



Kauno technologijos universitetas

Informatikos fakultetas

Dirbtinio intelekto metodų taikymas grafinio turinio generavime

Baigiamasis magistro studijų projektas

Ovidijus Grigas

Projekto autorius

Dr. Algirdas Noreika

Vadovas

Kaunas, 2022



Kauno technologijos universitetas

Informatikos fakultetas

Dirbtinio intelekto metodų taikymas grafinio turinio generavime

Baigiamasis magistro studijų projektas
Programų sistemų inžinerija (6211BX011)

Ovidijus Grigas

Projekto autorius

Dr. Algirdas Noreika

Vadovas

Prof. Alfonsas Misevičius

Recenzentas

Kaunas, 2022



Kauno technologijos universitetas

Informatikos fakultetas

Ovidijus Grigas

Dirbtinio intelekto metodų taikymas grafinio turinio generavime

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;
2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąrašė;
3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);
4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Ovidijus Grigas

Patvirtinta elektroniniu būdu

Ovidijus Grigas. Dirbtinio intelekto metodų taikymas grafinio turinio generavime. Magistro studijų baigiamasis projektas / vadovas dr. Algirdas Noreika; Kauno technologijos universitetas, informatikos fakultetas.

Studijų kryptis ir sritis (studijų kryptių grupė): Programų sistemų inžinerija, Informatikos mokslai.

Reikšminiai žodžiai: dirbtinis intelektas, generatyviniai antagonistiniai neuroniniai tinklai, animaciniai personažai, atvaizdų sintezė.

Kaunas, 2022. 79 p.

Santrauka

Kurti animacinius personažus dažnai užtrunka mėnesius dėl įvairių priežasčių: (1) ilgai užtrukęs idėjų generavimas, (2) suinteresuoti asmenys nepatenkinti galutiniu produktu arba (3) užtruko teisiniai procesai.

Analizės metu buvo apžvelgiamos rinkoje naudojamos turinio generavimo technologijos, egzistuojantys sprendimai ir sugeneruotų atvaizdų kokybės vertinimo metrikos. Pastebėta, kad rinkoje tokių sprendimų trūksta.

Magistro projektas automatizuoja pradinį personažo kūrimo proceso etapą – idėjų generavimą, siūlo sprendimą, kuris animacinio personažo eskizą verčia į rastrinį atvaizdą.

Suformuotos tyrimo hipotezės: (a) generuojamų duomenų kokybė priklauso nuo duomenų rinkinio kokybės ir (b) neuroninio tinklo dydis daro tiesioginę įtaką generuojamų duomenų kokybei.

Siekiant patikrinti hipotezes buvo išmokyti aštuoni skirtingi neuroninių tinklų architektūrų modeliai su žemesnės ir aukštesnės kokybės duomenų rinkiniais.

Eksperimentai parodė, kad generuojamų duomenų kokybei didelę įtaką daro duomenų rinkinio kokybė, o neuroninio tinklo dydis nedaro jokios įtakos.

Ovidijus Grigas. Application of AI methods in Visual Content Generation. Master's Final Degree Project / supervisor Ph. D. Algirdas Noreika; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Software Engineering, Informatics.

Keywords: artificial intelligence, generative adversarial networks, cartoon character, image synthesis.

Kaunas, 2022. 79 p.

Summary

Creating cartoon characters is resource intensive and may usually draw out for months due to reasons such as: (1) prolonged generation of ideas, (2) stakeholders are not satisfied with the product or (3) legal procedures.

This Master's degree projects research phase sought to identify suitable quality metrics, evaluate existing solutions, and compared content generation technologies, but found a lack of solutions specific to the field of cartoon character generation.

This project investigated the use of generative adversarial neural networks to automate the initial generation of the cartoon character, by converting sketches into raster images, as a potential solution to the resource intensive nature governed by the generation of ideas.

The research hypotheses were that the quality of generated characters depends on (a) the quality of the input dataset and (b) by the size of the neural network.

To test these hypotheses, eight different generative adversarial neural networks were trained on, respectively, a lower and a higher quality dataset.

The experiments found a significant dependency on the quality of the dataset, but no distinguishable dependency on the size of the neural network.

Turinys

Lentelių sąrašas	8
Paveikslų sąrašas	9
Santrumpų ir terminų sąrašas	11
Įvadas.....	13
1. Analitinė dalis	14
1.1. Egzistuojančios turinio generavimo technologijos paremtos dirbtiniu intelektu	14
1.1.1. Variaciniai autokoduotuvai	14
1.1.2. Generatyviniai antagonistiniai neuroniniai tinklai	17
1.1.3. Technologijų palyginimas	21
1.2. Pasirinktas modelis ir programinės priemonės	22
1.2.1. Pasirinkto modelio pagrindimas	22
1.2.2. Programinės priemonės	22
1.3. Egzistuojantys sprendimai	22
1.4. Įgyvendinimo problemos	23
1.5. Kokybės vertinimo metrikos	23
1.5.1. Pradžios metrika	23
1.5.2. Frečeto pradžios atstumas.....	24
1.5.3. Branduolio pradžios atstumas.....	24
1.5.4. Išmoktas skirtingų vaizdo dalių panašumo suvokimas	25
1.5.5. Daugialypis struktūrinis panašumas	25
2. Reikalavimų specifikacija	27
2.1. Naudotojai	27
2.1.1. Animatoriai.....	27
2.1.2. Edukatoriai	27
2.1.3. Vaikai	27
2.2. Apribojimai.....	27
2.2.1. Apribojimai sprendimui.....	27
2.2.2. Diegimo aplinka	28
2.3. Veiklos sudėtis.....	28
2.4. Sistemos sudėtis.....	29
2.4.1. Sistemos ribos.....	29
2.4.2. Panaudojimo atvejų sąrašas	31
2.5. Funkciniai reikalavimai ir reikalavimai duomenims	34
2.5.1. Funkciniai reikalavimai	34
2.5.2. Reikalavimai duomenims	34
2.6. Nefunkciniai reikalavimai	36
2.6.1. Reikalavimai sistemos išvaizdai	36
2.6.2. Reikalavimai panaudojamumui	36
2.6.3. Reikalavimai veikimo sąlygoms.....	36
2.6.4. Reikalavimai sistemos priežiūrai	37
2.6.5. Reikalavimai saugumui	37
2.7. Sistemos pateikimo žingsniai	37
3. Architektūros specifikacija.....	38
3.1. Architektūros pateikimas	38

3.2. Architektūros tikslai ir apribojimai	38
3.3. Sistemos statinis vaizdas	38
3.3.1. Modulių detalizavimas	39
3.4. Sistemos dinaminis vaizdas	42
3.4.1. Būsenų diagramos	42
3.4.2. Veiklos diagramos	45
3.4.3. Sekų diagramos	49
3.5. Išdėstymo vaizdas.....	53
3.6. Duomenų vaizdas	53
4. Tyrimo dalis	54
4.1. Tyrimo aplinka	54
4.2. Uždaviniai.....	54
4.3. Funkcionalumo, logikos ir realizacijos vertinimas.....	54
4.3.1. Strategijos	54
4.3.2. Rezultatai.....	55
4.4. Sistemos atitikimas reikalavimų specifikacijai	56
4.4.1. Strategijos	56
4.4.2. Rezultatai	57
4.5. Generuojamų duomenų kokybės vertinimas	58
4.5.1. Strategijos	58
4.5.2. Rezultatai.....	58
4.6. Eksperimentiniams tyrimams siūlomi sprendimai	58
4.6.1. Duomenų rinkinys	58
4.6.2. Neuroninio tinklo architektūra	59
5. Eksperimento dalis	61
5.1. Hipotezės	61
5.2. Hipotezių tikrinimas	61
5.3. Eiga.....	61
5.3.1. Duomenų rinkinio sudarymas.....	61
5.3.2. Neuroninių tinklų mokymas	65
5.4. Tyrimo rezultatai	67
5.4.1. Kiekybinis vertinimas.....	67
5.4.2. Kokybinis vertinimas.....	74
5.5. Tyrimo išvada.....	78
Išvados	79
Literatūros sąrašas	80
Priedai.....	85
1 Priedas. Užpildytos apklausos anketos pavyzdys Nr. 1	85
2 Priedas. Užpildytos apklausos anketos pavyzdys Nr. 2	93
3 Priedas. Programų sistemos perdavimo ir aprobavimo aktas	96

Lentelių sąrašas

1 lentelė Analizuotų technologijų palyginimas.....	21
2 lentelė Veiklos kontekste vaizduojamų srautų paaiškinimas	28
3 lentelė Panaudojimo atvejo „Piešti personažo eskizą“ specifikacija	31
4 lentelė Panaudojimo atvejo „Sugeneruoti personažo atvaizdą“ specifikacija.....	32
5 lentelė Panaudojimo atvejo „Peržiūrėti sukurtus personažus“ specifikacija.....	33
6 lentelė Panaudojimo atvejo „Kurti siužetą“ specifikacija.....	33
7 lentelė Esybės „User“ požymių specifikacija	35
8 lentelė Esybės „Storyboard“ požymių specifikacija	35
9 lentelė Esybės „Character“ požymių specifikacija	36
10 lentelė Esybės „CharacterGenerationRecord“ požymių specifikacija	36
11 lentelė Tyrimo aplinkų konfigūracijos	54
12 lentelė ISO 25010 standarto parametrų įvertinimas.....	57
13 lentelė Esamo duomenų rinkinio sudarymo efektyvumo rezultatai.....	62
14 lentelė Naujojo duomenų rinkinio sudarymo efektyvumo rezultatai.....	65
15 lentelė Skaitinės analizės su senuoju duomenų rinkiniu rezultatai	67
16 lentelė Skaitinės analizės su naujuoju duomenų rinkiniu rezultatai	68
17 lentelė Statistinė koreliacija tarp neuroninio tinklo dydžio ir metrikų reikšmių (su naujuoju duomenų rinkiniu)	73
18 lentelė Statistinė koreliacija tarp neuroninio tinklo dydžio ir metrikų reikšmių (su senuoju duomenų rinkiniu)	73

