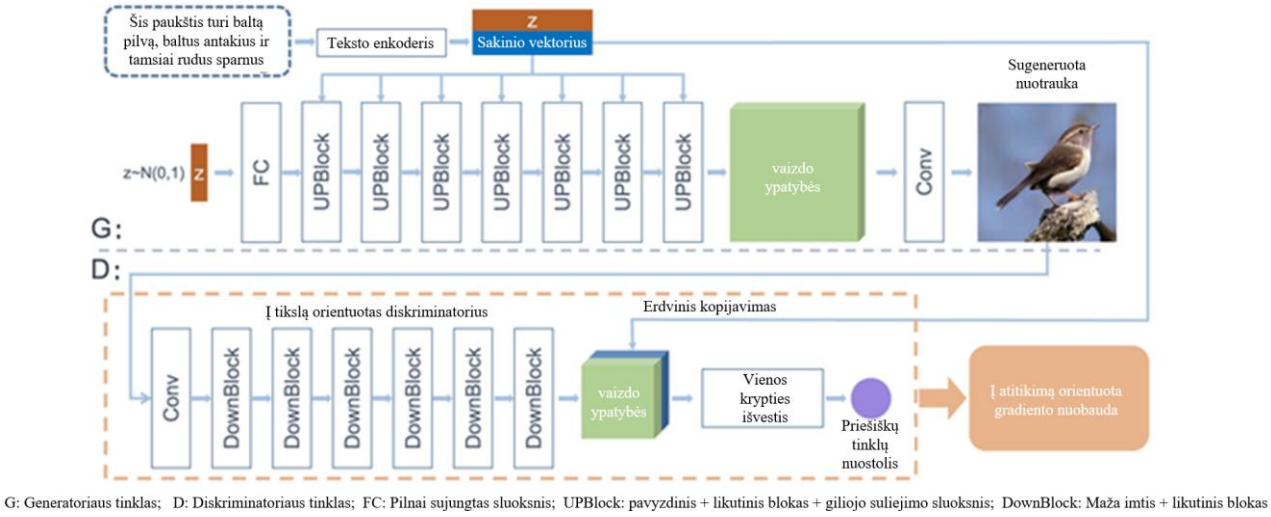
Tai yra pirmasis metodas, kuris bus apžvelgiamas šiame skyriuje. Pirmą kartą DF- GAN modelis straipsniuose pasirodė 2020 metais. Straipsnyje aprašant DF-GAN modelį yra iškeliami trys pagrindiniai trūkumai, kuriuos bando išspręsti nurodytas GAN [TTW+22]:

1) Daugiasluoksnė (*angl. stacked*) architektūra turi ryšius tarp skirtingų vaizdo mastelių generatorių;
2) Egzistuojančiose studijose dažniau taikomi papildomi tinklai apmokant generatyvinius modelius siekiant užtikrinti teksto ir vaizdo semantinį suderinamumą, tačiau tai riboja šių tinklų stebėsenos galimybes.
3) Kryžminiu (*angl. cross-modal*) dėmesiu pagrįstas vaizdo generavimas iš teksto, kuris buvo dažnai taikomas ankstesniuose darbuose, yra apribota keliomis specialiomis vaizdo skalėmis dėl didelių skaičiavimo sąnaudų.

Tam, kad įgyvendinti šiuos tikslus buvo pasiūlytas *Deep Fusion Generative Adversarial Networks* arba trumpiau DF-GAN. Sukuriant šį modelį yra siūloma sukurti naują vieno etapo vaizdų generavimo iš teksto pagrindą, kuris tiesiogiai generuoja aukštos raiškos vaizdus be sąryšių tarp skirtingų generatorių; naują kryžminių duomenų diskriminatorių, kuris susideda iš atitikties suvokimo gradientų bausmės ir vienkryptės išvesties, kuri pagerina teksto ir vaizdo semantinį suderinamumą be papildomų tinklų bei naują gilų teksto ir vaizdą suliejimo bloką, kuris giliau įsišaknija generavimo procese ir sujungia teksto ir vizualinius požymius pilnai [TTW+22].

2.1.1. DF-GAN architektūra

DF-GAN modelis išsiskiria iš kitų tuo, kad šis modelis yra vieno lygmens vadinasi turi tik vieną generatorių ir vieną diskriminatorių [MBG21]. Šis GAN yra sudarytas iš generatoriaus, diskriminatorius ir iš anksto paruošto teksto kodavimo įrenginio, kaip yra vaizduojama 3 paveiksle.



G: Generatoriaus tinklas; D: Diskriminatoriaus tinklas; FC: Pilnai sujungtas sluoksnis; UPBlock: pavyzdinis + likutinis blokas + giliojo suliejimo sluoksnis; DownBlock: Maža imtis + likutinis blokas

3 pav. DF-GAN architektūros modelis [TTW+22]

Kaip galima matyti 3 paveiksle generatorius turi dvi įvestis – sakinio vektorių (*angl. sentence vector*) užkoduotą teksto enkoderiu ir triukšmo vektorių (*angl. noise vector*), paimtu iš Gauso skirstinio, siekiant užtikrinti generuojamų vaizdų įvairovę. Triukšmo vektorius pirmiausia įvedamas į pilnai sujungtą sluoksnį ir pertvarkomas. Tada yra taikoma UPBlock serija, kad būtų galima išskirti vaizdo ypatybes. UPBlock yra sudarytas iš pavyzdinio sluoksnio, likutinio bloko ir giliojo suliejimo bloko (*angl. DF-Block*), kad sujungtų teksto ir vaizdo ypatybes vaizdo generavimo metu. Galiausiai konvoliucijos sluoksnis paverčia išskirtas ypatybes vaizdais [TTW+22]. Sugeneravus vaizdą diskriminatoriaus eilė veikti. Diskriminatorius konvertuoja vaizdus į vaizdo ypatybes naudodamas DownBlock seriją. Tada sakinio vektorius bus pakartojamas dar kartą ir sujungtas su vaizdo ypatybėmis. Norint įvertinti įvesties vizualinį tikroviškumą ir semantinį nuoseklumą, yra prognozuojamas priešiškų tinklų nuostolis (*angl. adversarial loss*). Atskirdamas sugeneruotus vaizdus nuo tikrų pavyzdžių, diskriminatorius skatina generatorių generuoti aukštesnės kokybės ir semantiškai nuoseklesnius vaizdus [TTW+22].

Teksto kodavimo priemonė šiai architektūrai yra dvikryptė ilgalaikė trumpalaikė atmintis (LSTM), kuri iš teksto aprašymo išskiria semantinius vektorius [TTW+22].

2.1.2. DF-GAN rezultatų vertinimas

Straipsnyje [TTW+22] rezultatai yra vertinami pagal kokybę it kiekybę. **Kiekybiniams rodikliams** priklauso pradinis balas (*angl. Inception score*) IS ir Frechet pradžios atstumas (*angl. Frechet Inception Distance*) FID. IS apskaičiuoja Kullback-Leibler (KL) skirtumą tarp sąlyginio ir ribinio skirstinio. Didesnė IS vertė reiškia aukštesnę generuojamų vaizdų kokybę, o kiekvienas vaizdas aiškiai priklauso konkrečiai klasei. FID apskaičiuoja Frechet atstumą tarp sintetinių vaizdų pasiskirstymo ir realaus pasaulio vaizdų iš anksto parengto Inception v3 tinklo funkcijų erdvėje. Priešingai nei IS, tikroviškesnių vaizdų FID yra mažesnis. Kad būtų galima apskaičiuoti IS ir FID

nurodytame straipsnyje buvo generuojama kiekvienam modeliui po 30 000 vaizdų (256 × 256 skiriamoji geba) iš teksto aprašymų, atsitiktinai atrinktų iš bandymo duomenų rinkinio. Tačiau straipsnyje taip pat yra minima, kad ne visų sugeneruotų vaizdų kokybę IS rodiklis gali įvertinti vienodai gerai, dėl to kai kuriais atvejais (specifiškai su COCO duomenų rinkiniu) yra naudojamas parametrų kiekis NoP vietoje IS, kad būtų galima palyginti modelio dydį su turimais metodais. O tam, kad patikrinti kaip **kokybiškai** yra generuojami vaizdai yra pasitelkiama žmogaus rega – sugeneruoti vaizdai vizualiai patikrinami ir aprašoma, kas yra matoma netikslinga.

[BA22] straipsnyje sugeneruoti vaizdai naudojant DF-GAN modelį taip yra vertinami kiekybiškai ir kokybiškai. **Kiekybiniam įvertinimui** yra naudojamos tos pačios skalės – IS ir FID, tačiau nėra taikomas NoP vertinimas, jei vaizdai yra prastai įvertinami naudojant IS. **Kokybiniai vertinimai** atliekami stebint gautus rezultatus ir juos lyginant, kiek detalių yra panašių į tikrus vaizdus.

2.2. Semantic-Spatial Aware GAN

Pirmą kartą Semantic-Spatial Aware GAN modelis straipsniuose pasirodė 2021 metais. Šis modelis buvo apibūdintas, kaip vienas naujausių generatyvinių besivaržančių neuroninių tinklų, skirtų vaizdams iš teksto generuoti, kuris siūlo naują semantinio-erdvinio suvokimo GAN (*angl. Semantic-Spatial Aware GAN*) arba trumpiau SSA-GAN, kuris yra ištisai treniruojamas, kad teksto enkoderis galėtų geriau panaudoti teksto informaciją. Konkrečiai, kūrėjai pristato naują semantinio-erdvinio suvokimo konvoliucijos tinklą, kuris (1) mokosi semantinės-adaptyvios transformacijos, paremtos tekstu, kad būtų efektyviai sujungtos teksto ir vaizdo ypatybės, ir (2) silpnai prižiūrimu būdu (*angl. weakly-supervised*) išmoksta kaukės žemėlapį (*angl. mask map*), kuris priklauso nuo teksto ir vaizdo sujungimo proceso, kad transformaciją būtų galima nukreipti erdviškai [HLY+22].

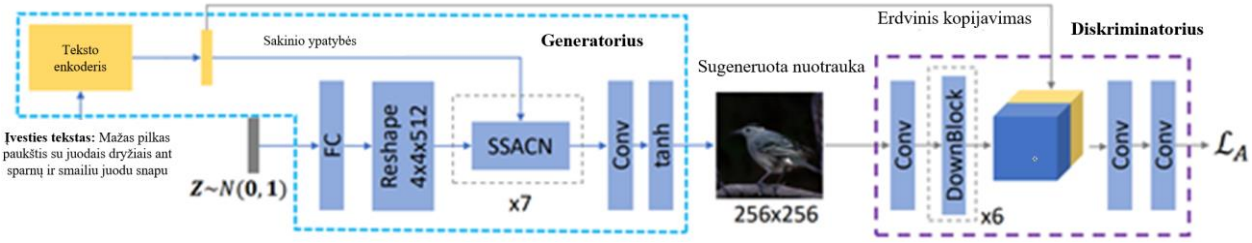
Šis metodas buvo pasiūlytas [HLY+22] straipsnio autorių tam, kad būtų galima pašalinti tokius apribojimus, kaip:

1) Sąlyginio krūvio normalizavimo metodai yra vienodai taikomi visiems paveikslėlio požymių žemėlapiams, ignoruojant vietinius semantinius skirtumus;
2) Mokymosi metu teksto enkoderis yra fiksuotas, o turėtų būti treniruojamas kartu su vaizdo generatoriumi, kad būtų išmokytas geriau atvaizduoti tekstą generuojant vaizdus.

SSA-GAN, kaip ir DF-GAN turi vieno lygmens architektūrą. Vadinasi SSA-GAN kaip ir DF-GAN turi tik vieną generatorių ir vieną diskriminatorių, tačiau skirtingai nuo DF-GAN, teksto enkoderis šiame modelyje yra treniruojamas kartu su vaizdo generatoriumi, kad išmoktų patobulintą tekstinę informaciją vaizdų generavimui [MBG21].

2.2.1. Semantic-Spatial Aware GAN architektūra

Kuriant SSA-GAN modelį buvo remtasi jau sukurtu DF-GAN modeliu, pakeičiant UPBlocks, kurie buvo implementuoti DF-GAN modelyje, į originaliai sukurtu būtent SSA-GAN modeliui – SSACN bloku. Semantic-Spatial Aware GAN architektūra yra pateikiama 4 paveiksle.



4 pav. Semantic-Spatial Aware GAN architektūros modelis [HLY+22]

SSA-GAN turi teksto enkoderį, kuris mokosi teksto atvaizdavimo; generatorių, turintį 7 SSACN blokus, skirtus teksto ir vaizdo suliejimo procesui pagilinti ir raiškai pagerinti, bei diskriminatorių, kuris naudojamas įvertinti, ar sugeneruotas vaizdas semantiškai atitinka pateiktą tekstą. SSA-GAN kaip įvestį paima tekstinį aprašymą ir triukšmo vektorių $z \in R^{100}$, paimtą iš normalaus pasiskirstymo, ir išveda 256 × 256 dydžio RGB vaizdą [HLY+22].

SSACN bloke esantis *UpSampling* blokas padvigubina vaizdinių požymių žemėlapių dimensijas. Padidinti vaizdų požymių žemėlapiai siunčiami į kaukės prognozavimo mechanizmą, kuris generuoja kaukės žemėlapį, kuris ne tik nurodo, kur reikia pridėti tekstinės informacijos, bet taip pat veikia kaip svoris, nusprendžiant kiek dar tekstinės informacijos reikia norint sustiprinti vaizdo požymius žemėlapiuose, kad patobulinti vaizdo požymių žemėlapiai būtų semantiškai sąveikaujantys su įvesties tekstu. Prognozuota kaukė yra pridedama kaip erdvinė sąlyga sąlyginio krūvio normalizavime, vadinamame semantiniu-erdviniu sąlyginio krūvio normalizavimu (SSCBN). CBN moduliacijos parametrai nustatomi iš teksto informacijos, o tikėtini kaukės žemėlapiai erdviškai valdo afininę transformaciją, derinant teksto ir vaizdo duomenis. Tokiu būdu jis netaiko sąlyginės partijų normalizacijos visiems vaizdo požymių žemėlapiams vienodai, dėl kurių praleidžiamos vietinės semantikos ypatybės. Likutinis blokas naudojamas tam, kad būtų išvengta pagrindinių vaizdo požymių turinio pokyčių, dėl kurių vaizdo informacija būtų užgožta teksto informacijos [MBG21].

SSA-GAN diskriminatorius yra panašus į DF-GAN modelio diskriminatorių. Šių abiejų modelių diskriminatoriai yra susiję su MA-GP, kuris nurodo generatoriui sugeneruoti foto realistiškus pavyzdžius pagal įvesties tekstinius aprašymus. Kartu yra naudojamas DAMSM modulis, kad pagerintų sugeneruotų vaizdų kokybę ir išlaikytų tvirtą semantinį ryšį tarp įvesties teksto ir išvesties vaizdų, taip pat padėtų mokyti teksto enkoderį kartu su generatoriumi [MBG21].

2.2.2. Semantic-Spatial Aware GAN rezultatų vertinimas

Semantic-Spatial Aware GAN sugeneruotiems vaizdams vertinti [HLY+22] straipsnyje buvo naudojamas pradinis balas (IS) ir Frechet pradžios atstumas (FID), kurie nurodė **kiekybinį** sugeneruoto vaizdo **vertinimą**. IS balui apskaičiuoti iš anksto buvo paruoštas Inception v3 tinklas naudojantis KL skirtumą tarp sąlyginio klasių pasiskirstymo (sugeneruotų vaizdų) ir ribinės klasės pasiskirstymo (tikrųjų vaizdų). Vertinama buvo sugeneravus 30 000 vaizdų, kurių skiriamoji geba yra 256 × 256, atsitiktine tvarka pasirenkant tekstinius aprašymus iš bandymui skirto duomenų rinkinio. Vienam iš duomenų rinkinių (COCO) buvo taikomas tik FID vertinimas, nes iš ankstesnių, straipsnyje pateiktų, tyrimų buvo išsiaiškinta, kad IS metrika visiškai nesugeba įvertinti sugeneruotų vaizdų. O **kokybinis vertinimas** vyko lyginant vizualiai pagal detales sugeneruotas iš skirtingų modelių – vertino fono ryškumą, detalių ryškumą, detalių atitikimą pagal tekstą.

2.3.DM-GAN

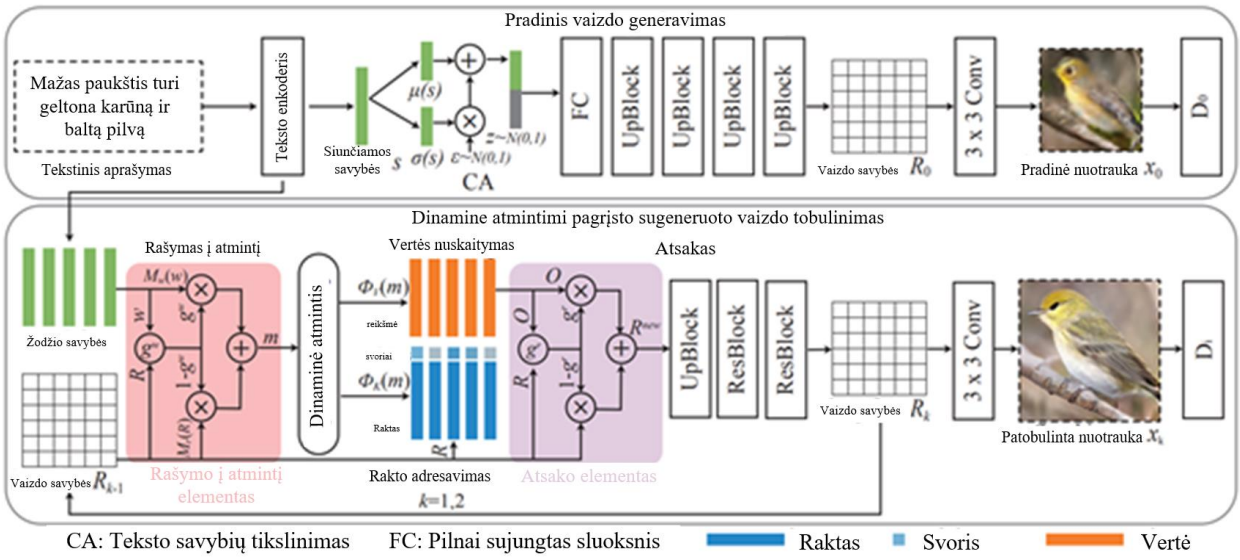
DM-GAN pirmą kartą straipsniuose pasirodė 2019 metais. Prieš realizuojant DM-GAN metodą kūrėjai pirmiausia išskyrė dvi problemas, kurias siekė pagerinti sukuriant numatytą GAN [ZPC+19]:

1) Iki tol sukurti metodai (tokie, kaip GAN-INT-CLS, GAWWN, StackGAN ir kt.) labai priklausė nuo pradinių vaizdų kokybės. Jei pradinis vaizdas nebuvo tinkamai inicijuotas, toliau nurodyti procesai vargu ar galėjo patobulinti vaizdą iki patenkinamos kokybės.
2) Kiekvienas žodis suteikia skirtingą svarbą vaizduojant skirtingą vaizdo turinį, tačiau buvusiuose vaizdo tobulinimo procesuose naudojamas nepakitęs teksto vaizdavimas.