Paveikslų sąrašas

1 pav. (Kairėje) spėjimo neuroninis tinklas – koduotuvas. (Dešinėje) generatoriaus neuroninis tinklas – iškoduotuvas [18]	15
2 pav. VAE architektūra [18].....	16
3 pav. GAN architektūra	17
4 pav. FCC-GAN generatoriaus architektūra [31]	18
5 pav. FCC-GAN diskriminatoriaus architektūra [31].....	18
6 pav. DCGAN generatoriaus architektūra [31]	20
7 pav. DCGAN diskriminatoriaus architektūra [31].....	20
8 pav. Diegimo aplinka	28
9 pav. Sistemos veiklos konteksto modelis.....	28
10 pav. Panaudojimo atvejų diagrama (1 dalis).....	29
11 pav. Panaudojimo atvejų diagrama (2 dalis).....	30
12 pav. Esybių ryšių modelis kuriamai sistemai.....	35
13 pav. Sistemos kūrimo procesas	37
14 pav. Sistemos paketų diagrama	39
15 pav. <i>Frontend</i> programinės įrangos dalies klasių diagrama	40
16 pav. <i>Backend</i> programinės įrangos dalies klasių diagrama.....	41
17 pav. Panaudojimo atvejo Nr. 4 būsenų diagrama.....	42
18 pav. Panaudojimo atvejo Nr. 6 būsenų diagrama.....	43
19 pav. Panaudojimo atvejo Nr. 7 būsenų diagrama.....	44
20 pav. Panaudojimo atvejo Nr. 9 būsenų diagrama.....	45
21 pav. Panaudojimo atvejo Nr. 4 veiklos diagrama	46
22 pav. Panaudojimo atvejo Nr. 6 veiklos diagrama	47
23 pav. Panaudojimo atvejo Nr. 7 veiklos diagrama	48
24 pav. Panaudojimo atvejo Nr. 9 veiklos diagrama	48
25 pav. Panaudojimo atvejo Nr. 4 sekų diagrama.....	49
26 pav. Panaudojimo atvejo Nr. 6 sekų diagrama.....	50
27 pav. Panaudojimo atvejo Nr. 7 sekų diagrama.....	51
28 pav. Panaudojimo atvejo Nr. 9 sekų diagrama.....	52
29 pav. Diegimo diagrama – fizinės architektūros schema.....	53
30 pav. Esamo duomenų rinkinio anotavimo pavyzdžiai	62
31 pav. Esamo duomenų rinkinio pavyzdys	63
32 pav. Naujojo duomenų rinkinio pavyzdys	64
33 pav. Naujojo duomenų rinkinio anotavimo pavyzdžiai	64
34 pav. <i>STYLEGAN</i> tinklo sugeneruoti animacinių personažų atvaizdai	67
35 pav. Frečeto pradžios atstumo metrikų palyginimas.....	69
36 pav. Pradžios metrikų palyginimas	70
37 pav. Branduolio pradžios atstumo metrikų palyginimas.....	70
38 pav. Daugialypio struktūrinio panašumo metrikų palyginimas	71
39 pav. Išmokto skirtingų vaizdo dalių suvokimo metrikų palyginimas	72
40 pav. Neuroninių tinklų svorių kiekis.....	73
41 pav. Personažų kokybės įvertinimo rezultatai (senasis duomenų rinkinys).....	74
42 pav. Sugeneruotų personažų atvaizdų pavyzdžiai (senasis duomenų rinkinys).....	75
43 pav. Personažų kokybės įvertinimo rezultatai (naujasis duomenų rinkinys).....	75

44 pav. Sugeneruotų personažų atvaizdų pavyzdžiai (naujasis duomenų rinkinys).....	76
45 pav. <i>FewShot-SMIS</i> tinklo mokymo metu naudojami duomenys	77
46 pav. <i>SuperStyleNet</i> sugeneruoti atvaizdai	78
47 pav. Pirmosios užpildytos apklausos pavyzdys 1 dalis.....	85
48 pav. Pirmosios užpildytos apklausos pavyzdys 2 dalis.....	86
49 pav. Pirmosios užpildytos apklausos pavyzdys 3 dalis.....	87
50 pav. Pirmosios užpildytos apklausos pavyzdys 4 dalis.....	88
51 pav. Pirmosios užpildytos apklausos pavyzdys 5 dalis.....	89
52 pav. Pirmosios užpildytos apklausos pavyzdys 6 dalis.....	90
53 pav. Pirmosios užpildytos apklausos pavyzdys 7 dalis.....	91
54 pav. Pirmosios užpildytos apklausos pavyzdys 8 dalis.....	92
55 pav. Pirmosios užpildytos apklausos pavyzdys 9 dalis.....	92
56 pav. Antrosios užpildytos apklausos pavyzdys 1 dalis	93
57 pav. Antrosios užpildytos apklausos pavyzdys 2 dalis	94
58 pav. Antrosios užpildytos apklausos pavyzdys 3 dalis	95
59 pav. Programų sistemos perdavimo ir aprobavimo aktas	96

Santrumpų ir terminų sąrašas

Santrumpos

2D – dvimatė erdvė;

3D – trimatė erdvė;

CGAN – sąlyginis generatyvinis antagonistinis neuroninis tinklas (angl. *Conditional Generative Adversarial Network*);

CPU – pagrindinis procesorius (angl. *Central Processing Unit*);

DBN – giliojo įsitikinimo tinklai (angl. *Deep Belief Networks*);

DCGAN – gilusis konvoliucinis generatyvinis antagonistinis neuroninis tinklas (angl. *Deep Convolutional Generative Adversarial Network*);

FCC-GAN – pilnai sujungtas ir konvoliucinis generatyvinis antagonistinis neuroninis tinklas (angl. *Fully Connected and Convolutional Generative Adversarial Network*);

FID – Frečeto pradžios atstumas (angl. *Fretchet Inception Distance*);

GAN – generatyvinis antagonistinis neuroninis tinklas (angl. *Generative Adversarial Network*);

GMM – Gauso maišos modelis (angl. *Gaussian Mixture Model*);

GPU – grafinis procesorius (angl. *Graphics Processing Unit*);

HMM – paslėptasis Markovo modelis (angl. *Hidden Markov Model*);

IS – pradžios metrika (angl. *Inception Score*);

KID – branduolio pradžios atstumas (angl. *Kernel Inception Distance*);

LDA – fiktyvusis Dirichlė skirstinys (angl. *Latent Dirichlet Allocation*);

LPIPS – išmoktas skirtingų vaizdo dalių panašumo suvokimas (angl. *Learned Perceptual Image Patch Similarity*);

LRELU – praleidžianti detektorinė linijinė aktyvacijos funkcija (angl. *Leaky Rectified Linear Unit*);

MSSSIM – daugialypis struktūrinis panašumas (angl. *Multi Scale Structural Similarity*);

PA – panaudojimo atvejis;

RELU – detektorinė linijinė aktyvacijos funkcija (angl. *Rectified Linear Unit*);

RGB – spalvų kodavimo erdvė (angl. *Red Green Blue*);

UML – vieninga modeliavimo kalba (angl. *Unified Modeling Language*);

VAE – variacinis autokoduotuvus (angl. *Variational Auto-Encoder*);

WEB – internetas.

Terminai

Atbulinis sklidimas – metodas leidžiantis atnaujinti neuroninio tinklo svorius ankstesniuose sluoksniuose tam, kad būtų atliekamas tinklo išmokymas (angl. *back propagation*);

Gradientinis nusileidimas – mašininio mokymo algoritmas (angl. *gradient descend*);

Nešo pusiausvyra – kai nei vienas žaidėjas iš dviejų nepadidins savo šansų laimėti, pakeitęs savo sprendimą, todėl reikia abiejų žaidėjų bendro pasirinkimo (angl. *Nash equilibrium*);

Neuroninis tinklas – tarpusavyje sujungtų dirbtinių neuronų grupė, kuri mėgdžioja žmogaus smegenų darbą;

Nuostolio funkcija – optimizavimo uždavinys neuroninių tinklų mokyme (angl. *loss function*);

Savybė – neuroninio tinklo išmokstama objekto savybė (angl. *feature*);

Semantinis objektas – semantinis objektas yra personažo kūno dalis (ranka, koja ir t.t.), kuris yra pavaizduotas semantiniame objektų žemėlapyje;

Semantinis objektų žemėlapis – eskizas/atvaizdas, kuriame kiekvienas semantinis objektas pažymimas objektui būdinga spalva.