Realizuojant DM-GAN buvo siūloma sukurti dinaminės atminties (*angl. dynamic memory*) generatyvinį besivaržantį tinklą (DM-GAN), kuris galėtų generuoti aukštos kokybės vaizdus. Dinaminės atminties modulis, kuris yra įdiegiamas į DM-GAN, yra skirtas neaiškių vaizdų turiniui patikslinti, kai pradiniai vaizdai nėra aiškiai sugeneruoti. Atminties rašymo vartai *(angl. memory writing gate)* yra skirti pasirinkti svarbią teksto informaciją pagal pradinį vaizdo turinį, todėl šis metodas leidžia tiksliai generuoti vaizdus iš teksto aprašymo. Taip pat šiame GAN yra naudojami atsako vartai *(angl. response gate)*, kad būtų galima adaptyviai sujungti informaciją, nuskaitytą iš atminties ir vaizdo ypatybių [ZPC+19].

2.3.1. DM-GAN architektūra

Dinaminės atminties generatyvinis besivaržantis tinklas, priešingai nuo jau aptartų modelių skiriasi tuo, kad yra daugiapakopis ir tai, kad būtent šis modelis yra sukurtas anksčiausiai. DM-GAN architektūrą galima matyti 5 paveiksle.



5 pav. DM-GAN architektūros modelis [ZPC+19]

Kaip parodyta 5 pav., DM-GAN modelio architektūra susideda iš dviejų etapų: pradinio vaizdo generavimo ir dinamine atmintimi pagrįsto vaizdo tobulinimo. Pradiniame vaizdo generavimo etape, pirma, įvesties teksto aprašymas paverčiamas tam tikru vidiniu atvaizdavimu (sakinio ypatybė *s* ir keletas žodžių ypatybių *W*), naudojant teksto enkoderį. Tada gilusis įprastas generatorius numato pradinį vaizdą x_0 su grubiomis formomis ir nedaug detalių pagal sakinio ypatybę ir atsitiktinio triukšmo vektorių, kuris pateikiamas 3 lygtyje.

$$z: x_0, R_0 = G_0(z, s), \text{ kur } R_0 \text{ yra vaizdo ypatybė} \quad (3)$$

Triukšmo vektorius paimamas iš normalaus skirstinio [ZPC+19]. Dinamine atmintimi grįsto vaizdo tobulinimo etape prie neaiškių pradinių vaizdų pridedamas smulkesnis vaizdinis turinys tam, kad būtų sukurtas tikroviškas vaizdas (4 lygtis).

$$x_i : x_i = G_i(R_{i-1}, W), \text{ kur } R_{i-1} \text{ yra vaizdo ypatybė iš paskutinio etapo} \quad (4)$$

Tobulinimo etapą galima pakartoti kelis kartus, kad būtų gauta tinkamesnė informacija ir būtų sukurtas didelės raiškos vaizdas su smulkiomis detalėmis [ZPC+19].

Dinamine atmintimi pagrįsta vaizdo tobulinimo stadija susideda iš keturių komponentų: rašymo į atmintį, rakto adresavimo, reikšmės skaitymo ir atsako. Rašymo į atmintį operacija išsaugo tekstinę informaciją struktūrizuotoje atmintyje (raktas-reikšmė), tolimesniam naudojimui. Tada naudojamos raktų adresavimo ir vertės skaitymo operacijos atminties modulio funkcijoms nuskaityti, kad būtų patobulintos žemos kokybės vaizdų savybės. Galiausiai atsako operacija yra pritaikyta valdyti vaizdo ypatybių ir atminties skaitymo suliejimą. Taip pat DM-GAN yra taikomi

rašymo į atmintį elementas, kad būtų išryškinama svarbi žodžių informacija pagal vaizdo turinį įrašymo į atmintį žingsnyje bei atsako elementas, kad būtų galima adaptyviai sujungti nuskaitytos iš atminties ir vaizdo ypatybių informaciją [ZPC+19].

2.3.2. DM-GAN rezultatų vertinimas

DM-GAN **kiekybinė** vertinimo metrika straipsnyje [ZPC+19] susideda, kaip ir prieš tai aptartuose modeliuose, iš pradinio balo (IS), Frechet pradžios atstumo (FID) ir papildomai apskaičiuojant R tikslumą. Buvo vertinta 30 000 vaizdų, visi jie buvo sugeneruoti pagal tekstų aprašymus, kurie buvo paimti iš nematytos bandymų aibės. Papildomai šio darbo vertinimui buvo naudojamas R tikslumas, skirtas įvertinti, ar sugeneruotas vaizdas atitinka pateiktą teksto aprašymą. R tikslumas matuojamas ieškant atitinkamo teksto pagal paveikslėlio užklausą. Apskaičiuojamas kosinuso atstumas tarp visuotinio vaizdo vektoriaus ir 100 galimų sakinio vektorių. Kandidatų teksto aprašymai apima R pagrindinę tiesą (*angl. ground truth*) ir *100-R* atsitiktinai atrinktų nesutampančių aprašymų. Kiekvienos užklausos atveju, jei *r* rezultatai būna R reitinguotuose paieškos aprašymuose, tuomet *r* tikslumas yra *r/R*. Praktikoje R tikslumas yra apskaičiuojamas su $R = 1$. Sugeneruoti vaizdai yra padalinami į dešimt skirstinių ir tuomet, kiekvienam iš jų yra apskaičiuojamas R ir imamas gautų balų vidurkis bei standartinis nuokrypis pagal gautus balus.

Nurodytame straipsnyje **kokybinis vertinimas** yra atliekamas vizualiai, atkreipiant dėmesį į vaizdų detalių ryškumą, fono aiškumą išrenkant atsitiktiniu būdu kelis sugeneruotus vaizdus ir juos lyginant su kitų generatyvinių modelių atsitiktinai išrinktais sugeneruotais vaizdais.

Straipsnyje [YYT+21], kuriame yra rašoma apie sugeneruotų vaizdų kokybės ir semantinio nuoseklumo gerinimą taikant kontrastinį mokymąsi taip pat yra atliekami eksperimentai, kuriuose vienas iš lyginamųjų modelių yra DM-GAN bei tokiu pačiu atveju DM-GAN patenka į vertinimo skiltį straipsnyje [RZZ+21], kuriuose yra naudojami tokie patys **kiekybiniai vertinimo metodai**, t.y. IS, FID ir R tikslumą.

Kokybiškai vertinama yra sulyginant kelių GAN sugeneruotus vaizdus su vaizdais, kurie yra sugeneruoti naudojant straipsniuose minimus GAN ar su vaizdais, kurie yra sugeneruoti su tam tikrais patobulinimais. Vertinimui skirti vaizdai yra išrenkami atsitiktiniu būdu iš visų sugeneruotų variantų.

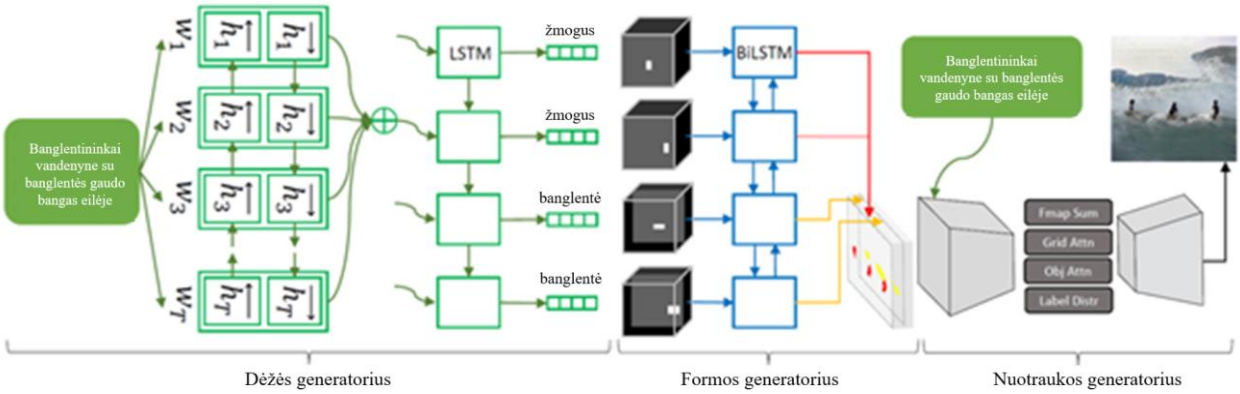
2.4. Obj-GAN

Tai yra dar vienas GAN, kuris pirmą kartą pasirodė 2019 metais. Šis modelis yra sudarytas iš objektais grįstų dėmesingų generatyvinių besivaržančių tinklų (Obj-GAN), kurie leidžia sukurti aukštos kokybės sudėtingus vaizdus su semantiškai prasmingu išdėstymu ir realistiškais objektais.

Šis modelis efektyviai fiksuoja ir panaudoja smulkią žodžių/objektų lygmens tekstinę informaciją generuojant vaizdus. Obj-GAN susideda iš objektu grindžiamo dėmesingo vaizdų generatoriaus ir objektų lygmens diskriminatoriaus poros, taip pat naujo objektu grindžiamo dėmesio mechanizmo. Vaizdų generatorius kaip įvestį įveda teksto aprašą ir iš anksto sugeneruotą semantinį išdėstymą po to iš turimų duomenų yra generuojami didelės raiškos vaizdai per kelias pakopas, kol yra išgryninamas vaizdas. Kiekviename etape generatorius generuoja vaizdo regioną, apimantį nurodytas ribas, sutelkdamas dėmesį į žodžius, kurie yra labiausiai susiję su objektu tose nurodytose ribose, tiksliau, naudodamasis naujuoju į objektą orientuotu dėmesio sluoksniu, jis naudoja klasės etiketę užklausos žodžiams sakiniuose, kad suformuotų žodžių konteksto vektorių ir tada generuoja vaizdo regioną, priklausomai nuo klasės etiketės ir žodžių konteksto vektoriaus. Objekto lygmens diskriminatorius patikrina kiekvieną nurodytą ribą, kad įsitikintų, jog sugeneruotas objektas iš tiesų atitinka iš anksto sukurtą semantinį išdėstymą. Norint efektyviai ir tuo pačiu metu apskaičiuoti visų nurodytų ribų diskriminacijos nuostolius yra taikomas objektinis diskriminatorius pagrįstas greituoju R-CNN, kuris gali apskaičiuoti dvejopą kryžminės entropijos nuostolį kiekvienai nurodytai ribai [LZZ+19] .

2.4.1. Obj-GAN architektūra

Obj-GAN, kaip ir DM-GAN yra daugiapakopis. Šis modelis teksto į vaizdą generavimo procesą atlieka dviem etapais: pirmiausia sugeneruoja semantinį išdėstymą (klasių etiketes, ribojančius langelius, ryškių objektų formas) ir tada sugeneruoja vaizdą. Vaizdo generavimo etape objektais pagrįstas dėmesingas generatorius ir objekto lygmens diskriminatorius yra sukurti taip, kad įgalintų vaizdo generavimą, atsižvelgiant į pirmame žingsnyje sugeneruotą semantinį išdėstymą [LZZ+19]. Objektais pagrįsto dėmesingo generatyvinio besivaržančio tinklo architektūra yra pateikiama 6 paveiksle.



6 pav. Obj-GAN architektūros modelis [LZZ+19]

Kaip galima matyti 6 paveiksle pirmame žingsnyje Obj-GAN gauna sakinį kaip įvestį ir generuoja semantinį išdėstymą. Dėžės generatorius (*angl. box generator*) pirmiausia sukuria besiribojančių langelių seką, o tada formų generatorius sukuria jų formas [LZZ+19] .

Dėžutės generatorius. Dėmesingas seq2seq modelis yra treniruojamas, kaip dėžutės generatorius. 5 lygtyje pateikiama matematinė išraiška.

$$B_{1:T} := [B_1, B_2, \ldots, B_T] \sim G_{box}(e) \quad (5)$$

Šioje lygtyje yra iš anksto paruošti bi-LSTM žodžių vektoriai, $B_t = (l_t, b_t)$ yra t objekto klasės etiketė ir jį ribojantis langelis, kurio matematinė išraiška pateikta 6 lygtyje [LZZ+19].

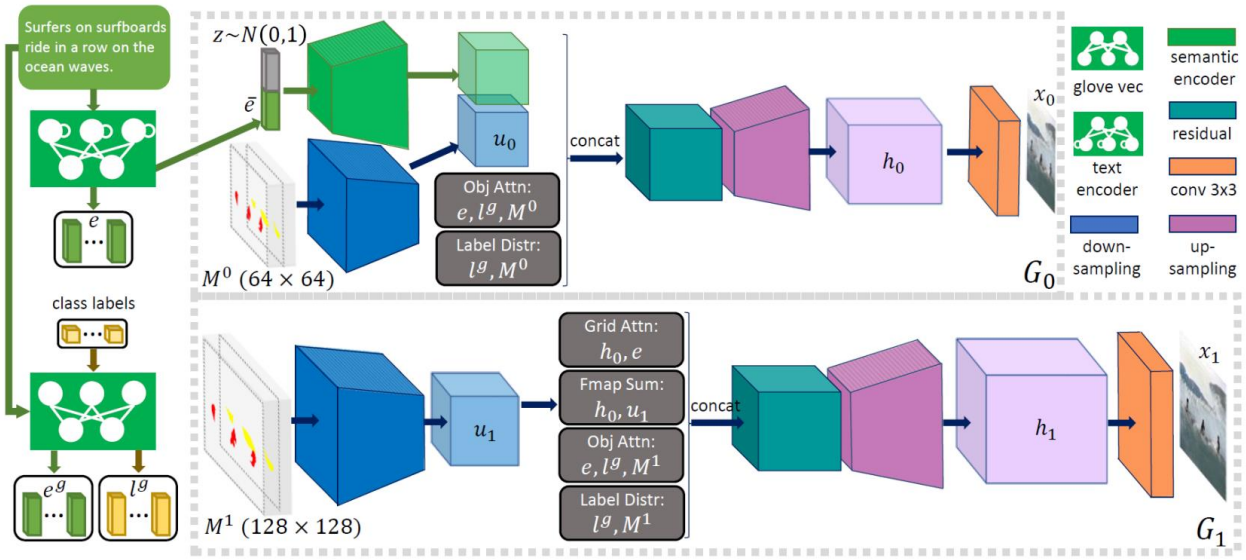
$$b = (x, y, w, h) \in R^4 \quad (6)$$

Formos generatorius *(angl. shape generator*). Atsižvelgiant į ribojančius langelius $B_{1:T}$, formos generatorius numato kiekvieno objekto formą, esantį jo ribose remiantis atsitiktiniu triukšmo vektoriumi (7 lygtis)

$$\widehat{M_{1:T}} = G_{shape}(B_{1:T}, z_{1:T}), \text{kur } z_t \sim N(0,1) \quad (7)$$

Kadangi sugeneruotos formos turi ne tik atitikti $B_{1:T}$ pateiktą informaciją apie vietą ir kategoriją, bet ir turi būti suderintos su aplinkiniu kontekstu. G_{shape} yra sukurtas remiantis dvikrypčiu konvoliuciniu LSTM, kaip parodyta 6pav. G_{shape} treniravimas yra grindžiamas GAN sistema, kurioje taip pat naudojami suvokimo nuostoliai, siekiant apriboti generuojamas formas ir stabilizuoti mokymą [LZZ+19] .

Nuotraukos generatorius. Nuotraukos generatoriaus architektūra pateikiama 7 paveiksle.



7 pav. Objektais grįsto dėmesingo nuotraukų generatoriaus architektūra [LZZ+19]

Kaip parodyta 7 pav., pasiūlytas dėmesingas daugiapakopis generatyvinis tinklas turi du generatorius (G_0, G_1). Bazinis generatorius G_0 pirmiausia sukuria mažos skiriamosios gebos vaizdą $\widehat{x_0}$, sąlygotą globalaus sakinio vektoriaus ir iš anksto sugeneruoto semantinio išdėstymo. Tada tikslintojas (*angl. refiner*) G_1 patikslina detales skirtinguose regionuose, atkreipdamas dėmesį į svarbiausius žodžius ir iš anksto sugeneruotas klasių etiketes ir sukuria didesnės raiškos vaizdą $\widehat{x_1}$ (8 ir 9 lygtys):

$$h_0 = F_0\left(z, \underline{e}, Enc(M^0), c^{obj}, c^{lab}\right), \widehat{x_0} = G_0(h_0) \quad (8)$$

$$h_1 = F_1\left(c^{pat}, h_0, Enc(M^1), c^{obj}, c^{lab}\right), \widehat{x_1} = G_1(h_1) \quad (9)$$

- z yra atsitiktinis vektorius su standartiniu normaliuoju skirstiniu;
- ($Enc(M^0)$ ($Enc(M^1)$) yra mažos skiriamosios gebos formų M^0 kodavimas (didesnės skiriamosios gebos formos M^1);
- $c^{pat} = F_{attn}^{grid}(e, h_0)$ yra tradicinio tinklelio dėmesio konteksto vektoriai pagal dėmenis;
- $c^{obj} = F_{attn}^{obj}(e, e^g, l^g, M)$ yra objekto konteksto vektoriai atsirandantys iš naujo objektais grįsto dėmesio, o $c^{lab} = c^{lab}(l^g, M)$ yra etikečių konteksto vektoriai iš klasių etikečių.

Į generavimo procesą galima įtraukti ir daugiau tikslinimo priemonių, tam, kad gauti didesnės raiškos vaizdus. Tačiau būtent Obj-GAN aprašytame straipsnyje yra naudoja du tikslintuvus (G_1, G_2). Ir galiausiai yra sugeneruojamas 256 × 256 raiškos vaizdas [LZZ+19].

2.4.2. Obj-GAN rezultatų vertinimas

Tokiuose straipsniuose, kaip [LZZ+19] ir [QCD+21] yra vertinami Obj-GAN gaunami rezultatai. Abiejuose straipsniuose yra minima, kad **kiekybiniai vertinimai** atliekami naudojantis FID ir R tikslumu, tačiau pirmame [LZZ+19] straipsnyje papildomai naudojamas IS, ko antrame [QCD+21] straipsnyje tiesiog atsisakoma dėl, kaip minima straipsnyje – netikslingai atvaizduojamų rezultatų.

Kokybiniai vertinimai abiejuose straipsniuose yra atliekami sulyginus kelių GAN sugeneruotas nuotraukas vizualiai. Papildomai antrame [QCD+21] straipsnyje daugiau dėmesio atkreipiama į sugeneruotoje nuotraukoje esančias formas, jų kontūrus bei nuotraukos natūralumą.