Įvadas

Dirbtinis intelektas yra informacinių technologijų sritis, kuri turi platų panaudojimų spektrą. Vykdoma daug įvairių tyrimų, kurių tikslas yra automatizuoti ar optimizuoti kurią nors veiklą, pavyzdžiui, versti tekstą, nuspėti orus ar generuoti turinį pasitelkiant srities žinių bazę. Visi tyrimai dažnai orientuoti į aktualias pasaulio, verslo problemas ir siūlo sprendimus, įrankius, padedančius atnaujinti naudojamus procesus. Generuojant turinį dirbtinis intelektas naudojamas tokiose pagrindinėse srityse: vaizdas, tekstas ir garsas. Šiame magistro baigiamajame projekte buvo koncentruotasi į vaizdų generavimą, siūlomas sprendimas aktualus verslui, kuris buvo kuriamas pagal realų įmonės užsakymą, o atliktas tyrimas palygina skirtingų metodų įtaką generuojamų vaizdų kokybei.

Problema

Animacinių personažų kūrimas yra brangus ir kruopštus darbas, kurį gali atlikti ne kiekvienas. Animatoriai dažnai praleidžia mėnesius bandydami skirtingas idėjas, modeliudami personažus, kurdami jų charakterio bruožus, o jei dar užsakovas nebūna patenkintas gautu rezultatu – tenka iš naujo piešti, modeliuoti, koreguoti įvairias personažo išvaizdos detales, kol pasiekiamas rezultatas, kuris tenkina visus projektu suinteresuotus asmenis. Personažo kūrime didelę dalį laiko užima idėjų generavimas. Kuriant personažą skirtingų tekstūrų ar modelių išbandymas reikalauja daug laiko ir išteklių, todėl šį procesą palengvinantis sprendimas galėtų sutrumpinti idėjų generavimo proceso trukmę. Tokių sprendimų rinkoje nėra daug, nes taikymo sritis menkai iširta, todėl atsiranda poreikis tokį sprendimą sukurti.

Tikslas – iširti neuroninių tinklų duomenų generavimo kokybę, pasitelkiant animacinių personažų rastrinius atvaizdus.

Uždaviniai:

1. išanalizuoti egzistuojančius animacinių personažų generavimo sprendimus, naudojamas technologijas ir generatyvinių neuroninių tinklų architektūras;
2. apibrėžti kuriamos sistemos reikalavimus;
3. suprojektuoti animacinių personažų generavimo sistemą;
4. sukurti animacinių personažų generavimo sistemą;
5. atlikti neuroninių tinklų tyrimus.

Darbo struktūra:

- analitinėje dalyje analizuojama pasirinkta mokslinė projekto sritis, aiškinamasi, kokias projekte naudoti technologijas, kaip įvertinti eksperimento metu įgyvendintus patobulinimus;
- reikalavimų specifikacijos dalyje analizuojami užsakovo reikalavimai sistemos funkcionalumui sukurti;
- architektūros specifikacijos dalyje aptariami esminiai sistemos projektiniai sprendimai;
- tyrimo dalyje įvertinamas techninio projekto metu sukurtos sistemos atitikimas reikalavimų specifikacijai, siūlomi sprendimai galintys patobulinti sistemą, suformuojamos tyrimo hipotezės;
- eksperimento dalyje pateikiami eksperimento metu gauti rezultatai;
- išvadose apibendrinami projekto metu gauti rezultatai ir pateikiamos išvalgos.

1. Analitinė dalis

Skyriaus paskirtis – išanalizuoti rinkoje naudojamus vaizdo turinio generavimo, pasirinkti įrankius ir technologijas projektui įgyvendinti, pasiruošti tyrimui pasirenkant kiekybinio vertinimo metrikas.

1.1. Egzistuojančios turinio generavimo technologijos paremtos dirbtiniu intelektu

Dirbtinio intelekto naudojimas generuojant turinį yra viena iš kryptių, kuri šiuo metu sulaukia gana didelio dėmesio. Galimybė generuoti turinį suteikia menininkams meno interpretacijų pavyzdžių, o žmonėms, kurie neturi gebėjimų kurti, galimybę pasijusti menininkais. Internete galime rasti įvairių straipsnių, kurie pritaiko dirbtinį intelektą muzikos, garsų [1, 2, 3], teksto [4, 5], vaizdo įrašų [6, 7], teksto vertimo į nuotraukas [8, 9] ar atvaizdų [10, 11, 12, 13, 14, 15, 16, 17] generavime. Duomenims generuoti naudojami tokie generatyviniai modeliai [18]: Gauso maišos modelis (angl. *Gaussian Mixture Model* – GMM), paslėptasis Markovo modelis (angl. *Hidden Markov Model* – HMM), fiktyvusis Dirichlė skirstinys (angl. *Latent Dirichlet Allocation* – LDA), Bolcmano mašinos (angl. *Boltzmann machines*), giliojo įsitikinimo tinklai (angl. *Deep Belief Networks* – DBN), tačiau pastarieji pagal minėtąjį straipsnį nėra praktiškai pritaikomi generuojant vaizdus. Šiam uždaviniui tinkantys modeliai: variaciniai autokoduotuvai (angl. *Variational Auto-Encoder* – VAE) ir generatyviniai antagonistiniai neuroniniai tinklai (angl. *Generative Adversarial Networks* – GAN). Metodai įprastą dirbtinio intelekto paskirtį atlikti klasifikavimą, aptikimą ar prognozavimą pritaiko visiškai kitam uždaviniui – turiniui generuoti. Vietoje duomenų panaudojimo analizei, sukauptą žinių bazę naudojame tokios pačios rūšies turiniui generuoti.

1.1.1. Variaciniai autokoduotuvai

Variaciniai autokoduotuvai [19] yra modeliai, kurie naudoja išmoktą aproksimacijos funkciją naujų duomenų generavimui. Šie modeliai dažnai išmokomi su gradientiniu nusileidimu (angl. *gradient descend*). Įprastai autokoduotuvai susideda iš dviejų tinklų: koduotuvo ir iškoduotuvo. Koduotuvo tinklas įvestyje gauta duomenų vektorių užkoduoja, kurį iškoduotuvai iškoduoja. Autokoduotuvai yra modeliai, kurie naudoja neprižiūrimąjį mokymąsi. Procesas vyksta taip: duomenys, patekę į įvesties sluoksnį yra užkoduojami vektoriuje Z kartu su svoriais. Kodavimo funkcija atrodo taip:

$$Z = f(WX + b), \quad (1.1)$$

kur b – yra poslinkis, o W – svoriai. Nuostolių funkcija (angl. *loss function*) naudojama autokoduotuvuose yra paprasta kvadratinų nuokrypių suma (L2 nuostolio funkcija):

$$L2 = \sum_{i=1}^N (y - \hat{y})^2, \quad (1.2)$$

kur \hat{y} – užkoduotoji reikšmė arba išvesties reikšmė. Suskaičiuota nuostolio funkcija naudojama atbuliniame sklidime (angl. *back propagation*), kurio metu atnaujinami svoriai ir optimizuojamas kodavimo algoritmas. Įprastiniai autokoduotuvai tinka triukšmo slopinimui atvaizduose, tačiau netinka naujų duomenų generavime dėl išmoktų duomenų atkartojimo. Norint generuoti duomenis, reikia modifikuoti įprastinį autokoduotuvą, kad jis galėtų išrinkti duomenis iš duomenų erdvės. Tarkime, kad duomenys, kuriuos turime $\{x^i\}_{i=1}^N$ yra sugeneruoti $p_{\theta}(z)$ pagal pasiskirstymo funkciją z (Gauso ar Bernulio), kur θ – modelio parametrai. Atliekant duomenų generavimą išrenkamas

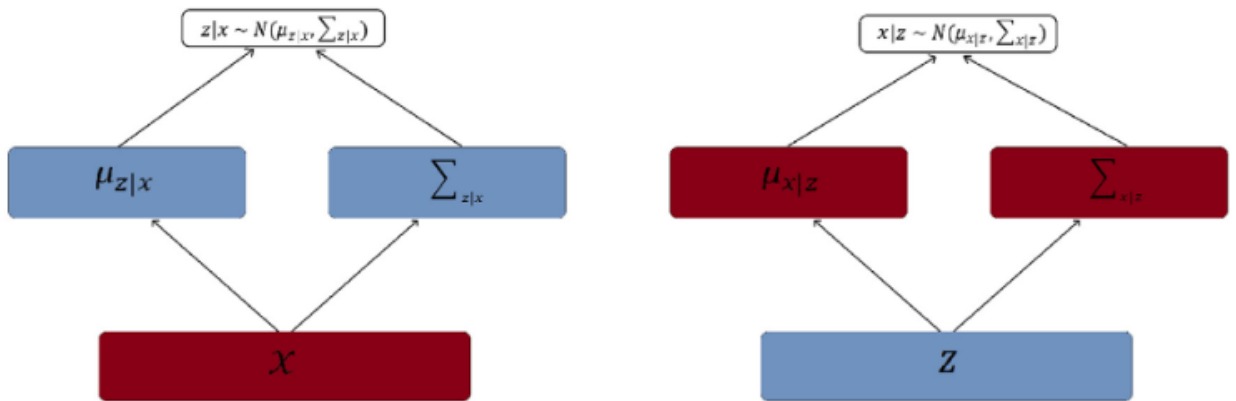
duomenų elementas x iš žinių bazės pagal tikimybę $p_\theta(X|z^i)$ ir nustatomi modelio parametrai θ . Generatyviniai modeliai stengiasi maksimizuoti mokymo duomenų tikimybinę išraišką:

$$p_\theta(x) = \int p_\theta(z)p_\theta(x|z)dz, \quad (1.3)$$

tačiau $p_\theta(x|z)$ negalime integruoti, o tikimybė:

$$p_\theta(z|x) = \frac{p_\theta(x|z)p_\theta(z)}{p_\theta(x)} \quad (1.4)$$

taip pat neintegruojama. Išspręsti šią problemą galime panaudoję koduotuvo neuroninį tinklą $q_\varphi(z|x)$ ir suskaičiavę apytikslę $p_\theta(z|x)$ reikšmę.



1 pav. (Kairėje) spėjimo neuroninis tinklas – koduotuvus. (Dešinėje) generatoriaus neuroninis tinklas – iškoduotuvus [18]

1 pav. matome du neuroninius tinklus: koduotuvus $q_\varphi(z|x)$ ir iškoduotuvus $p_\theta(x|z)$, kurie išvestyje gražina vidurkį ir kovariacijos dydžius. Mokymosi duomenų tikimybinę išraišką galime išreikšti logaritmine galimybe – E , atsižvelgiant į z koduotojo modelį $q_\varphi(z|x^i)$.

$$\log p_\theta(x^i) = E_{z \sim q_\varphi(z|x^i)} [\log p_\theta(x^i)], \quad (1.5)$$

kur $p_\theta(x^i)$ yra nepriklausomas nuo z

$$\log p_\theta(x^i) = E_z \left[\log \frac{p_\theta(x^i|z)p_\theta(z)}{p_\theta(z|x^i)} \right], \quad (1.6)$$

pritaikius Bajeso daugybės funkciją gauname

$$\log p_\theta(x^i) = E_z \left[\log \frac{p_\theta(x^i|z)p_\theta(z) q_\varphi(z|x^i)}{p_\theta(z|x^i) q_\varphi(z|x^i)} \right], \quad (1.7)$$

$$\log p_\theta(x^i) = E_z [\log p_\theta(x^i|z)] - E_z \left[\log \frac{q_\varphi(z|x^i)}{p_\theta(z)} \right] + E_z \left[\log \frac{q_\varphi(z|x^i)}{p_\theta(z|x^i)} \right], \quad (1.8)$$

$$\log p_{\theta}(x^i) = E_z[\log p_{\theta}(x^i|z)] - KL(q_{\varphi}(z|x^i)||p_{\theta}(z)) + KL(q_{\varphi}(z|x^i)||p_{\theta}(z|x^i)) \quad (1.9)$$

Pagal [18, 19] šaltinius įvedamas Kulbako-Leiblerio divergencijos (angl. *Kullback-Leibler divergence*) matas, kuris apibūdina, kaip vienos tikimybės pasiskirstymas skiriasi nuo kitos. Pagal 1.9 formulę matome, kad trečiasis dėmuo apibrėžia tikimybę $p_{\theta}(x^i|z)$, kuri žinome, jog yra neintegruojama. Naudodami [19] straipsnyje aprašytą parametrų apkeitimą, galime antrąjį dėmenį pakeisti dvejomis Gauso pasiskirstymo funkcijomis, o jų Kulbako-Leiblerio divergencijos matas leidžia suformuluoti galutinę formulę. Taigi, norėdami maksimizuoti duomenų išrinkimo iš mokymosi žinių bazės tikimybę, mums reikia maksimizuoti pirmąjį dėmenį ir minimizuoti antrąjį dėmenį. Kadangi trečiasis dėmuo yra visada teigiamas [19], galime suformuluoti tokią optimizavimo funkciją:

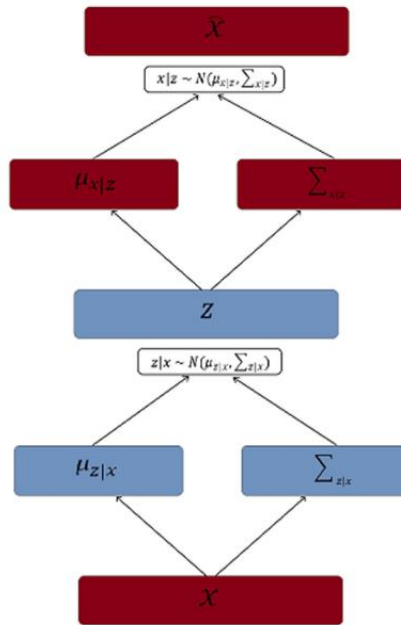
$$\varepsilon(x^i, \theta, \varphi) = E_z[\log p_{\theta}(x^i|z)] - KL(q_{\varphi}(z|x^i)||p_{\theta}(z)), \quad (1.10)$$

sulyginus formules gauname tokį kintantį (variacinį) apatinį rėžį:

$$\log p_{\theta}(x^i) \geq \varepsilon(x^i, \theta, \varphi). \quad (1.11)$$

Atliekant variacinio autokoduotuvo mokymą, ieškome θ, φ išvestinių θ', φ' maksimizuodami $\varepsilon(x^i, \theta, \varphi)$:

$$\theta', \varphi' = \arg \max_{\theta, \varphi} \sum_{i=1}^N \varepsilon(x^i, \theta, \varphi) \quad (1.12)$$

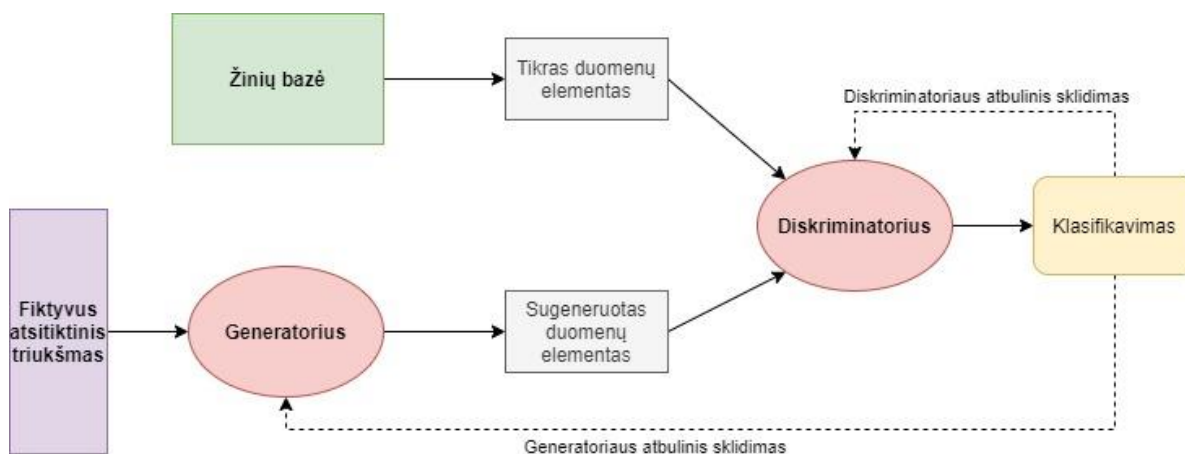


2 pav. VAE architektūra [18]

Modelio architektūra pavaizduota 2 pav. Tokie modeliai dažnai naudojami rekomendacijų sistemose [20], teksto klasifikavime [21], kalbos sintezėje [22], ar atvaizdų generavime [23]. Didžiausias metodo trūkumas – sugeneruojami atvaizdai turi daug triukšmo ir yra neryškūs [18, 24]. Tačiau variaciniai autokoduotuvai apmokomi greičiau, nei generatyviniai antagonistiniai neuroniniai tinklai.

1.1.2. Generatyviniai antagonistiniai neuroniniai tinklai

Ši neuroninių tinklų rūšis geba atlikti aukštos kokybės, sofistikuotą duomenų generavimą. Metodo principas yra toks, jog mokymosi procese naudojami du neuroniniai tinklai: generatorius (angl. *generator*) ir diskriminatorius (angl. *discriminator*), kurie tarpusavyje rungtyniauja. Rungtynių procesas pasižymi tuo, jog generatorius stengiasi sugeneruoti tokius duomenis, kurių diskriminatorius nesugebėtų atskirti tikrųjų nuo netikrųjų. Kitais žodžiais, stengiamasi maksimizuoti diskriminatoriaus tikimybę suklysti klasifikuojant sugeneruotus duomenis į tikruosius (angl. *ground truth*) ir ne. Šis rungtyniavimas prilygsta mini-max algoritmo pagrindu grindžiamam žaidimui. Apibendrintas proceso vaizdas pateikiamas 3 pav.



3 pav. GAN architektūra

Šio tipo neuroniniai tinklai buvo išrasti 2014 metais [25] Ian'o J. Goodfellow'o, kurie leido atsirasti daugybei išvestinių metodų, pritaikytų įvairiomis duomenų generavimo kryptimis: nuotraukos stiliaus savybių pritaikymas kitoms nuotraukoms [15], segmentacijos kaukių naudojimas nuotraukos rekonstrukcijoje [10, 17, 26, 27, 28, 29], nuotraukų dalių naudojimas jų rekonstrukcijoje [16], nuotraukų sintezė perduodant ryškiausias savybes [14]. Bendru atveju, GAN mokymo procesas siekia optimizuoti šią lygtį:

$$\min_G \max_D V(D, G) = E_x(\log D(x)) + E_z(\log 1 - D(G(z))), \quad (1.13)$$

kur z – atsitiktinis triukšmas, x – tikri duomenys, D – diskriminatoriaus diferencijuojama funkcija, G – generatoriaus diferencijuojama funkcija, E_x – tikėtina visų tikrų duomenų reikšmė, E_z – tikėtina visų netikrų duomenų reikšmė.