2.5. VQGAN-CLIP

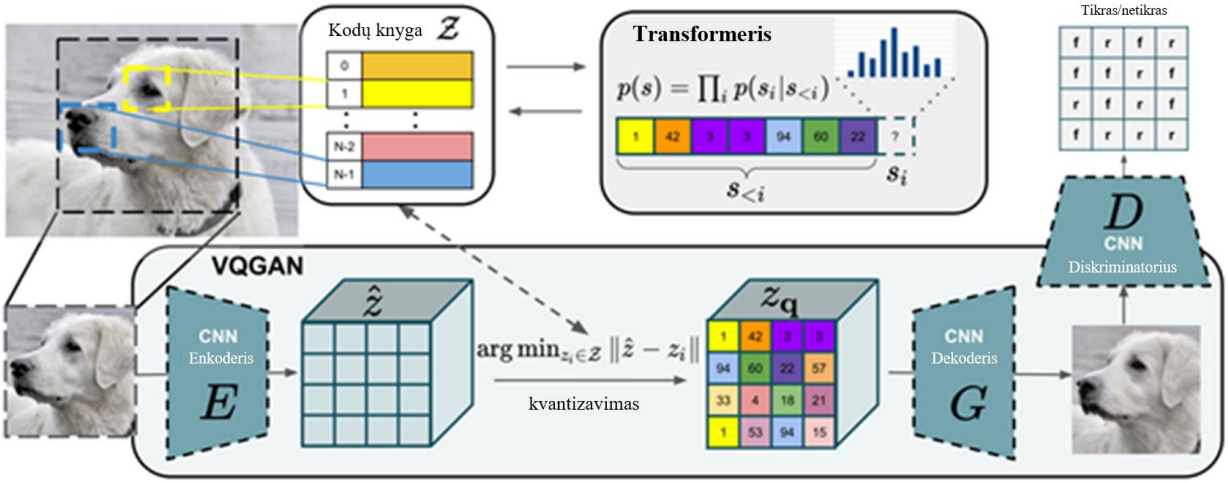
VQGAN-CLIP modelis pirmą kartą pristatytas dar 2021 pradžioje, tačiau rašytiniai šaltiniai apie šį modelį pasirodė po metų. VQGAN-CLIP gali būti išverstas, kaip vektorinis kvantizuotas generatyvinis besivaržantis tinklas (*angl. VQGAN - Vector Quantized Generative Adversarial Network*) sujungtas su kontrastiniu vaizdų ir kalbos parengiamuoju mokymu (angl. CLIP – *Contrastive Image-Language Pretraining*). VQGAN-CLIP modelis praktiškai yra sudarytas iš dviejų atskirų modelių, t.y. VQGAN, kuris yra generatyvinis besivaržantis tinklas generuojantis nuotraukas ir CLIP, kuris yra kitas neuroninis tinklas, tikrinantis ar sugeneruota nuotrauka sutampa su tekstu. Todėl grįstas tokiu požiūriu modelis yra kitoks nei prieš tai aptarti, nes yra sudarytas iš dviejų atskirų modelių, kurie atitinka generatoriaus ir diskriminatoriaus roles.

Straipsnyje [CBK+22] VQGAN-CLIP yra pristatomas, kaip pirmasis iš anksto paruoštas vaizdo generavimo modelis, kuris yra paremtas vieningu požiūriu į semantinio vaizdo generavimą ir redagavimą. Tokia metodika veikia naudojant daugiamodalį kodavimo įrenginį, kuris įvertina

(teksto, vaizdo) poros panašumą ir grįžtamuoju srautu persikelia į vaizdo generatoriaus paslėptą erdvę. Vaizdas yra iteratyviai atnaujinamas, kol jis tampa pakankamai panašus į tikslinį tekstą. Skirtumas tarp šios technikos naudojimo generuojant ir redaguojant yra tik generatoriaus inicializavimas su tam tikru vaizdu (redaguoti) arba su atsitiktiniu triukšmu (generuoti).

2.4.1. VQGAN-CLIP architektūra

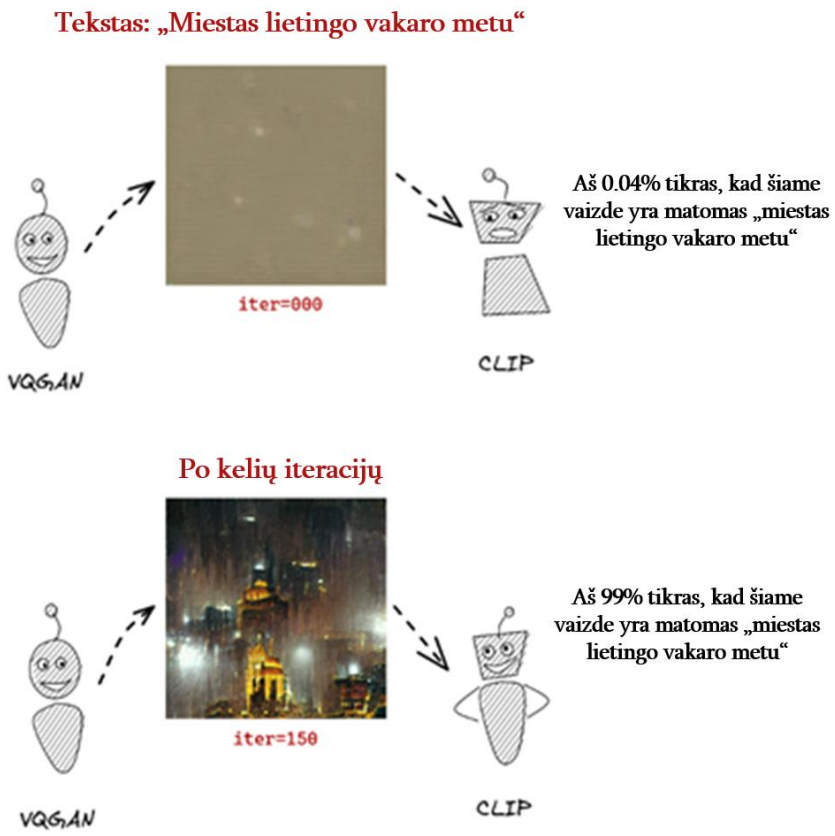
Šis modelis, kaip jau buvo minėta, naudoja du atskirus neuroninius tinklus – VQGAN ir CLIP, kaip iš anksto paruoštus modelius. Trumpai tariant CLIP nukreipia VQGAN link vaizdo, kuris geriausiai atitinka pateiktą tekstą [CBK+22]. Kadangi šis modelis yra sudarytas iš dviejų atskirų modelių, šiame darbe bus aprašoma tik VQGAN architektūra ir nurodoma, kaip yra naudojamas CLIP funkcionalumas. VQGAN modelio architektūra yra pateikiama 8 paveiksle.



8 pav. VQGAN modelio architektūra [CBK+22]

VQGAN modelis yra naudojamas tam, kad išmoktų konteksto turtingų vaizdinių dalių kodų knygą, kurios sudėtis vėliau modeliuojama naudojant autoregresinio transformatoriaus architektūrą. Diskrečioji kodų knyga yra šių architektūrų sąsaja, o lopais pagrįstas (*angl. patch-based*) diskriminatorius leidžia stipriai suspausti (*angl.compress*) vaizdą išlaikant aukštą suvokimo kokybę. Šis metodas užtikrina konvoliucinių metodų veiksmingumą transformatoriais pagrįstame didelės skiriamosios gebos vaizdų generavime [ERO21].

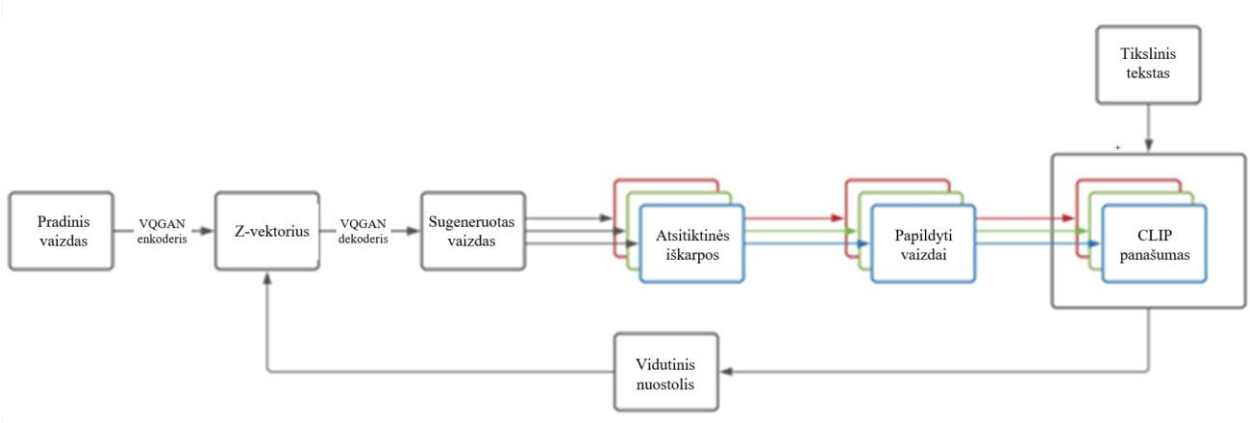
O CLIP šiame modelyje atlieka diskriminatoriaus vaidmenį, kaip yra parodyta 9 paveiksle.



9 pav. CLIP modelio atliekamos diskriminavimo funkcijos vizualizacija [Mir21]

9 pav. galima matyti, kaip veikia VQGAN kartu su CLIP. VQGAN sugeneravus nuotrauką, CLIP eilė veikti – šis modelis nusako, kaip tiksliai pateiktas aprašymas atitinka nuotrauką ir šis ciklas kartojasi tol, kol CLIP pateiktas rezultatas yra atitinkantis poreikius.

10 paveiksle vaizduojama, kaip šie du modeliai veikia kartu, siekiant stabilizuoti ir pagerinti optimizavimą.



10 pav. VQGAN-CLIP veikimas[CBK+22]

Apibendrinant visą procesą nuotraukos generavimas iš teksto naudojant VQGAN-CLIP vyksta taip: norint sukurti vaizdą, pradiniame vaizde yra atsitiktinių pikselių reikšmės. Optimizavimo procesas kartojamas, kad vaizdas būtų pakeistas, kol išvesties vaizdas palaipsniui pagerėja, kad semantiškai atitiktų tikslinį tekstą. Taip pat šiame modelyje yra galimybė redaguoti

esamus vaizdus, pradėdami redaguoti vaizdą kaip pradinį vaizdą. Teksto įvesties eilutė naudojama apibūdinti, kaip yra norima, kad vaizdas būtų pakeistas tokiu pačiu principu, kaip ir generuojant vaizdą iš atsitiktinių pikselių reikšmių [CBK+22].

2.4.2. VQGAN-CLIP rezultatų vertinimas

Originaliame VQGAN-CLIP [CBK+22] **kiekybinis modelio įvertinimas** vyko atsižvelgiant į tai, kiek laiko buvo užtrunkama sugeneruoti nuotrauką bei kiek laiko truko modelio treniravimas, taip pat visų modelių vertinimams buvo pasitelkiami tie patys GPU, kad būtų galima pamatyti tikslų skirtumą tarp skirtingų modelių. O tokio, kaip **kokybinio vertinimo** nebuvo atliekama apskritai, prieduose papildomai buvo pateikiama, kaip kiekvienas modelis gali sugeneruoti vaizdus remiantis parašytu tekstu paliekant skaitytojams patiems nuspręsti, kuris modelis yra tiksliausiai atitinkantis aprašymą.

VQGAN-CLIP buvo vienas iš lyginamųjų modelių ir [WLH+22] straipsnyje. Šiame straipsnyje **kiekybiškai vertinama** buvo remiantis IS, FID bei papildomai naudojantis CapS skaičiavimu. IS ir FID yra skaičiuojami lygiai taip pat, kaip buvo aprašyta anksčiau, o CapS matuoja semantinius panašumus tarp įvesties teksto ir sugeneruoto vaizdo. O **kokybiniu vertinimu** buvo pasirinkta vertinti vizualiai atsižvelgiant į sugeneruotų vaizdų detalumą.

2.6. Naujausių apžvelgtų GAN vaizdams iš teksto generuoti apibendrinimas

Antrame skyriuje buvo apžvelgti 5 generatyviniai besivaržantys neuroninių tinklų modeliai skirti vaizdams iš teksto generuoti, kurie skirtingai sprendė vaizdo generavimo iš teksto problemą vis bandant pagerinti vieną ar kitą vietą, kuri leistų vaizdus sugeneruoti žymiai tiksliau ir ryškiau atvaizduojant su vis daugiau detalių ir kuo labiau atitinkantį realius vaizdus. Tad norint įvertinti aptartus modelius ir išsirinkti kelis perspektyviausius, kurie bus toliau eksperimentiškai analizuojami 3 skyriuje yra pateikiama 1 lentelė.

1 lentelė. GAN modelių palyginimas

Palyginimo atributai	**GAN modeliai**				
	DF-GAN	**SSA-GAN**	**DM-GAN**	**Obj-GAN**	**VQGAN-CLIP**
Architektūra	Vieno lygmens (turi vieną generatorių ir vieną diskriminatorių)	Vieno lygmens (turi vieną generatorių ir vieną diskriminatorių)	Daugiapakopis (turi generavimo procesą ir pagerinimo procesą, kurie abu turi po generatorių ir diskriminatorių)	Daugiapakopis (sudarytas iš dviejų žingsnių: išdėstymo sudarymo ir nuotraukos generavimo, nuotraukos generavime yra du	Sudarytas iš dviejų atskirų modelių, VQGAN atlieka generatoriaus rolę, o CLIP atlieka diskriminatoriaus vaidmenį

				generatoriai ir vienas diskriminatorius)	
Sugeneruotų nuotraukų rezoliucija	256 × 256	256 × 256	256 × 256	256 × 256	Iki 8k rezoliucijos
Kiekybinio vertinimo matas	FID, IS, NoP	FID, IS	FID, R tikslumas, IS	FID, R tikslumas, IS	Laikas praleistas generuojant nuotrauką, IS, FID CapS
Kokybės vertinimo matai	Vizualiai patikrinamos nuotraukos ir aprašoma, kas yra matoma netikslinga; lyginant, kiek detalių yra panašių į tikrus vaizdus	Vizualiai vertinant fono bei detalių ryškumą, detalių atitikimą pagal tekstą	Vizualiai, atkreipiant dėmesį į vaizdų detalių ryškumą, fono aiškumą	Vertinama vizualiai, atkreipiant dėmesį į formas, jų kontūrus bei nuotraukos natūralumą.	Vizualiai pagal sugeneruotų vaizdų detalumą
Naujumas	2020	2021	2019	2019	2021

Kaip galima matyti 1 lentelėje yra pateikiami skirtumai ir panašumai visų aptartų GAN modelių, skirtų vaizdams iš teksto generuoti. Ir sulyginus juos visus reikia išsirinkti modelius gilesniam eksperimentiniam analizavimui. Kadangi yra poreikis analizuoti naujausius modelius, dėl to 3 skyriuje bus analizuojami tokie GAN skirti vaizdų generavimui iš teksto, kaip **SSA-GAN, VQGAN-CLIP bei DM-GAN**. SSA-GAN ir VQGAN-CLIP yra pasirenkami dėl to, kad jie yra naujausi ir abu turi skirtingą architektūrą. Ir trečiasis modelis vaizdų generavimui iš teksto buvo pasirinktas DM-GAN, nes savo architektūra yra mažiau sudėtingas lyginant su Obj-GAN nors ir yra laikomas daugiapakopis, kaip ir Obj-GAN.

3. Generatyvinių besivaržančių neuroninių tinklų eksperimentinė analizė

Šiame skyriuje bus aptariama trijų generatyvinių besivaržančių neuroninių tinklų vaizdams iš teksto generuoti – Semantic-Spatial Aware GAN, VQGAN-CLIP ir DM-GAN eksperimentinė analizė. Kiekvienas iš paminėtų generatyvinių besivaržančių neuroninių tinklų vaizdams iš teksto generuoti yra skirtingo architektūrinio sudėtingumo bei turintys skirtingą sprendimo būdą, kaip geriau/tiksliau sugeneruoti vaizdą pagal pateiktą aprašymą. Tad eksperimentiškai išanalizavus pasirinktus modelius bus galima matyti, kokie rezultatai yra gaunami šiuo specifiniu atveju. Į analizę įeis eksperimento eigos aprašymas, sugeneruotų nuotraukų pavyzdžiai bei jų vertinimai pagal nustatytus rodiklius. Pagrindiniai duomenys, kurie bus reikalingi eksperimentiniai analizei:

Duomenų rinkiniai: Pirmasis duomenų rinkinys, kuris bus naudojamas atliekant eksperimentą yra CUB paukščių duomenų rinkinys, jame išvis yra 200 paukščių kategorijų su 11788 vaizdų. Iš jų 8855 nuotraukų yra skirta apmokymui ir 2933 nuotraukų testavimui. Tarp nuotraukų, skirtų modelio treniravimui, yra 150 paukščių rūšių, o tarp testavimo nuotraukų yra 50 paukščių rūšių. Kiekvienas paukštis turi 10 tekstinių aprašymų. Antrasis duomenų rinkinys yra COCO su 80000 nuotraukų apmokymui ir 40000 nuotraukų testavimui. Kiekviena nuotrauka turi po 5 tekstinius aprašymus. COCO duomenų rinkinyje esantys vaizdai yra sudėtingesni palyginus su CUB duomenų rinkiniu, nes CUB duomenų rinkinyje yra atvaizduojami tik paukščiai, o COCO duomenų rinkinyje yra įvairiausių vaizdų – pradedant nuo daiktų nuotraukų iki žmonių, atliekančių įvairiausias veiklas, todėl naudojant COCO duomenų rinkinį vaizdų generavimas iš teksto tampa žymiai sudėtingesnė užduotis, dėl ko rezultatai gali būti žymiai prastesni.

Vertinimo rodikliai: kaip ir didžiojoje dalyje aptartų straipsnių, bus naudojami tokie patys kiekybiniai rodikliai, t. y. **IS** (*angl. Inception Score*) ir **FID** (*angl. Frechet Inception Distance*), kokybinis vertinimas bus atliekamas sulyginant nuotraukas vizualiai.

IS arba kitaip pradinis balas yra metrika, skirta automatiškai įvertinti vaizdų generavimo modelio kokybę [SGZ+16]. Ši metrika naudoja apmokytą *Inception v3* tinklą, kuris yra gili konvoliucinė architektūra skirta klasifikavimo užduotims atlikti. *Inception v3* tinklas yra apmokytas naudojant *ImageNet* duomenų rinkinį, kurį sudaro 1,2 milijono vaizdų, kurie gali būti suskirstyti į 1000 klasių. Tad IS naudojant aprašytą *Inception v3* tinklą apskaičiuoja tinklo išeigų statistiką, kai ji yra taikoma sugeneruotiems vaizdams [SR18]. Lygtis, kuri yra naudojama apskaičiuoti pradinį balą:

$$IS(G) = exp\left(E_{x\sim p}\, D_{KL}(p(x) \,||\, p(y))\right) \tag{10}$$

10 lygtyje $x \sim p_g$ nurodo, kad x yra vaizdas paimtas iš p_g , $D_{KL}(p||q)$ yra KL-divergencija tarp p ir q pasiskirstymo. $p(y|x)$ yra sąlyginis klasės skirstinys ir $p(y) = \int_x \quad p(y|x)p_g(x)$ yra ribinis klasių pasiskirstymas. Lygtyje ($IS(G)$) esantis $E_{x\sim p}$ yra skirtas tam, kad būtų galima

lengviau palyginti reikšmes, todėl jis yra ignoruojamas ir naudojamas, kaip $ln\left(IS(G)\right)$ neprarandant bendrumo [SR18].

Autoriai, kurie pasiūlė IS, kaip metriką siekė šiuo apskaičiavimu atvaizduoti dvi pageidaujamas generatyvinio modelio savybes:

- Sukurtuose vaizduose objektai turi būti aiškiai atvaizduojami (t. y. vaizdai yra ryškūs, o ne išplaukę) arba $p(y|x)$ turėtų būti mažos entropijos. Kitaip tariant, įvesties tinklas turėtų būti įsitikinęs, kad vaizde yra vienas objektas.
- Generatyvinis algoritmas turėtų pateikti didelę vaizdų įvairovę iš visų skirtingų *ImageNet* klasių arba $p(y)$ turėtų būti didelės entropijos.

Jei sugeneruotas vaizdas tenkina abu šiuos bruožus tikėtina, kad KL divergencija tarp skirstinių $p(y)$ ir $p(y|x)$ bus didelė, todėl gaunamas aukštas IS rezultatas [SR18].

Frechet pradžios atstumo balas (*angl. Frechet Inception Distance*) arba trumpiau FID yra generuojamų vaizdų kokybės įvertinimo metrika, specialiai sukurta siekiant įvertinti generatyvinių besivaržančių tinklų veikimą. FID rodiklį pasiūlė ir naudojo Martin Heusel ir kt. savo straipsnyje [MHT+17]. Balas buvo pasiūlytas kaip esamo pradinio balo (IS) patobulinimas. Pradinis balas įvertina sugeneruotų vaizdų kolekcijos kokybę pagal tai, kaip efektyviausias vaizdų klasifikavimo modelis *Inception v3* klasifikuoja juos kaip vieną iš 1000 žinomų objektų. Balai sujungia tiek sąlyginių klasių prognozių patikimumą kiekvienam sugeneruotam vaizdui (kokybei), tiek numatomų klasių ribinės tikimybės integralą (įvairovę). Pradinis balas neatspindi sugeneruotų vaizdų palyginimo su tikrais vaizdais. Kuriant FID balą, tikslas buvo įvertinti sugeneruotus vaizdus remiantis generuotų vaizdų kolekcijos statistika, palyginti su realių vaizdų kolekcijos statistika iš tikslinės srities. Kaip ir pradinis balas, FID balas naudoja *Inception v3* modelį. Konkrečiai, modelio kodavimo sluoksnis (paskutinis sutelkimo sluoksnis prieš išvesties vaizdų klasifikaciją) naudojamas kompiuterinei regai būdingoms įvesties vaizdo ypatybėms užfiksuoti. Šios aktyvacijos apskaičiuojamos realių ir sugeneruotų vaizdų kolekcijai. Aktyvacijos yra apibendrinamos, kaip daugiamatis Gausas, kuris apskaičiuoja vaizdų vidurkį ir kovariaciją. Ši statistika apskaičiuojama realių ir sugeneruotų vaizdų kolekcijoje. Tada atstumas tarp šių dviejų skirstinių apskaičiuojamas naudojant Frechet atstumą, dar vadinamą Wasserstein-2 atstumu. Balas, naudojamas apibendrinti kiekvieno vaizdo aktyvaciją naudojant *Inception v3* modelį vadinamas *Frechet Inception Distance*. Mažesnis FID rodo geresnės kokybės vaizdus; ir atvirkščiai, didesnis balas rodo prastesnės kokybės vaizdą. Rezultatų autoriai [MHT+17] rodo, kad mažesni FID balai koreliuoja su geresnės kokybės vaizdais, kai buvo taikomi sistemingi iškraipymai, pavyzdžiui, pridėtas atsitiktinis triukšmas ir neryškumas [YZD21].

Kitos svarbios detalės: Visi GAN bus bandomi naudojant NVIDIA GTX 1060 GPU, su tikslu gauti rezultatus paremtus vienodomis sąlygomis.

3.1. Semantic-Spatial Aware GAN eksperimentinė analizė

Pirmasis GAN, kuris yra bandomas eksperimentiškai tai yra Semantic-Spatial Aware GAN. Šis modelis yra sukurtas naudojantis *PyTorch*. Semantic-Spatial Aware GAN programinis kodas buvo pateikiamas kartu su straipsniu [HLY+22] ir jį buvo galima pasiekti *Github* aplinkoje, kaip viešai prieinamą projektą. Parsisiuntus minėtojo GAN programinį kodą pirmasis darbas buvo pabandyti pasileisti, patikrinti ar viskas veikia, ar netrūksta jokių bibliotekų ar kitų resursų. Deja, bet kaip ir buvo tikėtasi, iš pirmo karto niekas nepradėjo veikti, teko parsisiųsti visas trūkstamas bibliotekas ir kode pakeisti tas vietas, kurios jau buvo pasenusios ir nebenaudojamos pačių bibliotekų ar neturėjo naudojamų metodų Windows operacinės sistemos aplinkoje. Kadangi vienintelė prieinama dokumentacija apie šį modelį buvo originaliame straipsnyje ir *Github* paskyroje, kuriame buvo pateiktas kodas, teko klaidas aiškintis jas spausdinant į konsolę ir tikrinant, kas vienoje ar kitoje vietoje galėtų būti kitaip ar kokia informacija į specifinę vietą turėtų patekti. Po kiek laiko ir daugybės bandymų pavyko pasileisti ir pradėti bandyti patį modelį, jo galimybes ant turimo GPU ir kokie rezultatai yra sugeneruojami.

Papildomi duomenys, kurie buvo nustatyti norint gauti geriausius rezultatus: duomenų rinkinio dydis (*angl. batch size*) buvo nustatytas į 4, nes daugiau tiesiog fiziškai neužteko atminties turimam GPU. Modelio treniravimui naudojamas Adam optimizatorius, kurio $\beta1 = 0{,}0$ ir $\beta2 = 0{,}9$ ($\beta1$ ir $\beta2$ yra eksponentinis skilimo greitis momentiniams įverčiams, privalomas nurodyti naudojant Adam optimizatorių [KB14]). Generatoriaus ir diskriminatoriaus treniravimo rodikliai nustatyti atitinkamai 0,0001 ir 0,0004. Modelio treniravimas naudojant CUB duomenų rinkinį vyko 600 epochų, o naudojant COCO duomenų rinkinį vyko 120 epochų. Galutinius kiekybinius rodiklius galima matyti 2 lentelėje.

2 lentelė. Sugeneruotų vaizdų įvertinimai naudojant Semantic-Spatial Aware GAN

Duomenų rinkiniai	**IS ↑**	**FID ↓**
CUB	5,13 ± 0,19	18,41
COCO	20,52 ± 0,98	25,19

Kaip anksčiau buvo minėta – didesnis IS reiškia aukštesnę kokybę bei teksto ir vaizdo semantinį nuoseklumą, o FID nurodo sugeneruotos nuotraukos kokybę. Atsižvelgiant į 2 lentelę galima matyti ir vertinti gautus kiekybinius rezultatus pagal aprašytas metrikas ir susidaryti įspūdį, kad naudojant CUB duomenų rinkinį buvo generuojami tikslesni vaizdai nei pagal COCO. Bet COCO duomenų rinkinys turėjo žymiai platesnę nuotraukų įvairovę, dėl ko tikslesnes nuotraukas pagal aprašymą buvo žymiai sunkiau sugeneruoti dėl didesnio detalių ir tekstūrų kiekio. Tačiau

vertinti vien iš skaičiavimų yra sudėtinga nežinant, kaip atrodo sugeneruotos nuotraukos, tad pridedami 11 ir 12 paveikslai, kuriuose galima matyti vizualiai, kokius rezultatus sugeneravo Semantic-Spatial Aware GAN modelis naudojantis CUB ir COCO duomenų rinkiniais. Kokius rezultatus sugeneravo Semantic-Spatial Aware GAN modelis naudojantis CUB duomenų rinkinį galima matyti 11 paveiksle.

Spalvingas geltonas paukštis turi sparnus su tamsiomis juostelėmis ir mažomis akimis
A colorful yellow bird has wings with dark stripes and small eyes

Spalvingas rudas paukštis turi sparnus su tamsiomis juostelėmis ir mažomis akimis
A colorful brown bird has wings with dark stripes and small eyes

Spalvingas baltas paukštis turi sparnus su tamsiomis juostelėmis ir mažomis akimis
A colorful white bird has wings with dark stripes and small eyes

Spalvingas mėlynas paukštis turi sparnus su tamsiomis juostelėmis ir mažomis akimis
A colorful blue bird has wings with dark stripes and small eyes

11 pav. SSA-GAN sugeneruoti vaizdai naudojant CUB duomenų rinkinį

Kaip galima matyti 11 pav. pateikti paukščių vaizdai yra sugeneruojami gana tikroviškai ir žiūrint į juos galima sakyti, kad yra matomas paukštis daugiau ar mažiau panašus į aprašą, nors ir ne visuose sugeneruotose nuotraukose matomas pilnas paukštis. Taip pat verta atkreipti dėmesį, kad sugeneruotame vaizde galima matyti ne tik paukštį, bet ir sugeneruotą foną ar jo fragmentus, kuris bando atspindėti paukščio buvimo vietą. Pirmame stulpelyje galima teigti, jog paukštis tupi ant šakos medyje, o ketvirtame stulpelyje matomos šakos ar žolės fragmentai, kurie atrodo gana realistiškai.

Patikrinus, kaip Semantic-Spatial Aware GAN generuoja vaizdus remiantis paukščių duomenų rinkiniu reikia patikrinti ir galimybes, generuoti vaizdus remiantis COCO duomenų rinkiniu. 12 pav. pateikiami atsitiktiniu būdu išrinkti vaizdai, kurie buvo sugeneruoti naudojant COCO duomenų rinkinį.

Merginos, valgančios didelę picos gabalėlį, vaizdas
An image of a girl eating a large slice of pizza

Laikrodis, kuris yra bokšto šone
A clock that is on the side of a tower

Ant žolės besiganančių karvių banda
A herd of cows that are grazing on the grass

Vaikinas, kažką taisantis ant savo motociklo
A guy fixing something on his motor cycle

12 pav. SSA-GAN sugeneruoti vaizdai su COCO duomenų rinkiniu

Lyginant 11 pav. ir 12 pav. pateiktus vaizdus akivaizdžiai galima įžvelgti skirtumus ne tik dėl to, kad 11 pav. yra atvaizduojami paukščiai, o 12 pav. įvairesni objektai, bet ir tai, kad paukščiai yra sugeneruoti žymiai geriau ir suprantamiau nei įvairūs objektai iš COCO duomenų rinkinio. Bet vertinant 12 pav. pateikiamus vaizdus galima matyti, kad fonas taip pat yra ryškiai ir gana aiškiai sugeneruotas, atsižvelgiant į tekstą taip pat nereikia bandyti labai stipriai įsivaizduoti kur kas turėtų būti, kaip ir yra aišku, tačiau žiūrint į šį paveikslą galima sakyti, kad šie vaizdai nėra tikroviški ir iki tikroviškumo trūksta ne vienos iteracijos ir patobulinimo. Pagal 12 pav. galima matyti, kad Semantis-Spatial Aware GAN yra labai sunku sugeneruoti žmogų, gyvūnas kaip ir yra sugeneruojamas, bet jam trūksta detalių, o bokštas su laikrodžiu, kaip ir būtų geriausias pavyzdys, tačiau jam trūksta tikslumo taip pat. Bet visa tai yra dėl to, kad COCO duomenų rinkinys nėra sutelktas į vieną kategoriją, dėl to sugeneruotos nuotraukos nėra tokios aiškios lyginant su paukščiais.

3.2.VQGAN-CLIP eksperimentinė analizė

VQGAN-CLIP buvo antrasis besivaržančių neuroninių tinklų vaizdams iš teksto generuoti modelis, kurį reikėjo išanalizuoti eksperimentiniu būdu. Šis modelis taip pat buvo parašytas naudojantis *Pytorch* ir prieinamas viešai – kodas buvo pateikiamas internete, projektas buvo prieinamas naudojantis *Google Colab*. Deja, bet *Google Colab* versijos pasileisti niekaip nepavyko, nes bandant sukompiliuoti kodą vis atsirasdavo problemų susijusių su naudojamomis bibliotekomis. *Google Colab* versija buvo bandoma pirmiausiai, norint išbandyti modelį internetinėje erdvėje, tačiau nepavykus to padaryti buvo pereita prie programinio kodo analizavimo. Šį modelį buvo sunkiausia prisitaikyti savo turimoje aplinkoje, lyginant su kitais modeliais, kurie yra apžvelgiami šiame darbe. Šis GAN turėjo didžiausią kiekį instrukcijų, ką pakeisti ir kaip tinkamai parengti savo turimą aplinką, kad modelis veiktų tinkamai. Tačiau sekant rastus nurodymus kilo problemų, kurių sprendimo rasti ne visada pavykdavo, tad teko ieškoti alternatyvių šaltinių, kurie suteiktų informacijos, kaip tinkamai pasileisti VQGAN-CLIP *Windows* aplinkoje. Galiausiai buvo rastas vaizdo įrašas, kuris žingsnis po žingsnio parodė, kaip tinkamai pasileisti šį GAN ir iš kur imti papildomus duomenis norint priversti šį modelį veikti. Tai padėjo priartėti prie galutinio tikslo – pabandyti sugeneruoti nuotrauką naudojant VQGAN-CLIP. Tačiau tai dar buvo ne pabaiga, nes, kaip ir anksčiau aprašytame modelyje naudojant šį GAN taip pat teko susitvarkyti pasenusių bibliotekų problemas. Susitvarkius ir pritaikius savo aplinkai šį modelį buvo galima pradėti testuoti ir žiūrėti, kokie rezultatai yra gaunami.

Bandant generuoti nuotraukas pastebėta, kad turimi resursai yra žymiai mažesni nei, kad yra rekomenduojama naudojantis šiuo modeliu, turimi resursai yra 3GB vaizdo plokštės atminties, o rekomenduojama turėti mažiausiai 8GB. Visa laimė, tai nesustabdė pabandyti sugeneruoti

nuotraukų, tik šiek tiek apribojo, vietoje aukštos rezoliucijos 8k vaizdų buvo galima sugeneruoti mažus 200x200 pikselių paveikslėlius. Testuojant VQGAN-CLIP buvo pastebėta, kad nėra arba nebuvo rasta labai daug parametrų, kurios būtų galima keisti, pačiame modelyje, kaip Semantic-Spatial Aware GAN atveju. Bei generuojant nuotraukas su šiuo modeliu viskas buvo pagrįsta ne COCO ir CUB duomenų rinkiniais, nes VQGAN-CLIP visiškai kitaip prideda duomenų rinkinius ir jais remiantis treniruojasi. O pritaikyti COCO ir CUB duomenų rinkinius ir juos naudojant apmokyti VQGAN-CLIP nepavyko. Dėl šios priežasties nėra tikslinga matuoti šio GAN IS ar FID, nes sulyginti rezultatai nebus paremti tais pačiais duomenimis.

Nors negalima sulyginti šio modelio generuojamų vaizdų naudojant kiekybines metrikas, tačiau galima vizualiai įvertinti VQGAN-CLIP sugeneruotų vaizdų kokybę. VQGAN-CLIP sugeneruotų vaizdų pavyzdžiai pateikiami 13 paveiksle.

Miesto panorama per vandenį saulėlydžio metu

City skyline across the water at sunset

Violetinis paukštis turi sparnus su tamsiomis juostelėmis ir mažomis akimis

A purple bird has wings with dark stripes and small eyes

Spalvingas purpurinis paukštis turi sparnus su tamsiomis juostelėmis ir mažomis akimis

A colourful purple bird has wings with dark stripes and small eyes

Paveikslas su obuoliu vaisių dubenyje

A painting of an apple in a fruit bowl

13 pav. VQGAN-CLIP sugeneruoti vaizdai

Kaip galima matyti 13 pav. šis modelis gali sugeneruoti labai detalias nuotraukas, tačiau aprašymas turi būti labai paprastas arba nuotrauka turi pereiti daug generavimo ciklų duodant modeliui, kaip pirminę nuotrauką tą paveikslą, kuris buvo gaunamas anksčiau, kad būtų galima jį pagerinti, nes kitu atveju sugeneruota nuotrauka atrodys ne visai taip, kaip tikimasi. Taip pat verta paminėti, kad VQGAN-CLIP, kaip galima matyti generuoja ryškiausius ir aukščiausios kokybės vaizdus. VQGAN-CLIP generatyvinis besivaržantis neuroninis tinklas vaizdams iš teksto generuoti dažniausiai yra naudojamas menui kurti, dėl to naudojantis šiuo modeliu yra galimybė pasirinkti stilių, kuriuo remiantis galima generuoti nuotrauką. Kaip galima matyti 13 paveiksle pirmame, antrame ir trečiame stulpelyje nuotraukos yra generuotos realistišku stiliumi, o ketvirtame stulpelyje nuotrauka atrodo, kaip piešta, nes aprašyme yra žodis paveikslas. Nors patys objektai yra sugeneruojami, kaip turintys daug detalių, fonui tenka mažiau dėmesio, pavyzdys - pirmame stulpelyje esantis vanduo, kuris nėra pilnai panašus į vandenį, tačiau turintis vandeniui panašių savybių.

3.3.DM-GAN eksperimentinė analizė

Paskutinis modelis, kuris bus apžvelgiamas eksperimentiškai šiame darbe yra DM-GAN. Šis generatyvinis besivaržantis neuroninis tinklas, kaip ir anksčiau aptartieji yra sukurtas naudojant *Python* biblioteką *PyTorch*. Šio modelio originalus straipsnis [ZPC+19] taip pat turėjo nuorodą į programinį kodą, esantį *Github* aplinkoje, kuris buvo sukurtas aprašant šį GAN. Parsisiuntus jį, kaip ir su prieš tai buvusiais modeliais – pirmiausia reikėjo prisitaikyti prie turimos operacinės sistemos aplinkos ir įsitikinti, kad viskas veikia. Tai atlikti buvo lengviausiai lyginant su Semantic-Spatial Aware GAN ir VQGAN-CLIP. Nors pats modelis buvo prieinamas viešai ir jį buvo galima parsisiųsti, vykdant instrukcijas buvo pastebėta, kad deja, bet ne viskas buvo prieinama – IS apskaičiavimo formulė pritaikyta būtent šiam modeliui buvo viešai neprieinama. Todėl teko pritaikyti IS apskaičiavimo formulę iš SSA GAN modelio, kad būtų galima pateikti vieną iš kiekybinių metrikų, kurios ir trūko. Viską pasiruošus buvo galima pradėti tikrinti, kokius vaizdus šis modelis gali generuoti.