Diskriminatorius siekia maksimizuoti savo užduotį su tikrais duomenimis $D(x) \approx 1$ ir minimizuoti generatoriaus sugeneruotą užduoties išvestį $D(G(z)) \approx 0$. Taigi, diskriminatorius siekia kuo geriau atskirti tikrus duomenis nuo sugeneruotų, o generatorius sugeneruoti duomenis artimus 1, kad apgautų diskriminatorių. Šie neuroniniai tinklai skiriasi nuo įprastų tuo, kad jie nekonverguoja (mokymas vykdomas tol, kol pasiekiamas tikėtinas „globalus“ minimumas), bet jų mokymas nutraukiamas, kai pasiekiamas balno (angl. *saddle*) taškas [18]. Tai taškas, kai gradientinio nusileidimo metu atnaujinant neuroninio tinklo svorius, optimizuojamos funkcijos reikšmė nebemažėja, tačiau vis dar nepasiekė globalaus minimumo. Žaidimų teorijoje šis rezultatas

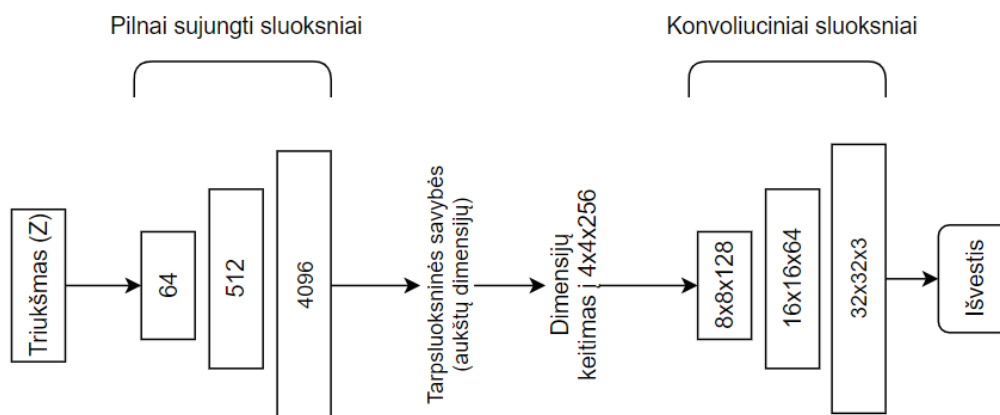
vadinamas Nešo pusiausvyra (angl. *Nash equilibrium*), kai nei vienas žaidėjas nepadidins savo šansų laimėti pakeitęs savo sprendimą, todėl reikia abiejų žaidėjų bendro pasirinkimo.

Šaltiniuose [18, 30] išskiriami pagrindiniai GAN tipai:

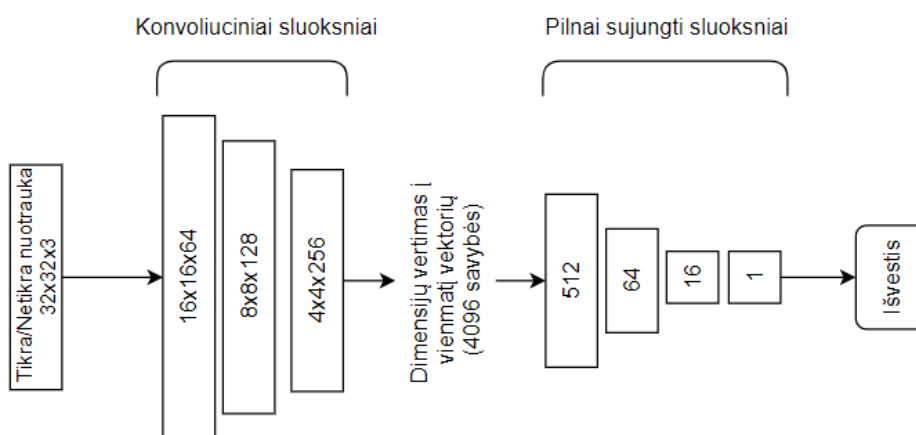
- pilnai sujungtas ir konvoliucinis GAN (angl. *Fully Connected and Convolutional Generative Adversarial Network* – FCC-GAN);
- gilusis konvoliucinis GAN (angl. *Deep Convolutional Generative Adversarial Network* – DCGAN);
- sąlyginis GAN (angl. *Conditional Generative Adversarial Network* – CGAN).

1.1.2.1. Pilnai sujungtas ir konvoliucinis GAN

Šio tipo GAN [31] patobulina įprastinius giliuosius konvoliucinius generatyvinius antagonistinius neuroninius tinklus, naudodami pilnai sujungtus neuronų sluoksnius.



4 pav. FCC-GAN generatoriaus architektūra [31]



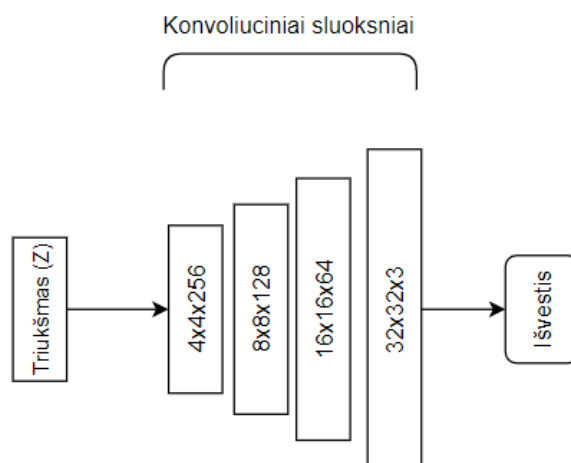
5 pav. FCC-GAN diskriminatoriaus architektūra [31]

4 pav. ir 5 pav. vaizduoja generatoriaus ir diskriminatoriaus architektūras. Generatorius kaip ir kituose GAN, įvestyje priima atsitiktinio triukšmo vektorių Z . Po įvesties sluoksnio seka 3 pilnai sujungti sluoksniai, kurie padidina duomenų dimensionalumą (savybių skaičių). Padidinus savybių skaičių, jos praleidžiamos per konvoliucinius sluoksnius, kad išvestyje gautume sugeneruotą 32 prie 32 prie

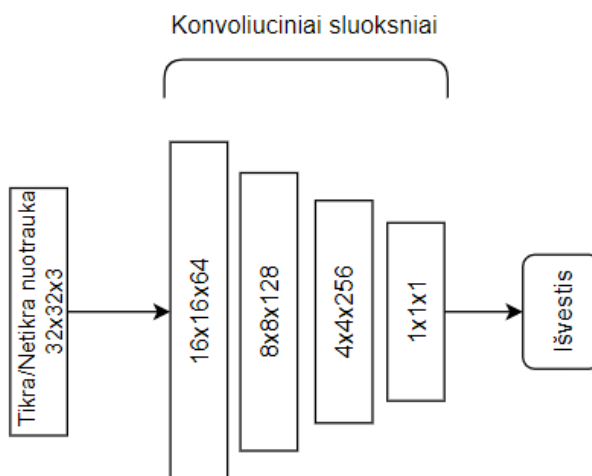
3 dydžio atvaizdą. Diskriminatorius savo įvestyje priima atvaizdą, todėl viskas vyksta atvirkščiai: priekyje išdėstyti konvoliuciniai sluoksniai, kurie didina duomenų savybių skaičių, tuomet pilnai sujungti sluoksniai, kurie jį mažina. Pabaigoje atliekamas klasifikavimas. GAN tipo modelių mokymas trunka ilgiau ir nėra stabilus. Naudojant tinkamus hiperparametrus ir mokymo technikas [32, 33], mokymo procesas gali būti stabilesnis.

1.1.2.2. Gilusis konvoliucinis GAN

Gilieji konvoliuciniai generatyviniai antagonistiniai neuroniniai tinklai buvo antroji šio tipo neuroninių tinklų iteracija (patys pirmieji turėjo tik pilnai sujungtus sluoksnius [25]). Žinant, kad konvoliuciniai neuroniniai tinklai geba efektyviai apdoroti atvaizdų duomenis, buvo nuspręsta konvoliucinius tinklus pritaikyti GAN tipo architektūrai. Tačiau pirmieji eksperimentai rodė, kad tokių tinklų mokymas yra sudėtingas, nes tradiciniai GAN gali po kelių mokymo epochų gali subyrėti (angl. *collapse*) [18], o generatorius pradeda generuoti visiškai neatpažįstamus atvaizdus. Todėl buvo pradėta ieškoti būdų kaip šias problemas išspręsti. Vienas iš jų buvo naudoti Laplaso piramidę realiems duomenims [30, 34], kitas [35] – naudoti du konvoliucinius neuroninius tinklus tiek generatoriui, tiek diskriminatoriui. Šis sprendimas leidžia išnaudoti žingsnines (angl. *strided*) ir iš dalies žingsnines konvoliucijas (angl. *partially-strided*), kad mokymo metu būtų išmoktos efektyvios erdvės dimensionalumo didinimo (angl. *spacial up-sampling*) ir mažinimo funkcijos (angl. *spacial down-sampling*). 6 pav. ir 7 pav. vaizduoja šių tinklų diskriminatoriaus ir generatoriaus architektūras. Kaip matome, generatoriuje triukšmas perduodamas į konvoliucinius sluoksnius, o diskriminatoriuje atvaizdas perduodamas į konvoliucinius sluoksnius. Šie tinklai nuo tradicinių skiriasi ir tuo, kad juose normalizuojami duomenys.