DM-GAN, kaip ir buvo numatyta, mokymui ir testavimui naudojo CUB ir COCO duomenų rinkinius. Bandant generuoti vaizdus naudojantis šiuo GAN, kad užtektų turimų resursų, teko pritaikyti tokias konfigūracijas: pagal numatytuosius nustatymus buvo paliekama $N_w = 256$, $N_r = 64$ ir $N_m = 128$, atitinkamai, kaip teksto, vaizdo ir atminties funkcijų vektorių matmenis, duomenų rinkinio dydis (*angl. batch size*) buvo nustatytas į 4, nes daugiau turimi resursai neleido. Adam optimizatorius mokymuisi buvo nustatytas į β1 = 0,5 ir β2 = 0,999 (β1 ir β2 yra eksponentinis skilimo greitis momentiniams įverčiams, privalomas nurodyti naudojant Adam optimizatorių [KB14]). Mokymosi rodiklis buvo nustatytas į 0,0002. DM-GAN modelio treniravimas naudojant CUB duomenų rinkinį vyko 600 epochų, o su COCO duomenų rinkiniu vyko 120 epochų. Galutinius kiekybinius rodiklius galima matyti 3 lentelėje.

3 lentelė. Sugeneruotų vaizdų įvertinimai naudojant DM-GAN

Duomenų rinkiniai	**IS ↑**	**FID ↓**
CUB	4,71±0,17	16,19
COCO	29,71±0,97	33,75

Kaip galima matyti 3 lentelėje naudojant CUB duomenų rinkinį rezultatai gaunami geresni nei su COCO duomenų rinkiniu. Panašius rezultatus galima pastebėti ir Semantic-Spatial Aware GAN eksperimentinėje analizėje. Vaizdai sugeneruoti iš COCO duomenų rinkinio, taip pat, kaip ir Semantic-Spatial Aware GAN eksperimentinėje analizėje parodo prastesnius rezultatus lyginant vizualiai ir pagal FID, nes kaip jau buvo minėta COCO duomenų rinkinys susideda iš įvairesnių kategorijų, kurios tuo pačiu atveria kitas galimybes generuoti įvairesnius vaizdus, tiek ir sudaro sunkumų, nes šio duomenų rinkinio neužtenka šiam specifiniui atvejui pilnai tinkamai apmokyti

GAN, kad gauti naują realistišką vaizdą. Grįžtant prie rezultatų – sulyginus Semantic-Spatial Aware GAN ir DM-GAN kiekybinius rezultatus galima pastebėti, kad šie skiriasi nežymiai.

O vizualūs rezultatai naudojant CUB duomenų rinkinį pateikti 14 nuotraukoje.

14 pav. DM-GAN sugeneruoti vaizdai naudojant CUB duomenų rinkinį

Kaip galima matyti 14 paveiksle, paukščiai sugeneruojami yra tikroviškai – galima suprasti, kad tai yra paukštis, skaitant aprašymą galima pastebėti aprašytas detales, nors ir ne visuose vaizduose aprašytos detalės pilnai atsispindi sugeneruotame vaizde. Taip pat įsižiūrėjus atidžiau galima matyti, kad paukščiai turi neaiškių/išplaukusių detalių iš kurių galima suprasti, kad tai yra sugeneruotas vaizdas, o ne nuotrauka. Be viso to reikia atsižvelgti ir į foną – galima matyti, jog šiuose paveiksluose fonas yra detalesnis ir ryškesnis nei Semantic-Spatial Aware GAN sugeneruotų nuotraukų, nors ne visos sugeneruotos detalės yra užbaigtos (pvz.: 14pav. 4 stulpelis), bet tai būtų galima pagerinti pritaikant daugiau tikslinimo ciklų, per kuriuos turėtų pagerėti bendra vaizdo kokybė.

Vizualiai išanalizavus sugeneruotų paukščių vaizdus reikia aptarti ir vaizdus, kurie buvo generuojami naudojant COCO duomenų rinkinį. 15 paveiksle pateikiami DM-GAN sugeneruoti vaizdai naudojant COCO duomenų rinkinį.

15 pav. DM-GAN sugeneruoti vaizdai naudojant COCO duomenų rinkinį

DM-GAN sugeneruoti vaizdai, naudojant COCO duomenų rinkinį, šiuo atveju nėra tokie realistiški, kaip 14 pav. pateikti paukščių vaizdai. Remiantis straipsniu apie DM-GAN [ZPC+19], šis modelis turėtų sugeneruoti geresnės kokybės vaizdus nei pateiktus 15 paveiksle, tačiau šiuo atveju galimai daug įtakos turėjo duomenų rinkinio dydis (*angl. batch size*), kuris buvo nustatytas į 4, dėl fizinių resursų stokos, kas galėjo būti tiesiog per mažai šiam generatyviniui besivaržančiam neuroniniui tinklui tinkamai funkcionuoti. Nors bandant skaityti aprašymą ir žiūrėti į sugeneruotą vaizdą galima matyti bendras detales, kurios padeda suprasti, kas yra pavaizduota 15 pav. pateiktuose sugeneruotuose vaizduose, bet iki realistiškumo šiems vaizdams dar reikia ne kartą būti tobulinamiems.

3.4. Eksperimentinės analizės apibendrinimas

Apžvelgus tris numatytus modelius – Semantic-Spatial Aware GAN, VQGAN-CLIP bei DM-GAN galima sulyginti iš eksperimentinės analizės gautus rezultatus ir pastebėti, ko galima tikėtis iš šių generatyvinių besivaržančių neuroninių tinklų vaizdams iš teksto generuoti. Visi gauti rezultatai pateikiami 4 lentelėje.

4 lentelė. Sugeneruotų vaizdų įvertinimai

GAN	Duomenų rinkiniai	IS ↑	FID ↓
Semantic-Spatial Aware GAN	CUB	5,13 ± 0,19	18,41
	COCO	20,52 ± 0,98	25,19
VQGAN-CLIP	CUB	-	-
	COCO	-	-
DM-GAN	CUB	4,71±0,17	16,19
	COCO	29,71±0,97	33,75

Pagal pateiktą lentelę galima matyti, kad atsižvelgiant į kiekybinius rodiklius DM-GAN modelis gali sugeneruoti vaizdus tikroviškiau, matomas 8,92 % tikroviškumo pagerėjimas pagal IS ir FID sumažėjimas 13,711 % lyginant su Semantic-Spatial Aware GAN gautais rezultatais, naudojantis CUB duomenų rinkiniu. Semantic-Spatial Aware GAN modelis šiuo atveju gali geriau sugeneruoti vaizdus naudojant COCO duomenų rinkinį – matomas 44,79 % tikroviškumo pagerėjimas pagal IS ir FID sumažėjimas 15,36 %. Taip pat kiekybiškai nieko negalima pasakyti apie VQGAN-CLIP, nes eksperimentiškai analizuojant šį modelį nebuvo remiamasi COCO ir CUB duomenų rinkiniais, jei ir būtų išmatuotos nurodytos metrikos nebūtų galima teigti, kad rezultatai yra tinkami, nes jie nebuvo gauti vienodomis sąlygomis.

Įvertinus sugeneruotus vaizdus vizualiai galima matyti, kad labiausiai išsiskiria savo sugeneruotų vaizdų tikroviškumu ir galimybe generuoti vaizdus įvairiais stiliais VQGAN-CLIP

generatyvinis modelis. Lyginant DM-GAN ir Semantic-Spatial Aware GAN sugeneruotus vaizdus, antrą vietą būtų galima skirti DM-GAN generatyviniui modeliui, nes šis modelis generuoja tikroviškesnius vaizdus remiantis pradiniais duomenimis bei šis vaizdas turi galimybę būti patikslintas, nes tai yra vienas iš DM-GAN veikimo principų – vaizdas gali būti pagerintas tiek kartų, kiek reikia.

Atsižvelgiant, kuris generatyvinis modelis sugeneruoja tiksliausius vaizdus naudojant COCO duomenų rinkinį, specifiškai šiuo atveju labai sunku nuspręsti, nes nei vienas GAN negeneruoja realistiškų vaizdų.

4. Egzistuojantys sugeneruotų vaizdų kokybiniai vertinimo būdai

Generatyvinių besivaržančių neuroninių tinklų sugeneruoti vaizdai gali būti vertinami kiekybiškai taip, kaip buvo atliekama eksperimentinėje analizėje – pasitelkus IS ir FID apskaičiavimą. Be jų yra ir daugiau įvairių matų, tokių kaip R tikslumas, nuotraukos generavimas laiko atžvilgiu, vaizdo gavimo našumas, tikslumo ir prisiminimo bei F balo skaičiavimai ir kt. Tačiau populiariausi ir dažniausiai naudojami išlieka pradinio balo (IS), FID ir R tikslumo skaičiavimai. Tad galima teigti, kad kiekybinių skaičiavimų pasiūlymų yra nemažai, o kaip yra su kokybiniais vertinimo būdais? Kaip buvo minėta pačioje pradžioje – dažniausias kokybinio vertinimo metodas yra vertinti vizualiai atkreipiant dėmesį į ryškumą, detalių atvaizdavimą, bendrą objekto formą ir kt. Šį vertinimo būdą taip pat galima pastebėti ir aprašytoje eksperimentinėje analizėje, tačiau tai nėra pats geriausias būdas, jei reikia vertinti didelį kiekį vaizdų, nes vertinimas gali priklausyti nuo daug kintamųjų ir tas pats paveikslas gali būti įvertintas skirtingai priklausomai nuo žmogaus ar kitų įtakos veiksnių, taip pat verta paminėti, kad kuo daugiau vaizdų yra vertinama tuo labiau išauga tikrinimo kaštai. Dėl to reikia automatinio būdo vertinti sugeneruotų nuotraukų kokybę, bet prieš bandant pasiūlyti naują metodą reikia išsiaiškinti metodus, kurie jau yra sukurti ir išsiaiškinti, kaip jie veikia, jų pliusus ir minusus, ką būtų galima panaudoti ar pagerinti.

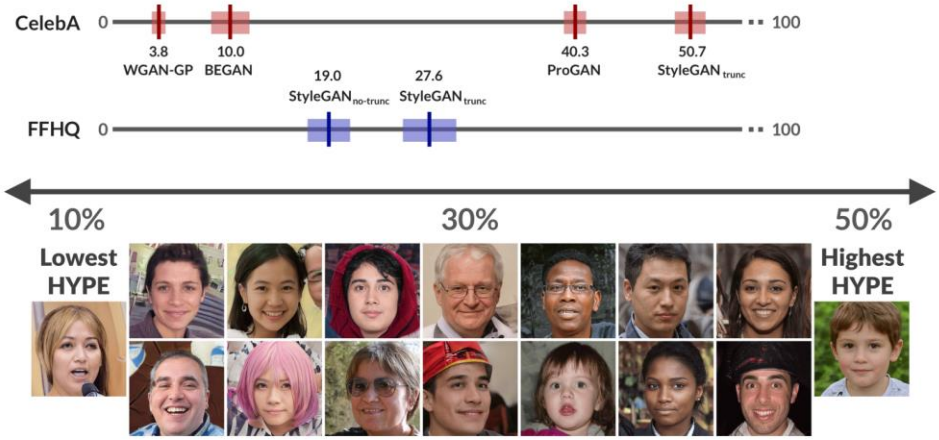
Apžvelgus šaltinius buvo galima rasti pagrindinius 5 kokybės vertinimo būdus:

- žmogaus akies suvokimo įvertinimas;
- *Neuroscore* matas;
- vertinimas, ko GAN negali sukurti;
- GAN išskaidymas;
- universalus padirbtų ir tikrų objektų detektorius.

Tad šiame skyriuje ir apžvelgsime plačiau kiekvieną iš šių nurodytų vertinimo būdų.

4.1. Žmogaus akies suvokimo įvertinimas

Pirmasis ir pagrindinis būdas įvertinti sugeneruotų nuotraukų tikroviškumą ir kokybę yra **žmogaus akies suvokimo įvertinimas** (*angl. Human Eye Perceptual Evaluation*) arba sutrumpinus HYPE. Straipsnyje [ZGK+19] buvo pasiūlytas žmogaus kilpoje (*angl. human-in-the-loop*) vertinimo metodas, pagrįstas psichofizikos vizualinio suvokimo tyrimais. Straipsnio autoriai pateikia du galimus HYPE variantus. Pirmasis ($Hype_{time}$) matuoja regimąjį suvokimą pagal prisitaikančius laiko apribojimus, siekiant nustatyti ribą, nuo kurios modelio sugeneruoti rezultatai atrodo realūs (pvz., 250 ms). Antrasis variantas ($Hype_{\infty}$) matuoja žmogaus klaidų lygį rodant sugeneruotus ir realius vaizdus be laiko apribojimų. 16 paveiksle galima matyti, kokie rezultatai buvo gauti atlikus tyrimą [Bor22].

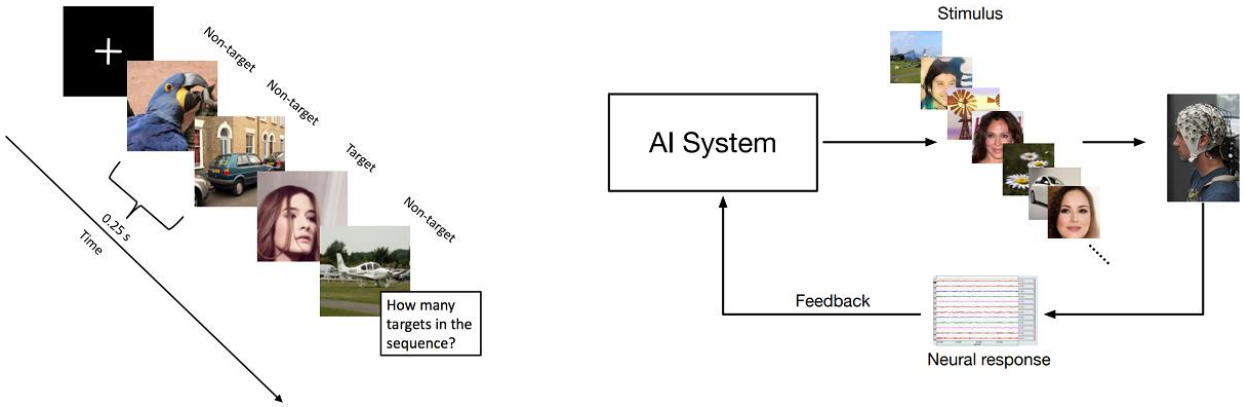


16 pav. HYPE metodo gauti rezultatai tyrime. Šaltinis [ZGK+19]

16 pav. pateikiami du grafai (vienas su skaičiais nuo 0 iki 100, kitas su nuotraukomis), kuriuose matoma, kaip buvo vertinamas tyrimas. Grafe su skaičiais yra pavaizduotos dvi tiesės su skirtingais pavadinimais, šie pavadinimai nurodo, kad tyrime buvo naudoti CelebA ir FFHQ duomenų rinkiniai. Šie rinkiniai buvo taikomi norint apskaičiuoti HYPE balus, jei rezultatas yra gaunamas lygus 50% tai reiškia, kad sugeneruoti rezultatai yra nesiskiriantys nuo realių, o rezultatas viršijantis 50% reiškia hiperrealizmą. Apatinis grafas (grafas su nuotraukomis) atvaizduoja paveikslų pavyzdžius, atrinktus naudojant StyleGAN atrinkimo (*angl. truncation*) triuką, parengtą naudojant FFHQ duomenų rinkinį. Dešinėje pusėje esantys vaizdai turi aukščiausius HYPE balus (t. y. pasižymi aukščiausiu suvokimo tikslumu). [KZZ+19] straipsnio autoriai pasiūlė metodą, aproksimuojantį HYPE, ir pranešė apie 66% tikslumą prognozuojant žmogaus vaizdų tikroviškumo balus, atitinkantį žmonių tarpusavio sutarimo rodiklį. Pagrindinis žmogaus vertinimo metodų, tokių kaip HYPE, trūkumas yra jų mastelio nustatymas [Bor22].

4.2. *Neuroscore* matas

Dar vienas būdas vertinti sugeneruotų vaizdų kokybę yra naudoti ***Neuroscore*** matą, pasiūlytą [ZGA+19] straipsnyje. Šio mato apskaičiavimo metu yra naudojami neuroniniai signalai ir greitas serijinis vaizdinis pateikimas (*angl. rapid serial visual presentation arba RSVP*), kad būtų galima tiesiogiai išmatuoti žmogaus suvokimo reakciją į generuojamus stimulus. Tyrimo metu dalyviams buvo nurodoma atkreipti dėmesį į tikslinius vaizdus (realius ir sugeneruotus), kuriuose didesnė vaizdų dalis buvo sugeneruoti (17 pav.).

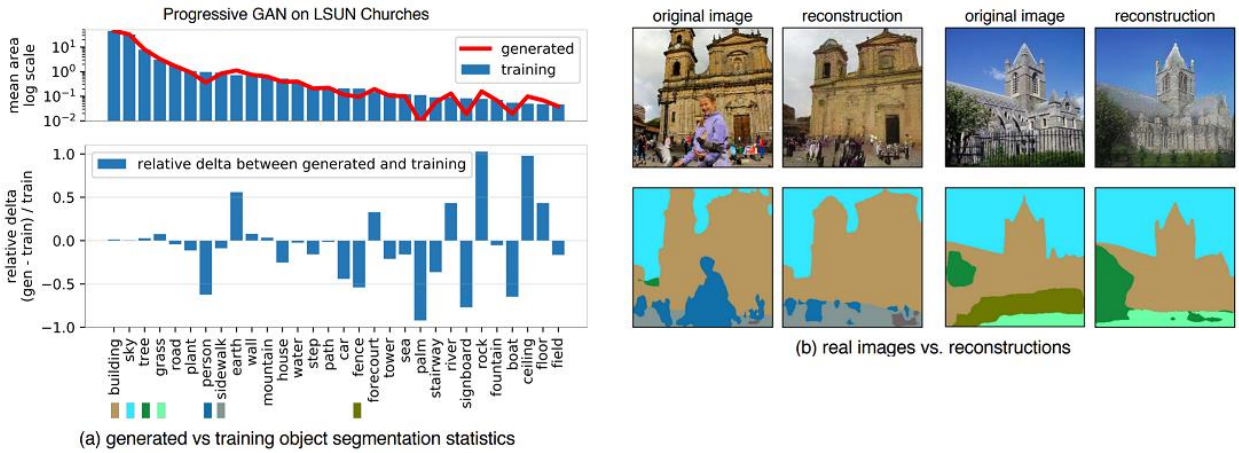


17 pav. Neuroscore tyrime naudota neuro-AI sąsajos schema. Šaltinis [ZGA+19]

Ši paradigma, vadinama *„oddball“* paradigma, dažniausiai naudojama su įvykiais susijusiam potencialui (ERP) P300 sukelti, o tai yra teigiamas įtampos poslinkis, kuris paprastai atsiranda nuo 300 ms iki 600 ms po to, kai yra pastebimas retai atsirandantis vizualus tikslas. 17 paveiksle parodytas šio matavimo metodo veikimo vaizdas. Straipsnyje autoriai teigia, kad ***Neuroscore*** labiau atitinka žmogaus vertinimus, palyginti su įprastinėmis metrikomis. Jie taip pat išmokė konvoliucinį neuroninį tinklą tiesiogiai prognozuoti *Neuroscore* iš GAN sukurtų vaizdų, nenaudojant neuroninių atsakymų.

4.3. Vertinimas, ko GAN negali sukurti

Kitokį požiūrį į kokybinį sugeneruotų nuotraukų vertinimą pateikė Bau ir kt. [BZW+19] straipsnyje. Autoriai nepateikia specifinio pavadinimo savo metodui, tad metodo pavadinimu galima laikyti straipsnio pavadinimą, tad šis metodas bando **vertinti, ko GAN negali sukurti**. Autoriai taiko semantinį segmentavimo tinklą, kad palygintų segmentuotų objektų pasiskirstymą sugeneruotuose vaizduose ir realiuose vaizduose. Statistinių duomenų skirtumai atskleidžia objektų klases, kurias GAN praleidžia. Jų metodas leidžia vizualizuoti GAN neatitikimus praleistai klasei ir palyginti atskirų vaizdų ir jų apytikslių inversijų skirtumus. 18 pav. parodytas šis metodas.



18 pav. Vertinimo, ko GAN negali sukurti metodo vizualizacija. Šaltinis [BZW+19]

18 paveikslo a dalyje galima matyti objektų segmentacijų pasiskirstymą LSUN bažnyčių mokymo aibėje ir atitinkamą pasiskirstymą pagal sugeneruotus vaizdus. Tokius vaizdus, kaip žmonės, automobiliai ir tvoros, generatorius išmeta. b) realių vaizdų poros ir jų rekonstrukcijos, kuriose negalima sugeneruoti atskirų žmogaus ir tvoros pavyzdžių.

4.4. GAN išskaidymas

Metais anksčiau Bau ir kt. [BZS+18] straipsnyje pasiūlė **GAN išskaidymo** (*angl. GAN Dissection*) metodiką. Šiuo metodu autoriai siūlo metodą, kaip išskaidyti ir vizualizuoti vaizdų generatoriaus vidinį veikimą. Pagrindinė idėja yra nustatyti GAN vienetus (t. y. generatoriaus neuronus), kurie generuojamuose vaizduose yra atsakingi už semantines sąvokas ir objektus (pvz., medį, dangų ir debesis). Turint tokio lygio neuronų smulkumą, galima redaguoti esamus vaizdus (pvz., pridėti arba pašalinti medžius, kaip parodyta 19 pav.), priverstinai aktyvuojant ir deaktyvuojant (išjungiant) atitinkamus norimų objektų vienetus. Aprašyta technika taip pat leidžia sugeneruotuose vaizduose rasti artefaktus ir todėl gali būti naudojama vertinant ir tobulinant GAN.

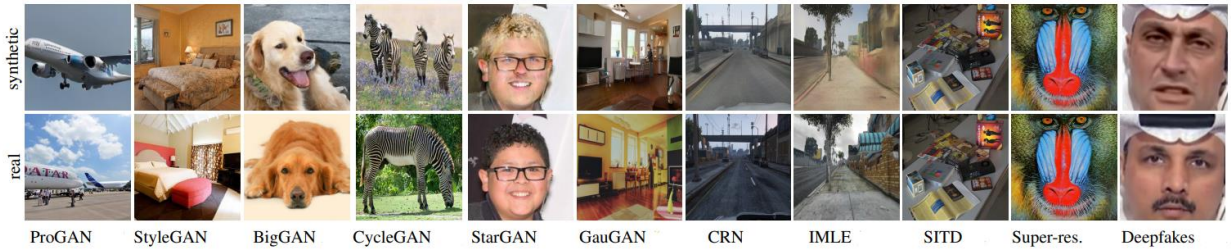
19 pav. GAN išskaidymo apibendrinimas. Šaltinis [BZS+18]

19 paveiksle galima matyti GAN išskaidymo apžvalgą [BZS+18]. Nuotraukose galima matyti santykių tarp atvaizdavimo vienetų ir medžių išvesties matavimas naudojant (a) išskaidymą ir (b) intervenciją. Išskaidymas matuoja susikirtimą tarp vieneto u ir sąvokos c, lygindamas jo slenkstinį padidinto pavyzdžio šilumos žemėlapį su sukurto vaizdo $s_c(x)$ semantiniu segmentavimu. Intervencija matuojamas priežastinis vienetų U rinkinio poveikis sąvokai c,

lyginant šių vienetų įjungimo (vieneto įterpimas) arba išjungimo (vieneto panaikinimas) poveikį. Segmentavimas s_c atskleidžia, kad medžių padaugėja po įterpimo ir sumažėja po pašalinimo. Vidutinis medžių pikselių skirtumas matuoja vidutinį priežastinį poveikį. Trečioje eilutėje nuo viršaus taikomas skaidymo metodas sugeneruotam lauko bažnyčios vaizdui. Disekcijos metodas taip pat gali būti naudojamas diagnozuoti ir tobulinti GAN, nustatant ir pašalinant artefaktus sukeliančius vienetus [Bor22].

4.5.Universalus padirbtų ir tikrų objektų detektorius

Paskutinis rastas metodas, kuris buvo siūlomas Wang ir kt. [WWZ+20] straipsnyje yra – **universalus padirbtų ir tikrų objektų detektorius**. Autoriai uždavė klausimą, ar įmanoma sukurti „universalų“ detektorių, skirtą atskirti tikrus vaizdus nuo sugeneruotų, naudojant generuotų vaizdų duomenų rinkinį, sugeneruotą 11 CNN pagrįstų generatyvinių modelių (20 paveikslas)



20 pav. CNN GAN generuotų nuotraukų pavyzdžiai [WWZ+20]

20 paveiksle parodyta, kad klasifikatorius, apmokytas atskirti vaizdus, sukurtus tik vieno GAN (ProGAN, kairiajame stulpelyje) nuo realių vaizdų, taip pat gali aptikti vaizdus, kurie yra sukurti kitų generatyvinių modelių (likusieji stulpeliai) [WWZ+20].

Šiuo straipsniu autoriai parodo, kad vaizdų klasifikatorius, išmokytas naudojant tik vieną konkretų CNN generatorių, gali gerai apibendrinti nematytas architektūras, duomenų rinkinius ir mokymo metodus. Jie pabrėžia, kad šiandieniniai CNN generuojami vaizdai turi bendrų sisteminių trūkumų, neleidžiančių jiems pasiekti tikrovišką vaizdų generavimą. Apie panašius darbus pranešė [CBL+20], [YDF19]. Pritardamas šiems rezultatams, Gragnaniello ir kt. [GCM+21] taip pat daro išvadą, kad dar toli gražu neturime patikimų priemonių GAN vaizdų nustatymui [Bor22].

4.6. Sugeneruotų vaizdų kokybinių vertinimo būdų apibendrinimas

Apžvelgus 5 paminėtus vertinimo būdus galima matyti, kad visi šie būdai matuoja generuotų vaizdų kokybę labai skirtingai. Aprašymuose galima pastebėti, kad keli iš šių vertinimo būdų yra skirti ne tik vertinimui, bet ir nuotraukos manipuliacijai – GAN išskaidymo metodas bei vertinimas, ko GAN negali sukurti. Tad šie metodai gali nebūti pirmieji mintyje renkantis vertinimo būdą vien tam, kad būtų įvertinta ar sugeneruotas vaizdas yra tikras, nes šie vertinimo

būdai nebūtinai pateiks reikiamus rezultatus, kadangi yra fokusuojamasi į detalių išskyrimą ar jų atpažinimą, kas nepasako ar bendroje visumoje nuotrauka yra sugeneruota realistiškai.

Atsižvelgiant į *Neuroscore* matą, galima galvoti, kad šis vertinimas jau duoda tokį rezultatą, kokio reikėtų – yra įvertinama sugeneruoto vaizdo kokybė, bei tam jau yra sukurtas konvoliucinis neuroninis tinklas, kuris gali bandyti atspėti ar nuotrauka yra tikra, pagal turimus duomenis, gautus iš atlikto tyrimo. Tačiau šio mato konvoliucinio neuroninio tinklo apmokymas yra fiksuotas tiek, kiek buvo apmokyta tyrimo metu, o norint tęsti šio neuroninio tinklo treniravimą reikia turėti daug resursų, kas yra labai brangu, vien įrangos turėjimas jau sukelia problemų. Ir galimai šis variantas dėl to ir nėra pasirenkamas vertinant sugeneruotų vaizdų kokybę, nes laikui bėgant generuojamos nuotraukos yra vis realistiškesnės, o apmokytas metodas daugiau duomenų negauna, kas gali daryti įtakos vertinant.

Universalus padirbtų ir tikrų objektų detektorius atlieka savo darbą – iš pateiktų dviejų nuotraukų atrenka, kuri yra sugeneruota ir kuri yra tikra. Naudojant šį kokybės vertinimo metodą atsiranda nemažai papildomo darbo – vienai sugeneruotam vaizdui surasti vieną tikrą. Vadinasi išauga darbo kaštai, ypatingai, jei nuotraukų kiekis yra didelis. Dėl šios priežasties universalus padirbtų ir tikrų objektų detektorius gali nebūti tinkamas pasirinkimas, jei norima vertinti sugeneruoto vaizdo realistiškumą automatiškai su kuo mažiau žmogaus įsikišimo. Aptarus visų vertinimo metodikų trūkumus ir kodėl šie metodai gali būti ne pirmas pasirinkimas vertinant sugeneruoto vaizdo kokybę prieinama prie išvados, kad pagrindinis vertinimo būdas išlieka HYPE, nes jis yra paprasčiausias tiek laiko, tiek sudėtingumo atžvilgiu ir juo gali naudotis bet kas be jokių papildomų pastangų, taip pat kaip rodo tyrimas, šis metodas nėra prastas. Tačiau automatinis būdas, kuris patikrintų kokybę ir įvertintų ar sugeneruotas vaizdas yra suprantama išlieka siekiamybė, nes naudojant tokį būdą didelį kiekį vaizdų būtų galima įvertinti automatiškai su minimaliu žmogaus įsikišimu dėl aiškesnio gaunamo rezultato.

5. Sukurto kokybinio vertinimo metodo eksperimentinis tyrimas

Aptarus ketvirtame skyriuje jau esamus kokybės vertinimo būdus bei išsiaiškinus, kodėl HYPE arba kitaip, žmogaus akies suvokimo įvertis vis dar išlieka pagrindinis sugeneruotų vaizdų kokybinis vertinimo būdas reikia aptarti siūlomą naują automatizuotą vertinimo būdą.

Naujas kokybės vertinimo būdas gali įvertinti automatiškai ar pateiktas sugeneruotas vaizdas yra realistiškas. Vertinti sugeneruotą vaizdą galima pagal savo turimą vaizdų kolekciją, kurią reikia nurodyti kartu su vertinamu vaizdu arba nenurodyti nieko ir tuomet realistiška nuotrauka yra paimama iš interneto pagal tai, kas yra siūloma parodžius sugeneruotą vaizdą. Taip yra gaunamos dvi nuotraukos – yra žinoma, kuri sugeneruota, o kuri turi būti tikra ir tuomet yra vertinama, kaip stipriai skiriasi sugeneruotas vaizdas nuo realaus. Šis metodas yra panašus į universalų padirbtų ir tikrų objektų detektorių, tačiau naudojant šį kokybės vertinimo metodą nereikia ieškoti antros realistiškos nuotraukos, kadangi tai yra padaroma automatiškai. Įvertinus sugeneruotą nuotrauką metodas grąžina atsaką, kuris nurodo, kiek tikėtina, kad generuotas vaizdas yra atrodantis realistiškai. Kokybės vertinimo metodas nuotrauką gali priskirti į vieną iš trijų grupių – vaizdas sugeneruotas tikroviškai, vaizdas galimai sugeneruotas tikroviškai bei vaizdas neatrodo tikroviškai.

Trumpai aptarus naujo kokybės vertinimo būdo idėją, šiame skyriuje bus detaliau aptariamas pasiūlytas kokybės vertinimo metodas, metodo realizacija bei kokie rezultatai yra gaunami bandant įvertinti prieš tai GAN tyrimuose sugeneruotas nuotraukas.

5.1.Sukurto kokybės vertinimo metodo apžvalga

Kokybės vertinimas yra svarbus procesas, kuris nustato vaizdo vizualinę kokybę ir tinkamumą tam tikram tikslui, specifiškai šiuo atveju tikslas yra nustatyti ar sugeneruota nuotrauka yra realistiška. Vertinimas apima įvairių vaizdo aspektų analizavimą, tokių kaip skiriamoji geba, ryškumas, spalvų tikslumas, šviesumas, kontrastas, triukšmo lygis ir bendra kompozicija. Kai tai atlieka žmogus, tai gaunasi intuityviai, tačiau norint tai automatizuoti reikia pridėti matematinių skaičiavimų, kurie, deja, intuicijos neturi. Nors intuicijos neturi, bet automatiniam kokybės vertinimo metodui sukurti yra tinkamas. Yra keletas dažniausiai naudojamų matavimų, skirtų įvertinti vaizdų panašumą:

- **Vidutinė kvadratinė klaida (MSE)**: MSE matuoja vidutinį kvadratinį dviejų vaizdų pikselių intensyvumo skirtumą [ZBS+04]. Šis skaičiavimas paprastai naudojamas kaip pagrindinė panašumo metrika, kai mažesnė MSE vertė rodo didesnį panašumą.
- **Struktūrinio panašumo indeksas (SSIM)**: SSIM lygina dviejų vaizdų struktūrinę informaciją, atsižvelgiant į ryškumą, kontrastą ir struktūrinį panašumą [ZBS+04]. Šis skaičiavimas suteikia balą nuo -1 iki 1, kur 1 reiškia tobulą panašumą.

- **Didžiausias signalo ir triukšmo santykis (PSNR)**: PSNR matuoja didžiausios galimos signalo (šiuo atveju vaizdo) galios ir jį veikiančio iškraipymo galios santykį [ZBS+04]. Šis skaičiavimas dažnai naudojamas vaizdo suspaudimo algoritmams vertinti, kai didesnės PSNR vertės rodo didesnį panašumą.
- **Kosinusinis panašumas**: Kosinuso panašumas matuoja kampo tarp dviejų vektorių, vaizduojančių vaizdų pikselių intensyvumą, kosinusą [PLF15]. Panašumas apskaičiuojamas remiantis vektorių orientacija, o ne dydžiu, todėl gaunama vertė nuo -1 iki 1, kai 1 reiškia visišką panašumą.
- **Euklido atstumas**: Euklido atstumas yra paprasta atstumo metrika, pagal kurią apskaičiuojamas tiesiaeigis atstumas tarp dviejų vektorių (vaizdų) [SWS+00]. Jis matuoja pikselių intensyvumo skirtumo dydį. Mažesni atstumai rodo didesnį panašumą.
- **Histogramų palyginimas**: Histogramomis pagrįsti metodai lygina vaizdų spalvų pasiskirstymą [SWS+00]. Dažniausiai taikomi šie metodai: histogramų susikirtimo, *Bhattacharyya* atstumo ir *chi-kvadrato* atstumo [HO06]. Šiais metodais panašumo balai nustatomi pagal spalvų histogramų panašumą.
- **Giliuoju mokymusi pagrįsti požymiai**: giliojo mokymosi metodu pagrįsti požymiai, išgauti iš giliųjų neuroninių tinklų (pvz., naudojant iš anksto apmokytus modelius, tokius kaip VGG, *ResNet* arba *Inception*), tapo populiarūs vertinant vaizdų panašumą [WYW+22]. Šie metodai naudoja išmoktas reprezentacijas, kad užfiksuotų aukšto lygio vaizdinius požymius.

Pagal tai galima pastebėti, kad vieno universalaus skaičiavimo nėra patikrinti ar generuotas vaizdas yra realistiškai atrodantis. Tad šiam specifiniui atvejui buvo pasirinktas kosinusinio panašumo skaičiavimas, kartu su giliuoju mokymusi pagrįstu požymių atradimu, nes kuriant naują kokybės vertinimo metodą ši kombinacija išryškėjo, kaip gaunanti geriausius rezultatus. Taip pat, kuriant vertinimo metodą, nustatyti sugeneruoto vaizdo kokybę buvo pabandyta pritaikyti ir kitus panašumo skaičiavimus, kurie yra pateikiami sąraše.

Pritaikius vidutinės kvadratinės klaidos skaičiavimą gauti rezultatai nebuvo tokie, kokių buvo tikimasi – nepriklausomai nuo to, ar vaizdas sugeneruotas atpažįstamai ar ne visi vaizdai buvo įvertinti ~0.01 ir mažiau, kas nusako, jog nuotrauka ir sugeneruotas vaizdas praktiškai yra identiški, kas nebuvo tiesa, nes sugeneruoti vaizdai skyrėsi tiek fonu, tiek sugeneruoto pagrindinio objekto atvaizdavimu (jo forma, dydis, žvilgsnio pusė ir kt.). Kitaip tariant, MSE skaičiavimas pateikė gerus rezultatus net tada, kai sugeneruotas vaizdas nebuvo atpažįstamas, tačiau spalvos

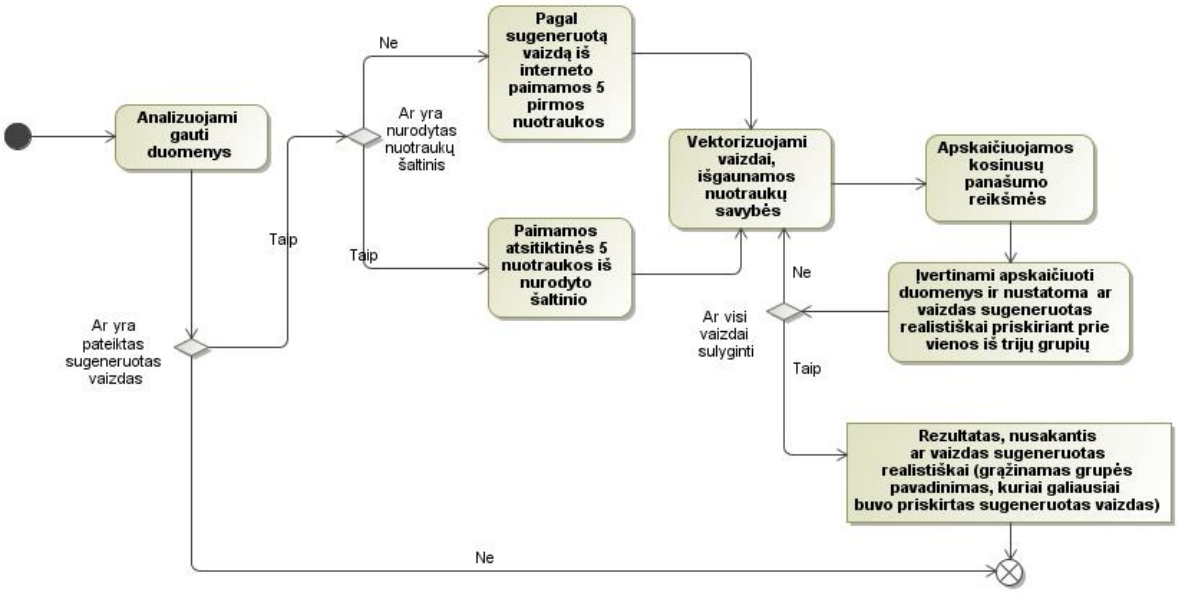
sutapo su realių nuotraukų turimomis spalvomis, dėl to vidutinės kvadratinės klaidos skaičiavimas buvo priskirtas, kaip netinkamas kokybės vertinimui šiuo atveju.