6 pav. DCGAN generatoriaus architektūra [31]



7 pav. DCGAN diskriminatoriaus architektūra [31]

Normalizavimas – funkcija, kuri normalizuoja duomenis, tai yra, turėtų nulinį vidurkį ir vienetinę dispersiją. Tai padeda pagerinti mokymo procesą ir, tikėtina, išvengti tinklo subyrėjimo. Šioje architektūroje naudojamos detektorinės linijinės (angl. *Rectified Linear Unit* – RELU) aktyvacijos funkcijos [36] generatoriui ir praleidžiančios detektorinės linijinės (angl. *Leaky Rectified Linear Unit* – LRELU) aktyvavimo funkcijos diskriminatoriui [37]. Tai leidžia pasiekti aukštesnės raiškos modeliavimą, nes LRELU praleidžia neuronus neigiamoje zonoje, kai RELU praleidžia tik teigiamas reikšmes. Aktyvavimo funkcijų formulės pateikiamos žemiau:

RELU neurono aktyvavimo funkcija:

$$f(x) = \max(0, x) \quad (1.14)$$

LRELU neurono aktyvavimo funkcija:

$$f(x) = \begin{cases} x & \text{jei } x > 0, \\ ax & \text{jei } x \leq 0 \end{cases} \quad (1.15)$$

Kaip matome, LRELU aktyvavimo funkcija, kai neuronas turi teigiamą reikšmę ją praleidžia, o kai neigiamą, tuomet priklauso nuo hiperparametro a .

Šio tipo neuroniniai tinklai dažniausiai naudojami atliekant atvaizdų sintezės sprendimuose. FCGAN tipo neuroniniai tinklai dar tik skinasi kelių rinkoje ir nėra plačiai naudojami.

1.1.2.3. Sąlyginis GAN

Šio tipo neuroniniai tinklai [38] leidžia kontroliuoti duomenų generavimą įvedus papildomą kintamąjį į GAN architektūros funkciją:

$$\min_G \max_D V(D, G) = E_x(\log D(x|y)) + E_z(\log 1 - D(G(z|y))) \quad (1.16)$$

kur y – papildoma informacija, pavyzdžiui, grupių pavadinimai į kurias patenka generuojami duomenys. Šio tipo neuroniniai tinklai naudojami tokiose srityse: žmonių veidų generavimas [39], atvaizdo į atvaizdą konvertavimas [10], senėjimo proceso žmonių veiduose generavimas [40], stiliaus ir kitų savybių iš kelių referencinių atvaizdų generuojamam atvaizdui perdavimas [41].

1.1.3. Technologijų palyginimas

Lyginant metodus buvo pastebėta, kad visi jie turi privalumų ir trūkumų. Technologijos buvo lyginamos pagal tokius kriterijus:

- aukštos kokybės atvaizdai – aukšta sugeneruotų animacinių personažų atvaizdų kokybė yra privaloma, kad animatoriai suprastų matomą vaizdą (vaizdas turi būti nesuliejęs, be didelio triukšmo);
- mokymosi stabilumas – ar mokymo procesas stabilus (GAN tipo tinklai yra ne tokie stabilūs);
- duomenų išrinkimas generavimui – generuojamų duomenų išrinkimas iš fiktyvios (triukšmo) erdvės, paprastumas (ar reikia papildomų priemonių);
- mokymo trukmė – svarbu vykdant eksperimentus dėl laiko apribojimų.

1 lentelė Analizuotų technologijų palyginimas

Metodas/Kriterijai	Aukštos kokybės nuotraukos	Mokymosi stabilumas	Duomenų išrinkimas generavimui	Mokymo trukmė
FCC-GAN	+	Nestabilus	Paprastas	Ilgesnė
DCGAN	+	Nestabilus	Paprastas	Ilgesnė
CGAN	+	Nestabilus	Paprastas	Ilgesnė
VAE	-	Stabilus	Sudėtingesnis	Trumpesnė

1 lentelėje matome, kad GAN modeliai pasižymi aukštos kokybės atvaizdų generavimu, tačiau jų mokymo procesas nestabilus ir trunka ilgiau, bet juos paprasčiau realizuoti. Taip pat GAN neuroniniai tinklai plačiai naudojami rinkoje ir randama daug egzistuojančių sprendimų (neuroninių tinklų architektūrų) pavyzdžių.

CGAN šio darbo uždaviniams pasiekti netinka dėl savo taikymo srities. DCGAN ir FCC-GAN tinklai yra tinkami, tačiau DCGAN yra plačiau naudojami ir apie juos randama daugiau informacijos bei skirtingų įgyvendinimo variantų internete, o tai palengvina realizacijos ir tyrimų procesą.

1.2. Pasirinktas modelis ir programinės priemonės

1.2.1. Pasirinkto modelio pagrindimas

Atlikus srities apžvalgą buvo nuspręsta naudoti DCGAN tipo neuroninius tinklus, nes jie pasižymi aukšta generuojamų duomenų kokybe (aukštesne nei VAE) ir yra plačiai naudojami rinkoje. Pritaikius diskriminatoriaus augmentavimo metodą galime išmokyti su reliatyviai nedideliu duomenų kiekiu [42], o su gerai parinktais hiperparametrais išmokyti stabiliai [32, 33, 43].

1.2.2. Programinės priemonės

Norint įgyvendinti pasirinktą metodą, reikia apžvelgti giliojo mokymo bibliotekas, kurios turi įgyvendintus reikalingus įrankius ir algoritmus. Lyginant bibliotekas svarbu:

- mokymas naudojant grafinį procesorių (angl. *Graphics Processing Unit* – GPU) – svarbu, nes mokymo procesas yra greitesnis dėl optimizuotų lygiagrečių skaičiavimo operacijų (angl. *Central Processing Unit*);
- derinimo priemonės – jų būvimas leidžia greičiau aptikti klaidas ir trūkumus neuroninio tinklo architektūroje ar mokymo procese.

Pagal [44] šaltinį išskiriamos tokios populiariausios giliojo mokymosi bibliotekos:

1. Tensorflow;
2. Keras;
3. PyTorch.

Tensorflow yra sukurta *Google* kompanijos ir yra plačiai naudojama eksperimentinėje ir produkcinėje veikloje. *Keras* yra aukštesnio nei *Tensorflow* lygio biblioteka dėl paprastesnės konfigūracijos ir yra orientuota į prototipavimą. Į *Tensorflow* nuo 2.0 versijos buvo įdiegtas *Keras* įskiepis, kuris giliojo mokymo biblioteką padarė patrauklesne rinkoje.

PyTorch biblioteka yra sukurta *Facebook* ir suteikia visas funkcijas, kurias siūlo *Tensorflow*, tačiau esminis skirtumas tarp *PyTorch* ir *Tensorflow* – aukšto lygio sąsaja (kaip ir *Keras*). Tai leidžia sukurti neuroninio tinklo modelį paprasčiau. *PyTorch* kaip ir *Tensorflow*, leidžia daug funkcijų atlikti žemo lygio programavime, jei naudotojas pageidauja. *Tensorflow* ir *PyTorch* turi derinimo įrankių *Tensorboard* įskiepi, kuriame yra daug pagalbinių priemonių (grafų, statistinių duomenų atvaizdavimas ir t.t.). Priemonės leidžia efektyviai dirbti ir ieškoti klaidų realizacijoje. Visi šie giliojo mokymo moduliai palaiko mokymą su GPU.

Projekto ir tyrimo realizacijai bus naudojamas *PyTorch* modulis dėl paprasto konfigūracijos proceso, į prototipavimą orientuotos sąsajos ir prieigos prie *Tensorboard* įskiepio.

1.3. Egzistuojantys sprendimai

Rinkoje panašių sistemų į kurią, kurios generuoja atvaizdus apstu [10, 11, 14, 39, 40], tačiau generuojančių animacinių personažų atvaizdus vos keli [45, 46, 47], visi jie daugiausia naudoja DCGAN tipo tinklus. Kaip buvo kuriamas GAN skirtas japoniškos *anime* veikėjų generavimui,

aprašoma [48] straipsnyje. Autoriai sukomponavo DCGAN su CGAN, o tam naudojo [49] straipsnyje aprašytas metodikas. Tai leido generuoti *anime* veikėjus pagal tam tikras savybes, pavyzdžiui, plaukų ilgį, spalvą, akių spalvą ar pagal tai, ar veikėjas turi kepurę ir pan. Vienas iš privalumų, atskleistų šiame darbe yra tas, kad galima sugeneruoti aukštos kokybės atvaizdus, jei tik duomenų rinkinys yra kruopščiai paruoštas, visi duomenų elementai yra panašūs, vienodai pasiskirstę.