Pabandžius pritaikyti SSIM skaičiavimą nustatyti ar sugeneruotas vaizdas yra panašus į tikrą buvo gauti rezultatai, kurių buvo tikimasi. Tačiau pabandžius su didesne generuotų vaizdų įvairove buvo galima pastebėti tendenciją, kad pateikus kai kuriuos generuotus vaizdus, kuriems buvo randamos nuotraukos neturinčios panašaus fono, vertinimas pakrypdavo į neigiamą pusę nors vizualiai vaizdai buvo sugeneruoti gana realistiškai. Todėl SSIM buvo atmestas, kadangi šis skaičiavimas daugiau atsižvelgia į ryškumą, kontrastą, kas reiškia, jei sugeneruotas vaizdas bus kitokio fono nei, kad rasta realistiška nuotrauka, kaip ir buvo išsiaiškinta eksperimentuojant su šiuo kokybės vertinimo metodu, iškart bus gaunami žemi balai ir tai nenusakys ar pats vaizdas yra sugeneruotas realistiškai.

Didžiausias signalo ir triukšmo santykis bei histogramų palyginimas iškart buvo atmesti šiai užduočiai atlikti. Didžiausias signalo ir triukšmo santykis buvo atmestas, nes nauju kokybės vertinimo metodu yra siekiama patikrinti realistiškos nuotraukos ir sugeneruoto vaizdo panašumą, ko šis skaičiavimas nedaro. O histogramų palyginimas buvo atmestas, nes šis skaičiavimas tikrina vaizdų spalvų pasiskirstymą ir jeigu realus vaizdas turės kitokį foną nei sugeneruotas vaizdas bus gaunami žemi vertinimai nepriklausomai nuo sugeneruoto vaizdo tikrosios kokybės.

Euklido atstumas taip pat buvo atmestas, kadangi eksperimentuojant su šiuo panašumo skaičiavimu buvo gaunami panašūs rezultatai nepriklausomai ar sugeneruotas vaizdas buvo išplaukęs ar atrodantis realistiškai.

Išbandžius kosinusinio panašumo skaičiavimą kartu su giliuoju mokymusi pagrįstu požymių atradimu buvo gaunamas tinkamiausias rezultatas, kuris pirmiausiai išgauna iš sugeneruoto vaizdo bei realios nuotraukos požymius ir juos sulygina naudojant kosinusų panašumo skaičiavimą, taip nustatant ar objektų vektorių orientacija yra panaši. Sukurto kokybės vertinimo metodo veikimas yra pateikiamas 21 paveiksle pasinaudojant veiklos diagrama.



21 pav. Kokybės vertinimo metodo veiklos diagrama

Kaip galima matyti, 21 paveiksle yra pateikiama sukurto kokybės vertinimo metodo veiklos diagrama, kurioje atsispindi, kaip veikia sukurtas metodas. Kaip galima matyti viskas prasideda nuo duomenų įvesties. Turi būti įvedamas sugeneruoto vaizdo adresas ir pasirinktinai nurodytas nuotraukų šaltinis, iš kurio bus imamos nuotraukos, kad patikrinti ar sugeneruota nuotrauka yra realistiška. Sugeneruoto vaizdo adresas yra privalomas, nes jei jis yra nenurodomas procesas yra baigiamas, kitu atveju yra tikrinama ar yra nurodytas nuotraukų šaltinis. Jei nuotraukų šaltinis yra nurodytas, tuomet iš jo yra paimamos penkios atsitiktinės nuotraukos, su kuriomis bus lyginamas sugeneruotas vaizdas, kad įvertinti ar vaizdas atrodo realistiškai. Kitu atveju paimamos penkios nuotraukos iš interneto. Paimamos specifiškai 5 nuotraukos, nes 1 nuotraukos yra tiesiog per mažai, kad nustatyti ar sugeneruotas vaizdas atrodo realistiškai. Tačiau šį skaičių visada galima keisti pagal savo poreikius. Turint visus vaizdus, kurių reikia, t.y. sugeneruotą vaizdą ir nuotraukas, su kuriomis bus lyginama, prasideda kokybės vertinimo procesas. Pirmiausia vaizdai yra vektorizuojami ir išskiriami vaizdų požymiai, naudojant *Imagenet* apmokytą modelį. Išgavus vaizdų požymius yra apskaičiuojamas vaizdų požymių panašumas pritaikius kosinusų panašumo skaičiavimą. Turint kosinusų panašumo rezultatą yra įvertinami apskaičiuoti duomenys ir nustatoma, ar palyginus su realia nuotrauka, sugeneruotas vaizdas atrodo panašus priskiriant vieną iš trijų grupių – vaizdas sugeneruotas tikroviškai, vaizdas galimai sugeneruotas tikroviškai bei vaizdas neatrodo tikroviškai. Šis procesas yra kartojamas tol, kol yra pateiktų realių nuotraukų ir galiausiai, kai yra sulyginamos visos nuotraukos yra gaunamas galutinis rezultatas, nusakantis ar sugeneruota nuotrauka atrodo realistiškai. Realistiškai sugeneruotas vaizdas yra laikomas tuomet, kai bent viena iš realių nuotraukų turėjo aukštą atitiktį su sugeneruotu vaizdu.

Kokybės vertinimo metodo kodas yra pateikiamas 1 priede.

5.2.Sukurto metodo eksperimentinis tyrimas

Sukūrus ir aprašius kokybės vertinimo metodo veikimą galima atlikti eksperimentinį tyrimą ir patikrinti, kaip šis metodas vertina GAN eksperimentinės analizės metu sugeneruotas nuotraukas. Bus tikrinamos keturios nuotraukos – dvi nuotraukos bus imamos su paukščiais, kurios buvo sugeneruotos naudojant CUB duomenų rinkinį. Viena nuotrauka bus sunkiai suprantama, kita bus realistiška, lyginant su kitomis nuotraukomis. Trečia nuotrauka bus sugeneruota naudojant COCO duomenų rinkinį. Iš šio rinkinio bus paimta tik viena nuotrauka, nes eksperimentinės analizės metu nebuvo sugeneruotos nei vienos nuotraukos, kurią būtų galima vadinti realistiška. Ketvirtoji nuotrauka bus paimta iš VQGAN sukurtų, nes eksperimentinės analizės metu šis GAN generavo nuotraukas remdamasis kitokiu duomenų rinkiniu.

Vertinama bus pritaikius sukurtą kokybės vertinimo metodą taip, kaip yra aprašyta sukurto metodo vertinimo apžvalgos skyriuje. Vienas sugeneruotas vaizdas bus vertinamas dviem atvejais – pagal internete gautas nuotraukas ir nuotraukas, kurios bus paimamos iš originalaus duomenų rinkinio, kuris buvo pateikiamas GAN modelių treniravimo metu (generatyvinių besivaržančių neuroninių tinklų eksperimentinės analizės metu). Kaip jau žinoma, įvertintas sugeneruotas vaizdas yra tada, kai naujasis kokybės vertinimo metodas priskiria sugeneruotą vaizdą į vieną iš trijų numatytų grupių – vaizdas sugeneruotas tikroviškai, vaizdas galimai sugeneruotas tikroviškai bei vaizdas neatrodo tikroviškai. Šios grupės yra nustatomos pagal kosinusų panašumo rezultatą, kuris, kaip jau žinoma grąžina rezultatą nuo -1(0) iki 1, kai 1 reiškia visišką panašumą. Tad šiam specifiniam eksperimentui kiekvienai iš grupių buvo nustatytas rėžis, pagal kurį buvo sprendžiama į kurią grupę sugeneruota nuotrauka patenka pagal kosinusų panašumo rezultatą. Atitiktis galima matyti 5 lentelėje.

5 lentelė. Kokybės vertinimo grupės

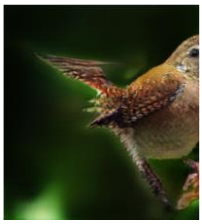
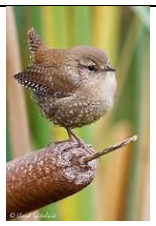
Grupės pavadinimas	Kosinusų panašumo rėžis
Vaizdas sugeneruotas tikroviškai	Daugiau nei 0,5
Vaizdas galimai sugeneruotas tikroviškai	Daugiau nei 0,35
Vaizdas neatrodo tikroviškai	Mažiau nei 0,35

Kaip galima matyti 5 lentelėje yra 3 grupės ir jų rėžiai buvo pasirinkti specifiškai tokie, nes atliekant šį eksperimentą tokie rėžiai tiksliausiai atvaizdavo turimų nuotraukų kokybę.

Aptarus, kas bus vertinama ir kaip, galima pradėti vertinti sugeneruotus vaizdus. Pirmasis vaizdas, kuris bus vertinamas yra paukštis, kuris yra sugeneruotas ne visai realistiškai. Vaizde nesimato paukščio galvos, sulieta uodega. Pirmiausia šis paukštis bus tikrinamas su nuotraukomis,

kurios yra gaunamos iš interneto. Sugeneruoto paukščio atvaizdas ir nuotraukos, su kuriomis buvo lyginamas sugeneruotas vaizdas bei gauti rezultatai pateikiami 6 lentelėje.

6 lentelė. Kokybės vertinimo metodo rezultatai pirmam sugeneruotam vaizdui

	Sugeneruota nuotrauka	**Lyginamosios nuotraukos**	**Metodo vertinimas**	**Galutinis vertinimas**
Lyginamosios nuotraukos paimtos iš interneto pagal sugeneruotą vaizdą			Vaizdas neatrodo tikroviškai (0,15)	Vaizdas neatrodo tikroviškai
			Vaizdas neatrodo tikroviškai (0,06)	
			Vaizdas neatrodo tikroviškai (0,23)	
			Vaizdas neatrodo tikroviškai (0,07)	
			Vaizdas neatrodo tikroviškai (0,22)	
Lyginamosios nuotraukos yra duodamos naudotojo			Vaizdas neatrodo tikroviškai (0,22)	Vaizdas neatrodo tikroviškai
			Vaizdas neatrodo tikroviškai (0,08)	
			Vaizdas neatrodo tikroviškai (0,07)	
			Vaizdas neatrodo tikroviškai (0,14)	
			Vaizdas neatrodo tikroviškai (0,08)	

Kaip galima matyti iš pateiktų rezultatų, kuriuos sugeneravo kokybės vertinimo metodas – šis vaizdas nėra sugeneruotas realistiškai nei lyginant su nuotraukomis, kurios yra paimtos iš interneto nei su nuotraukomis, kurios yra paimtos iš CUB duomenų rinkinio.

Antrasis generuotas vaizdas taip pat yra paukštis, tačiau šį kartą paukštis yra sugeneruotas realistiškai ir pažiūrėjus į paukštį, galima matyti ir suprasti, kad ten yra tikrai paukštis. Rezultatai yra pateikiami 7 lentelėje.

7 lentelė. Kokybės vertinimo metodo rezultatai antram sugeneruotam vaizdui

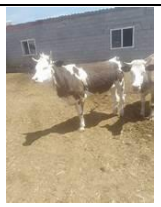
	Sugeneruota nuotrauka	**Lyginamosios nuotraukos**	**Metodo vertinimas**	**Galutinis vertinimas**
Lyginamosios nuotraukos paimtos iš interneto pagal sugeneruotą vaizdą			Vaizdas sugeneruotas tikroviškai (0,55)	Vaizdas sugeneruotas tikroviškai
			Vaizdas galimai sugeneruotas tikroviškai (0,45)	
			Vaizdas sugeneruotas tikroviškai (0,59)	
			Vaizdas sugeneruotas tikroviškai (0,51)	
			Vaizdas galimai sugeneruotas tikroviškai (0,36)	

Lyginamosios nuotraukos yra duodamos naudotojo			Vaizdas galimai sugeneruotas tikroviškai (0,45)	Vaizdas galimai sugeneruotas tikroviškai
			Vaizdas neatrodo tikroviškai (0,25)	
			Vaizdas galimai sugeneruotas tikroviškai (0,43)	
			Vaizdas neatrodo tikroviškai (0,29)	
			Vaizdas galimai sugeneruotas tikroviškai (0,43)	

Kaip galima matyti 7 lentelėje šį kartą rezultatai truputį išsiskyrė. Pagal iš internete gautas nuotraukas sugeneruota nuotrauka yra įvertinama, kaip realistiškai sugeneruota, tačiau pagal nuotraukas, kurios yra pateikiamos iš duomenų rinkinio galutinis rezultatas yra gaunamas, kaip vaizdas galimai sugeneruotas tikroviškai. Atkreipus dėmesį į pačias nuotraukas, galima pastebėti, kad daugiau įtakos galėjo turėti fone esantys vaizdai, nes duomenų rinkinyje fonas buvo detalesnis. Tačiau bendroje visumoje vaizdas pagal naują kokybės vertinimo metodą atrodo realistiškai, arba galimai realistiškai sugeneruotas priklausomai nuo nuotraukų kolekcijos, kuri yra pateikiama.

Trečias vaizdas, kuris yra vertinamas naudojant naujai sukurtą kokybės vertinimo metodą, yra sugeneruotas naudojant COCO duomenų rinkinį. Šiame vaizde yra bandoma atvaizduoti karvę, tačiau ne pilnai realistišką. Generuotą vaizdą ir gautus rezultatus galima matyti 8 lentelėje.

8 lentelė. Kokybės vertinimo metodo rezultatai trečiam sugeneruotam vaizdui

	Sugeneruota nuotrauka	**Lyginamosios nuotraukos**	**Metodo vertinimas**	**Galutinis vertinimas**
Lyginamosios nuotraukos paimtos iš interneto pagal sugeneruotą vaizdą			Vaizdas neatrodo tikroviškai (0,09)	Vaizdas neatrodo tikroviškai
			Vaizdas neatrodo tikroviškai (0,08)	
			Vaizdas neatrodo tikroviškai (0,11)	
			Vaizdas neatrodo tikroviškai (0,12)	
			Vaizdas neatrodo tikroviškai (0,11)	
Lyginamosios nuotraukos yra duodamos naudotojo			Vaizdas neatrodo tikroviškai (0,17)	Vaizdas neatrodo tikroviškai
			Vaizdas neatrodo tikroviškai (0,11)	
			Vaizdas neatrodo tikroviškai (0,16)	
			Vaizdas neatrodo tikroviškai (0,09)	
			Vaizdas neatrodo tikroviškai (0,12)	

8 lentelėje pateikiami 3-ios nuotraukos rezultatai, kaip galima matyti iš COCO duomenų rinkinio sugeneruota karvė tikrai neatrodo realistiškai, tą patį nusako ir rezultatai, kurie yra gaunami naudojant naujai sukurtą kokybės vertinimo metodą.

Paskutinysis vaizdas, kuris bus vertinamas naudojant naujai sukurtą kokybės vertinimo būdą yra vaizdas sugeneruotas naudojant VQGAN-CLIP. Šis vertinimas bus atliekamas tik su nuotraukomis iš interneto, kadangi eksperimentiškai analizuojant šį metodą nebuvo naudojamas nei COCO nei CUB duomenų rinkinys apmokyti nurodytą GAN. Rezultatai pateikiami 9 lentelėje.

9 lentelė. Kokybės vertinimo metodo rezultatai ketvirtam sugeneruotam vaizdui

	Sugeneruota nuotrauka	**Lyginamosios nuotraukos**	**Metodo vertinimas**	**Galutinis vertinimas**
Lyginamosios nuotraukos paimtos iš interneto pagal sugeneruotą vaizdą			Vaizdas galimai sugeneruotas tikroviškai (0,45)	Vaizdas galimai sugeneruotas tikroviškai
			Vaizdas neatrodo tikroviškai (0,24)	
			Vaizdas neatrodo tikroviškai (0,27)	
			Vaizdas neatrodo tikroviškai (0,33)	
			Vaizdas neatrodo tikroviškai (0,32)	

Kaip galima matyti 9 lentelėje pateiktas vaizdas įvertintas, kaip galimai sugeneruotas tikroviškai ir atkreipus dėmesį į tai, su kokiomis nuotraukomis sugeneruotas vaizdas buvo lyginamas tikrai galima pritarti šiam faktui, nes nei vienoje iš pateiktų nuotraukų nėra pilnai violetinio paukščio. Taip pat verta atkreipti dėmesį, kad sugeneruotas paukštis turi daug detalių,

dėl ko galėjo gauti aukštesnį balą. Bei įsižiūrėjus į vaizdą, galima pastebėti, kad šiam nėra sugeneruotos pilnai kojos ir turi drugelio sparną, kad galėjo paveikti rezultatus ir paukštis atsidūrė ties grupe, kuri nurodo, kad vaizdas galimai sugeneruotas tikroviškai.

5.3.Sukurto metodo eksperimentinio tyrimo apibendrinimas

Šiame skyriuje apžvelgus, kaip buvo sukurtas naujasis kokybės vertinimo metodas bei eksperimentiškai jį išbandžius galima teigti, jog naujas kokybės vertinimo būdas gali įvertinti automatiškai ar pateiktas sugeneruotas vaizdas yra realistiškas. Vertinama yra apskaičiavus dviejų ar daugiau pateiktų realių nuotraukų ir sugeneruoto vaizdo kosinusų panašumą. Jei rezultatas telpa į tam tikrus rėžius, tuomet galima teigti, jog vaizdas yra sugeneruotas tikroviškai. Taip pat naudojant šį kokybės vertinimo metodą galima pasirinkti pagal ką bus vertinama ar sugeneruoti vaizdai yra realūs, t.y. pagal savo turimą vaizdų kolekciją, ar internetą.

Atlikus eksperimentinę sukurto vertinimo metodo analizę buvo pateikta, kaip šis metodas vertina vaizdus ir pastebėta, jei vaizdas yra sugeneruotas prastai (nerealistiškai), tiek pagal nuotraukas, gautas iš interneto, tiek pagal iš anksto numatytą vaizdų kolekciją vaizdas yra pripažįstamas, kaip neatvaizduojantis realių objektų. Tačiau, jei vaizdas panašėja į atpažįstamą akiai objektą, tuomet rezultatai gerėja ir priklausomai nuo to, iš kur yra imamos realios nuotraukos gali skirtis rezultatai. Taip pat galima pastebėti, kad imtos iš interneto nuotraukos sulyginus su generuota nuotrauka parodė geresnius rezultatus lyginant su vertinimu, kai nuotraukos buvo imtos iš duomenų rinkinio.