1.4. Įgyvendinimo problemos

Įgyvendinimo problemos siejamos su tuo, kad GAN tipo tinklai mokymosi metu būna nestabilūs [25] ir dažnai po kelių epochų gali subyrėti (angl. *collapse*), kada neuroninis tinklas pradeda generuoti visiškai nesuprantamus ir bereikšmius duomenis. Tokios problemos egzistuoja dėl pačio metodo idėjos, jog vienu metu mokomi du neuroniniai tinklai ir jie tarpusavyje konkuruoja. Egzistuoja sprendimų kaip sumažinti subyrėjimo riziką [32, 33, 43]. Kita problema su GAN tinklais, kad nėra ypač efektyvių būdų spręsti apie sugeneruotų duomenų teisingumą ir kokybę. Dažnai be matematinių metodų tenka kliautis subjektyviu vertinimu. Šiuo atveju, galima įtraukti naudotojus į statistinį vertinimą, kuris taip pat tinka įvertinti kokybei.

1.5. Kokybės vertinimo metrikos

Norint atlikti generuojamų duomenų kokybinę analizę, galima naudoti šias metrikas:

1. pradžios metrika (angl. *Inception Score* – IS);
2. Frečeto pradžios atstumas (angl. *Fretchet Inception Distance* – FID);
3. branduolio pradžios atstumas (angl. *Kernel Inception Distance* – KID);
4. išmoktas skirtingų vaizdo dalių panašumo suvokimas (angl. *Learned Perceptual Image Patch Similarity* – LPIPS);
5. daugialypis struktūrinis panašumas (angl. *Multi Scale Structural Similarity* – MSSSIM).

Norint įvertinti generuojamus duomenis kokybiškai, reikia naudotojų apklausos, nes duomenys net jei ir kokybiškai sugeneruoti, bet neprasmingi – nenaudingi. Tiriamojoje dalyje aprašoma, kokia anketa naudota kokybiniam vertinimui.

1.5.1. Pradžios metrika

Pradžios metrika yra populiari GAN tinklų kiekybinio vertinimo metrika, kuri nustato, kaip realistiškai generuojami duomenys. Pagal autorius [43], metrika prilygsta žmonių vertinimui. Metrika matuoja ar generuojami atvaizdai turi pakankamai skirtingų savybių ir ar kiekvienas atvaizdas yra aiškus, pavyzdžiui, atrodo kaip animacinis personažas. Jei abu šie aspektai tiesa, tuomet metrikos reikšmė bus aukšta, jei priešingai – žema.

Matematinė šios metrikos išraiška atrodo taip:

$$IS = e^{KL(p(y|x) || p(y))} \quad (1.17)$$

kur KL yra Kulbako – Leiblerio divergencija, $p(y|x)$ – semantinių klasių atvaizde pasiskirstymas, y – semantinės klasės, x – atvaizdas, $p(y)$ – ribinis pasiskirstymas.

Kuo aukštesnė metrikos reikšmė, tuo geriau, nes tai parodo, kad neuroninis tinklas išmoko generuoti daug skirtingų aiškių atvaizdų. Žemiausias galimas įvertis yra lygus nuliui, o aukščiausias matematine prasme – begalybė.

1.5.2. Frečeto pradžios atstumas

Frečeto pradžios atstumas yra metrika, kuri parodo, kaip generuojamų duomenų pasiskirstymas sutampa su realių duomenų pasiskirstymu. Tai plačiausiai naudojama metrika kiekybiškai vertinant GAN. Siekiant rasti duomenų pasiskirstymą, naudojamas žemo lygio savybių ištraukėjas (angl. *extractor*), pavyzdžiui, klasifikatorius *Inception V3*, kurio paskutinių sluoksnių išvestyje būna užkoduotas žemo lygio savybių vektorius apibendrinantis atvaizdą. Metrikos formulė [50]:

$$FID = d^2((m, C), (m_\omega, C_\omega)) = \|m - m_\omega\|_2^2 + Tr\left(C + C_\omega - 2(CC_\omega)^{\frac{1}{2}}\right) \quad (1.18)$$

kur d – yra atstumas tarp Gauso pasiskirstymo su vidurkiu ir realių duomenų kovariacija (m, C) , ir Gauso pasiskirstymo su vidurkiu ir sugeneruotų duomenų kovariacija (m_ω, C_ω) .

Kuo žemesnė metrikos reikšmė, tuo sugeneruoti duomenys yra artimesni realiems. Autorių teigimu, ši metrika dera su didėjančiais trikdžiais duomenyse ir žmonių kokybiniu vertinimu.

1.5.3. Branduolio pradžios atstumas

Metrika yra maksimalaus vidutinio neatitikimo (angl. *Maximum Mean Discrepancy*) plėtinys. Maksimalus vidutinis neatitikimas yra statistinė metrika, kuri skirta nustatyti ar du skirstiniai yra identiški. Metrika nustato atstumą tarp vidutinių savybių vektorių ir yra išreikšta šia formule [51]:

$$MMD^2(X, Y) = \|\mu_X - \mu_Y\|_k^2 \quad (1.19)$$

kur X ir Y yra skirstiniai, kuriuos norime palyginti, k – pasiskirstymo funkcija. Tačiau dažniau naudojama empirinė formulės versija:

$$MMD^2(X, Y) = \frac{1}{m(m-1)} \sum_i \sum_{j \neq i} k(x_i, x_j) - 2 \frac{1}{m \cdot m} \sum_i \sum_j k(x_i, y_j) + \frac{1}{m(m-1)} \sum_i \sum_{j \neq i} k(y_i, y_j) \quad (1.20)$$

kur m – elementų kiekis skirstinyje. Taigi KID pasinaudojus MMD metrika suformuoja tokią formulę [52]:

$$KID = MMD^2(f_T, f_{NT}) \quad (1.21)$$

kur f_T – tikrųjų atvaizdų skirstinys, o f_{NT} – netikrųjų atvaizdų skirstinys. Taip pat KID naudoja šią skirstinio funkciją:

$$k(x, y) = (\gamma * x^T y + koef)^{laipsnis} \quad (1.21)$$

kur γ – savybių vektoriaus ilgis, $koef$ – poslinkio reikšmė (hiperparametras), $laipsnis$ – polinominis (daugianario) laipsnis, duomenų elemento dimensijų kiekis. Atvaizdų skirstinių lyginimo atveju, $laipsnis$ yra lygus 3, nes atvaizdai susideda iš 3 dimensijų duomenų elementų – kiekvienam atvaizdo sluoksniui po vieną pikselių intensyvumo vektorių.

Kuo žemesnė metrikos reikšmė, tuo yra geriau, nes tuomet du skirstiniai yra labiau panašūs nei skirtingi.

1.5.4. Išmoktas skirtingų vaizdo dalių panašumo suvokimas

Metrika lygina dviejų atvaizdų, jų skirtingų dalių aktyvacijų vektorius iš pasirinkto giliojo konvoliucinio tinklo sluoksnio. Pagal autorius, šios aktyvacijos savyje turi užkoduotą informaciją, kuri apibrėžia, kaip neuroninis tinklas suvokia atvaizdą. Autorių teigimu, metrika prilygsta žmogaus panašių atvaizdų vertinimui [53].

$$d(x, x_0) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \|w_l * (\hat{y}_{hw}^l - \hat{y}_{0hw}^l)\|_2^2 \quad (1.22)$$

kur \hat{y}^l ir \hat{y}_0^l yra savybių vektoriai iš sluoksnio l , w – hiperparametras, kuris leidžia keisti duomenų mastelį (angl. *scaling*), H ir W – atvaizdo aukštis ir plotis. Ši metrika dažnai naudojama mokant neuroninius tinklus ir gali būti pritaikoma su žemiau pateikta funkcija:

$$L(x, x_0, x_1, h) = -h \log G(d(x, x_0), d(x, x_1)) - (1 - h) \log(1 - G(d(x, x_0), d(x, x_1))) \quad (1.22)$$

kur d – atstumas tarp dviejų skirtingų atvaizdų dalių, G – generatoriaus tinklas, h – iš anksto apmokytas suvokimo indeksas [53].

Žemas šios metrikos dydis leidžia suprasti ar neuroninis tinklas išmoko suvokti ar skirtingi atvaizdai yra panašūs.

1.5.5. Daugialypis struktūrinis panašumas

Metrika remiasi prielaida, kad žmogus geba puikiai suprasti atvaizdo scenos struktūrinę informaciją, o tai padeda suvokti atvaizdo prasmę. Todėl metrika, kuri geba nustatyti neuroninio tinklo struktūrinį atvaizdo suvokimą, yra geras atvaizdo kokybės nustatymo artinys (angl. *approximation*) [54]. Ši metrika iš esmės yra SSIM – struktūrinio panašumo indekso (angl. *Structural Similarity Index Measure*) plėtinys, kuris į metriką įtraukia atvaizdų savybių vektorius skirtinguose atvaizdo dydžiuose. SSIM metrika aprašoma taip:

$$SSIM(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma \quad (1.23)$$

kur x ir y – atvaizdai, kuriuos lyginame. Parametrai α, β, γ yra hiperparametrai ir skirti kontroliuoti komponentus l, c, s . Šie komponentai pateikiami žemesnėse 1.24, 1.25 ir 1.26 formulėse.