Išvados

1. Analitiškai apžvelgus tokius generatyvinius besivaržančius neuroninius tinklus vaizdams iš teksto generuoti, kaip DF-GAN, Semantic-Spatial Aware GAN, DM-GAN, Obj-GAN, VQGAN-CLIP nustatyta, kad minėti modeliai yra vertinami kokybiškai – apžiūrint sugeneruotą nuotrauką ir įvertinant vizualiai, bei kiekybiškai – taikant FID (Frechet Inception Distance), IS (Inception Score) ir R (Relative Distinguishing Power) tikslumo skaičiavimus.
2. Atlikus Semantic-Spatial Aware GAN, DM-GAN ir VQGAN-CLIP generatyvinių besivaržančių neuroninių tinklų eksperimentinę analizę buvo pastebėta, kad duoti modelių kodai dažniausiai neveikia dėl senų/nebepalaikomų bibliotekų ir pasenusių metodų, bet juos sutvarkius/atnaujinus generuojami rezultatai yra gana panašūs į pateiktus šaltiniuose neatsižvelgiant į tai, kad buvo turėta mažiau resursų.
3. Apžvelgus dabar egzistuojančius kokybės vertinimo matus buvo nustatyta, kad vizualus vertinimas išlieka pagrindiniu dėl kitų metodų ne tokio paprasto pritaikymo, bet tuo pačiu vizualus vertinimas yra reikalaujantis daug resursų, jei reikia įvertinti didelį kiekį generuotų vaizdų. Todėl buvo pasiūlytas naujas kokybės vertinimo metodas, kuris būtų grįstas nuotraukų panašumo vertinimu – viena nuotrauka būtų sugeneruota, kita (ar daugiau) nuotrauka/nuotraukų paimta iš nuotraukų bibliotekos (pavyzdys – *google*).
4. Atlikus eksperimentinius sukurto metodo tyrimus, buvo nustatyta, kad neaiškūs/nerealistiški vaizdai buvo įvertinami labai panašiai nepriklausomai nuo nuotraukų šaltinio, o vertinant realistiškesnius vaizdus turimas nuotraukų šaltinis lėmė, kiek panašumų buvo rasta. Nors metodas jau yra sukurtas, būtų galima atlikti tolesnius tyrimus ir patobulinimus, kad būtų pagerintas metodo tikslumas ir efektyvumas.

Literatūra

[AKK19] Alqahtani Hamed, Kavakli-Thorne Manolya, Kumar Ahuja Dr. Gulshan. (2019). Applications of Generative Adversarial Networks (GANs): An Updated Review. *Archives of Computational Methods in Engineering*, 525-552.

[BA22] Mohammed Berrahal, Mostafa Azizi. (2022). Optimal text-to-image synthesis model for generating portrait. *Indonesian Journal of Electrical Engineering and Computer Science*, 972-979.

[Bor22] Borji, A. (2022). Pros and cons of GAN evaluation measures: New developments. *Computer Vision and Image Understanding*, 1077-3142.

[BZS+18] Bau, David and Zhu, Jun-Yan and Strobelt, Hendrik and Zhou, Bolei and Tenenbaum, Joshua B. and Freeman, William T. and Torralba, Antonio. (2018). GAN Dissection: Visualizing and Understanding Generative Adversarial Networks. arXiv.

[BZW+19] Bau, David and Zhu, Jun-Yan and Wulff, Jonas and Peebles, William and Strobelt, Hendrik and Zhou, Bolei and Torralba, Antonio. (2019). Seeing What a GAN Cannot Generate. arXiv.

[CBK+22] Crowson, Katherine and Biderman, Stella and Kornis, Daniel and Stander, Dashiell and Hallahan, Eric and Castricato, Louis and Raff, Edward. (2022). *VQGAN-CLIP: Open Domain Image Generation and Editing with Natural Language Guidance.* arXiv.

[CBL+20] Chai, Lucy and Bau, David and Lim, Ser-Nam and Isola, Phillip. (2020). What makes fake images detectable? Understanding properties that generalize. *European Conference on Computer Vision* (p. 103-120). Springer.

[CJ18] Cristian Bodnar, Jonathan Shapiro. (2018). *Text to Image Synthesis Using Generative Adversarial Networks.* Manchester: University of Cambridge.

[ERO21] Esser, Patrick and Rombach, Robin and Ommer, Bjorn. (2021). Taming Transformers for High-Resolution Image Synthesis. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (p. 12873-12883). CVPR.

[ESE13] Elhoseiny M., Saleh B., Elgammal A. (2013). Write a Classifier: Zero-Shot Learning Using Purely Textual Descriptions. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (p. 2584-2591). Sydney: ICCV.

[FEH+09] Farhadi A., Endres I., Hoiem D., Forsyth D. (2009). Describing objects by their attributes. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 177-1785.

[FHR+21] Frolov Stanislav, Hinz Tobias, Raue Federico, Hees Jörn, Dengel Andreas. (2021). Adversarialtext-to-imagesynthesis:Areview. *Neural Networks*, 187-209.

[GCM+21] Gragnaniello, Diego and Cozzolino, Davide and Marra, Francesco and Poggi, Giovanni and Verdoliva, Luisa. (2021). Are GAN generated images easy to detect? A critical analysis of the state-of-the-art. arXiv.

[Gol17] Goldberg, Y. (2017). Neural Network Methods for Natural Language Processing. *Synthesis Lectures on Human Language Technologies*, 1-309.

[GPM+14] Goodfellow Ian, Pouget-Abadie Jean, Mirza Mehdi, Xu Bing, Warde-Farley David, Ozair Sherjil, Courville Aaron, Bengio Y. (2014). *Generative Adversarial Nets.* Montr´eal: arXiv.

[HLY+22] Hu, Kai and Liao, Wentong and Yang, Michael Ying and Rosenhahn, Bodo. (2022). *Text to Image Generation with Semantic-Spatial Aware GAN.* Ithaca, N: arXiv.

[HO06] Haibin Ling and Okada, K. (2006). Diffusion Distance for Histogram Comparison. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, (p. 246-253).

[HSS+21] Hossain, Md. Zakir and Sohel, Ferdous and Shiratuddin, Mohd Fairuz and Laga, Hamid and Bennamoun, Mohammed. (2021). Text to Image Synthesis for Improved Image Captioning. *IEEE Access*, 64918-64928.

[HTH+17] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, Dimitris N. Metaxas. (2017). StackGAN: Text to Photo-Realistic Image Synthesis With Stacked Generative Adversarial Networks. *IEEE International Conference on Computer Vision (ICCV)* (p. 5907-5915). Venice, Italy: IEEE.

[JYT+19] Joseph Harms,Yang Lei,Tonghe Wang,Rongxiao Zhang,Jun Zhou,Xiangyang Tang,Walter J. Curran,Tian Liu,Xiaofeng Yang. (2019). Paired cycle-GAN-based image correction for quantitative cone-beam computed tomography. *Medical Physics*, 3998-4009.

[KB14] Kingma, Diederik P. and Ba, Jimmy. (2014). *Adam: A Method for Stochastic Optimization.* arXiv.

[KGT01] Kosslyn S., Ganis G., Thompson W. (2001). Neural foundations of imagery. *Nature Reviews Neuroscience*, 635-642.

[KZZ+19] Kolchinski, Y. Alex and Zhou, Sharon and Zhao, Shengjia and Gordon, Mitchell and Ermon, Stefano. (2019). Approximating Human Judgment of Generated Image Quality. arXiv.

[LZZ+19] Li, Wenbo and Zhang, Pengchuan and Zhang, Lei and Huang, Qiuyuan and He, Xiaodong and Lyu, Siwei and Gao, Jianfeng. (2019). *Object-driven Text-to-Image Synthesis via Adversarial Training.* arXiv.

[MBG21] Mehmood, Rayeesa and Bashir, Rumaan and Giri, Kaiser J. (2021). Comparative Analysis of AttnGAN, DF-GAN and SSA-GAN. *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)* (p. 370-375). Greater Noida, India: IEEE .

[MHT+17] Martin Heusel and Hubert Ramsauer and Thomas Unterthiner and Bernhard Nessler and Gunter Klambauer and Sepp Hochreiter. (2017). GANs Trained by a Two Time-Scale Update Rule Converge to a Nash Equilibrium. *CoRR*.

[Mir21] Miranda, L. J. (2021). The Illustrated VQGAN. *ljvmiranda921.github.io*.

[PLF15] Peipei Xia and Li Zhang and Fanzhang Li. (2015). Learning similarity with cosine similarity ensemble. *Information Sciences*, 39-52.

[QCD+21] Qiao, Yanyuan and Chen, Qi and Deng, Chaorui and Ding, Ning and Qi, Yuankai and Tan, Mingkui and Ren, Xincheng and Wu, Qi. (2021). R-GAN: Exploring Human-like Way for Reasonable Text-to-Image Synthesis via Generative Adversarial Networks. *Proceedings of the 29th ACM International Conference on Multimedia* (p. 2085–2093). New York, NY, USA: Association for Computing Machinery.

[RZZ+21] Ruan, Shulan and Zhang, Yong and Zhang, Kun and Fan, Yanbo and Tang, Fan and Liu, Qi and Chen, Enhong. (2021). DAE-GAN: Dynamic Aspect-Aware GAN for Text-to-Image Synthesis. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (p. 13960-13969). Montreal, BC, Canada: ICCV.

[SGZ+16] Salimans, Tim and Goodfellow, Ian and Zaremba, Wojciech and Cheung, Vicki and Radford, Alec and Chen, Xi and Chen, Xi. (2016). Improved Techniques for Training GANs. *Advances in Neural Information Processing Systems.* Barcelona: Curran Associates, Inc.

[SR18] Shane, Barratt and Rishi, Sharma. (2018). A Note on the Inception Score.

[SS21] Sonal Ajay Bankara, Satish Ketb. (2021). An Analysis of Text-to-Image Synthesis. *Proceedings of the International Conference on Smart Data Intelligence (ICSMDI 2021)* (p. 7). Tamil Nadu, India: SSRN.

[STF+16] Stanislav Frolov, Tobias Hinz, Federico Raue, Jörn Hees, Andreas Dengel. (2016). Adversarial Text-to-Image Synthesis: A Review. *Proceedings of the 33 rd International Conference on Machine* (p. 10). New York, NY, USA: Proceedings mlr press.

[SWS+00] Smeulders, A.W.M. and Worring, M. and Santini, S. and Gupta, A. and Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1349-1380.

[SZX+16] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, Honglak Lee. (2016). Generative Adversarial Text to Image Synthesis. *Proceedings of Machine Learning Research* (p. 1060--1069). New York, New York, USA: PMLR.

[TTW+22] Tao, Ming and Tang, Hao and Wu, Fei and Jing, Xiao-Yuan and Bao, Bing-Kun and Xu, Changsheng. (2022). *DF-GAN: A Simple and Effective Baseline for Text-to-Image Synthesis.* arXiv.

[WLH+22] Wang, Zihao and Liu, Wei and He, Qian and Wu, Xinglong and Yi, Zili. (2022). *CLIP-GEN: Language-Free Training of a Text-to-Image Generator with CLIP.* arXiv.

[WWZ+20] Wang, Sheng-Yu and Wang, Oliver and Zhang, Richard and Owens, Andrew and Efros, Alexei A. (2020). CNN-generated images are surprisingly easy to spot... for now. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Vol. 7* (p. 8695-8704). IEEE Xplore.

[WYW+22] Wei Chen and Yu Liu and Weiping Wang and Erwin Bakker and Theodoros Georgiou and Paul Fieguth and Li Liu and Michael S. Lew. (2022). Deep Learning for Instance Retrieval: A Survey. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 7270-7292.

[XJK+16] Xinchen Yan, Jimei Yang, Kihyuk Sohn, Honglak Lee. (2016). *Attribute2Image: Conditional Image Generation from Visual Attributes.* Ithaca, NY: Cornell University.

[YDF19] Yu, Ning and Davis, Larry and Fritz, Mario. (2019). Attributing Fake Images to GANs: Learning and Analyzing GAN Fingerprints. *Proceedings of the IEEE/CVF International Conference on Computer Vision* (p. 7556–7566). IEEE Xplore.

[YM19] Yong-Hoon Kwon, Min-Gyu Park. (2019). Predicting Future Frames Using Retrospective Cycle GAN. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (p. 1811-1820). Long Beach, CA: CVPR.

[YTT+14] Yanwei Fu, Timothy M. Hospedales, Tao Xiang, Zhenyong Fu, Shaogang Gong. (2014). Transductive Multi-view Embedding for Zero-Shot Recognition and Annotation. *European Conference on Computer Vision* (p. 584-599). Switzerland: Springer, Cham.

[YYT+21] Ye, Hui and Yang, Xiulong and Takac, Martin and Sunderraman, Rajshekhar and Ji, Shihao. (2021). *Improving Text-to-Image Synthesis Using Contrastive Learning.* arXiv.

[YZD21] Yu, Yu and Zhang, Weibin and Deng, Yun. (2021). *Frechet Inception Distance (FID) for Evaluating GANs.* ResearchGate.

[ZBS+04] Zhou Wang and Bovik, A.C. and Sheikh, H.R. and Simoncelli, E.P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 600-612.

[Zel15] Harris, Z. S. (2015). Distributional Structure. *WORD*, 146-162.

[ZGA+19]Zhengwei Wang and Graham Healy and Alan F. Smeaton and Tomas E. Ward. (2019). Use of Neural Signals to Evaluate the Quality of Generative Adversarial Network Performance in Facial Image Generation. *Cognitive Computation*, 13-24.

[ZGK+19] Zhou, Sharon and Gordon, Mitchell L. and Krishna, Ranjay and Narcomey, Austin and Fei-Fei, Li and Bernstein, Michael S. (2019). HYPE: A Benchmark for Human eYe Perceptual Evaluation of Generative Models. arXiv.

[ZPC+19] Zhu, Minfeng and Pan, Pingbo and Chen, Wei and Yang, Yi. (2019). *DM-GAN: Dynamic Memory Generative Adversarial Networks for Text-to-Image Synthesis.* arXiv.

Priedas 1. Kokybės vertinimo metodo kodas

```
import tensorflow as tf
import numpy as np
import cv2
import requests
from bs4 import BeautifulSoup
from urllib.parse import quote_plus
import argparse
from selenium import webdriver
import requests
import json
import time
import os
import urllib.request
import random

def check_similarity(src_imgs, img):
    print("similarity")
    # Load the pre-trained model
    model = tf.keras.applications.VGG16(weights='imagenet', include_top=False,
input_shape=(224, 224, 3))
    # Load and preprocess the images
    if not (isinstance(src_imgs, list)):
        src_imgs = [src_imgs]

    similarities = []
    for src_img in src_imgs:
        img1 = cv2.imread(src_img)
        img1 = cv2.resize(img1, (224, 224))
        img1 = np.expand_dims(img1, axis=0)
        img1 = tf.keras.applications.vgg16.preprocess_input(img1)

        img2 = cv2.imread(img)
        img2 = cv2.resize(img2, (224, 224))
        img2 = np.expand_dims(img2, axis=0)
        img2 = tf.keras.applications.vgg16.preprocess_input(img2)

        # Get the features for both images
        features1 = model.predict(img1).flatten()
        features2 = model.predict(img2).flatten()

        # Calculate the similarity score using the cosine similarity metric
        features_consumption = np.dot(features1, features2) /
(np.linalg.norm(features1) * np.linalg.norm(features2))

        # Add the similarity score to the array
        similarities.append(features_consumption)
    similarities = max(similarities);
    if similarities > 0.5:
```

```
        return "Image is realistic"
    elif similarities > 0.35:
        return "Image should be realistic"
    else:
        return "Image is not realistic"

# Select images source for comparing
# if leaving empty will be bing by default
def image_to_compare(original_image = "", image_source = ""):
    comparison_images = []
    if(image_source):
        for x in range(0,5):
            random_image = random.choice(os.listdir(image_source))
            print(image_source+ "/"+random_image)
            comparison_images.append(image_source+ "/"+random_image)
        return comparison_images
    else:
        # Define the URL of the original image
        if(original_image):
            file_path =  original_image
        else:
            return False;
        file_path = upload_image(file_path);
        # Bing image search URL
        bing_url =
'https://www.bing.com/images/searchbyimage?FORM=IRSBIQ&cbir=sbi&imgurl=' + file_path

        # Send GET request to Bing search page
        browser = webdriver.Chrome()
        browser.get(bing_url)
        time.sleep(5)
        html = browser.page_source
        soup = BeautifulSoup(html, 'html.parser')
        a = soup.find_all('img', class_="nofocus")
        counter = 0
        # set the directory to save similar images to
        save_directory = "downlaod_img"
        if not os.path.exists(save_directory):
            os.makedirs(save_directory)
        for img in a:
            if file_path not in str(img):
                counter+=1
                # create the save directory if it doesn't exist
                resource = urllib.request.urlopen(img['src'] )
                original_name = original_image.split('/')
                if(len(original_name) <= 1):
                    original_name = original_image.split('\\')
                original_name = original_name[-1]
                original_name = original_name.split('.')
                original_name = original_name[0]
                if not os.path.exists(save_directory+"/"+original_name):
```

```
                    os.makedirs(save_directory+"/"+original_name)
                output =
open(save_directory+"/"+original_name+"/"+str(counter)+".jpg","wb")
                output.write(resource.read())
                output.close()
                comparison_images.append(save_directory+"/"+original_name+"/"+str(cou
nter)+".jpg")
            if(counter > 5):
                return comparison_images
        return False;

def upload_image(image_url):
    # Client id of imgur
    client_id = 'enter_your_id'
    headers = {'Authorization': 'Client-ID ' + client_id}
    # Upload image to Imgur
    with open(image_url, 'rb') as f:
        response = requests.post('https://api.imgur.com/3/image', headers=headers,
files={'image': f})

    # Parse response JSON to get image link
    if response.status_code == 200:
        json_data = json.loads(response.text)
        link = json_data['data']['link']
        return link
    else:
        print('Error uploading image to Imgur:', response.text)

parser = argparse.ArgumentParser()
parser.add_argument('--img', help='image which should be checked for similarity')
parser.add_argument('--src', help='source of images which should be used to be
compared with generated image')

args = parser.parse_args()
if args.img and args.src:
    images = image_to_compare(args.img, args.src);
    if(images):
        print(check_similarity(images, args.img))
    else:
        print("any similar image is not found (folder)")
elif args.img:
    images = image_to_compare(args.img);
    if(images):
        print(check_similarity(images, args.img))
    else:
        print("any similar image is not found (bing)")
else:
    print("Please provide image (--img) (mandatory) and source (--src) (optional)")
```