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \quad (1.24)$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \quad (1.25)$$

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \quad (1.26)$$

l – atvaizdo šviesumas, c – kontrastas, s – struktūra. μ_x ir μ_y yra vidutinės x ir y reikšmės, o σ_x ir σ_y atvaizdų variacija. $\sigma_x\sigma_y$ – x ir y kovariacija. Konstantos C_1, C_2 ir C_3 apibrėžiamos taip: $C_1 =$

$(K_1L)^2$, $C_2 = (K_2L)^2$ ir $C_3 = \frac{C_2}{2}$. Čia $K_1, K_2 \ll 1$ ir yra skaliarinės konstantos, o L – dinaminės pikselių reikšmės. Pagal autorius, konstantos K_1 ir K_2 , turi tokias reikšmes: $K_1 = 0.01, K_2 = 0.03$. Taikant algoritmą, naudojamas kintančio lango (angl. *sliding window*) metodas, o nustatytos reikšmės išsaugojamos „kokybės“ žemėlapyje [54]. Apmokant neuroninį tinklą reikia, kad metrika turėtų vieną reikšmę, todėl naudojama vidutinė SSIM reikšmė. MSSIM metrika pritaiko mastelį M , kuris leidžia įvertinti SSIM skirtinguose atvaizdo dydžiuose:

$$MSSIM(x, y) = [l_M(x, y)]^{\alpha_M} \prod_{j=1}^M [c_j(x, y)]^{\beta_j} [s_j(x, y)]^{\gamma_j} \quad (1.29)$$

Funkcija gali būti optimizavimo uždavinys neuroninių tinklų mokyme ją modifikavus: $L = 1 - MSSIM$. MSSIM aukštesnė reikšmė reiškia didesnį panašumą, tačiau atliekant mokymą, mums reikia mažinti (optimizuoti) reikšmę.

2. Reikalavimų specifikacija

Šioje dalyje pateikiama projekto reikalavimų specifikacija, apibrėžiamos funkcijos ir kiti svarbūs reikalavimai bei apribojimai.

2.1. Naudotojai

2.1.1. Animatoriai

Tai pagrindiniai sistemos naudotojai, kurie kasdien susiduria su animacinių personažų kūrimu, jų vystymu ir tobulinimu. Naudotojai dažnai turi aukštąjį išsilavinimą informacinių technologijų srityje, todėl kompiuterinio raštingumo įgūdžiai puikūs. Jie kruopštūs ir atsakingi.

2.1.2. Edukatoriai

Vieni iš pagrindinių sistemos naudotojų yra edukatoriai. Jie veda mokymus, kūrybines dirbtuves ar edukacines programas mokyklinio amžiaus vaikams. Jie turi vidurinį arba aukštąjį išsilavinimą, yra baigę pedagogikos kursus ir gali dirbti su mokyklinio amžiaus vaikais. Edukatorių kompiuterinio raštingumo įgūdžiai gali būti vidutiniški – turi mokėti naudotis naršykle. Šie žmonės turi būti atsidavę darbui, nes mokyti vaikus yra iššūkis, o jų lavinimas svarbus siekiant užtikrinti darnų vaiko vystymąsi.

2.1.3. Vaikai

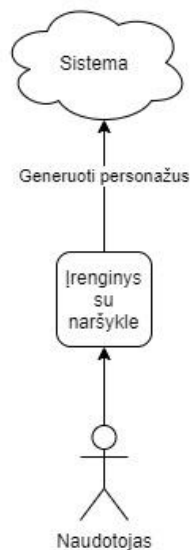
Vaikai yra antriniai sistemos naudotojai, kurie naudos sistemą kūrybinių dirbtuvių ar edukacinių programų metu, kurias ves edukatoriai. Šios kategorijos naudotojai patenka į mokyklinio amžiaus grupę, dažnai turi mažai patirties su informacinėmis technologijomis.

2.2. Apribojimai

2.2.1. Apribojimai sprendimui

Sistema privalo būti internetinė aplikacija ir pasiekama tinkle, nes ją siekiama komercializuoti ir įdiegti kaip paslaugą (angl. *Software as a Service*).

2.2.2. Diegimo aplinka

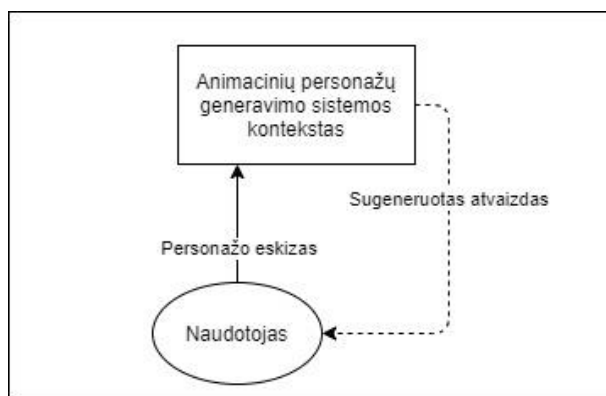


8 pav. Diegimo aplinka

8 pav. pateikiama diegimo aplinkos schema. Ji vaizduoja, kaip sistemos naudotojai savo įrenginyje su naršykle gali pasiekti debesyse įdiegtą sistemą. Naudotojai sistemos pagalba gali generuoti animacinius personažus.

2.3. Veiklos sudėtis

Sistemos veiklos konteksto modelis pavaizduotas 9 pav.



9 pav. Sistemos veiklos konteksto modelis

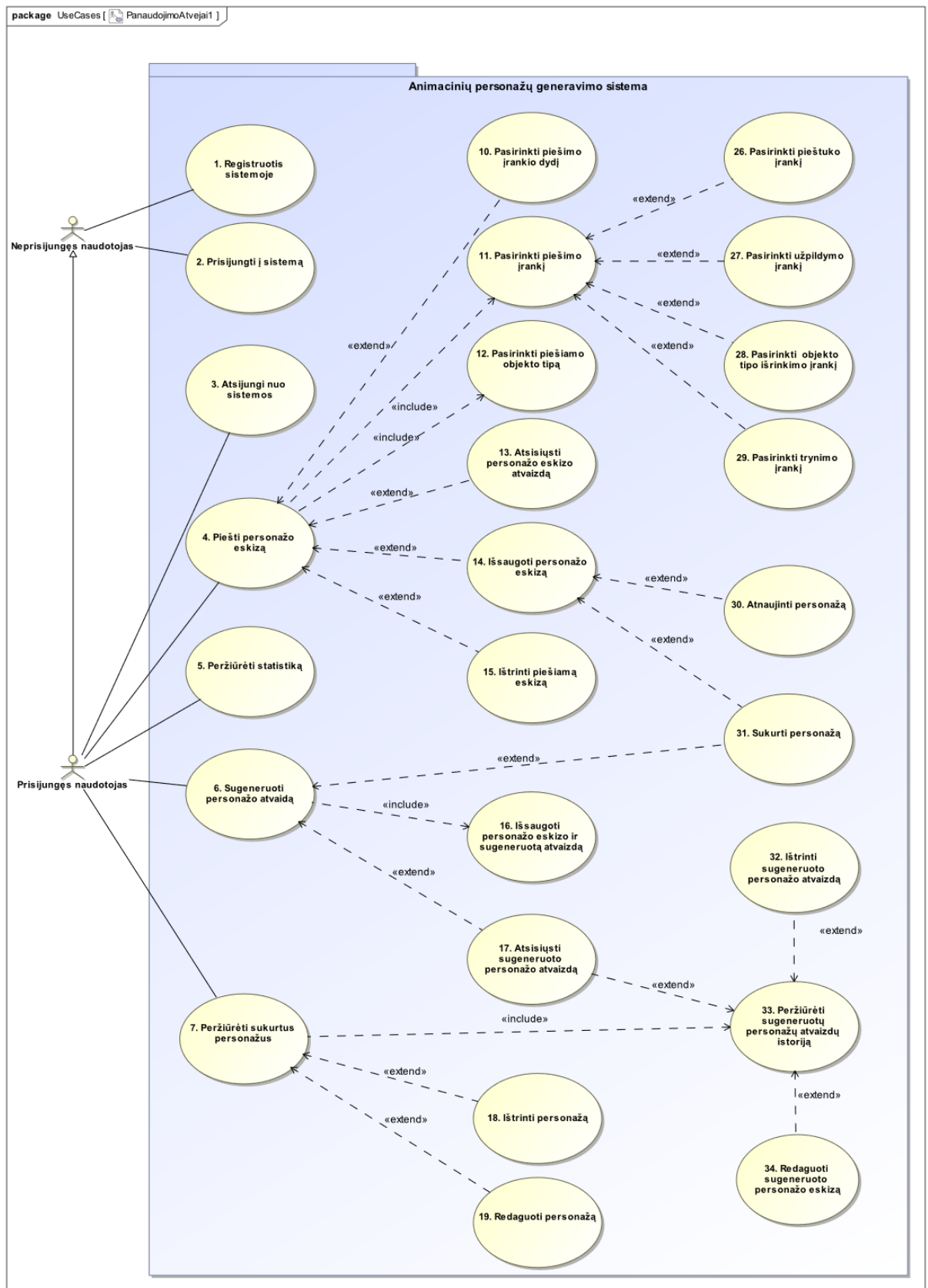
Naudotojas sistemos grafinėje sąsajoje piešia personažo idėjos eskizą, kuris naudojamas sugeneruoti atvaizdą. Veiklos srautai paaiškinti 2 lentelėje.

2 lentelė Veiklos kontekste vaizduojamų srautų paaiškinimas

Srautas	Dalyvis	Paaiškinimas
Personazo eskizas	Iš naudotojo	Naudotojas piešia personažo idėjos eskizą, kurį panaudoja animaciniam personažui sugeneruoti.
Sugeneruotas atvaizdas	Iš sistemos	Sistema pagal naudotojo eskizą sugeneruoja personažo atvaizdą ir jį parodo grafinėje sąsajoje.

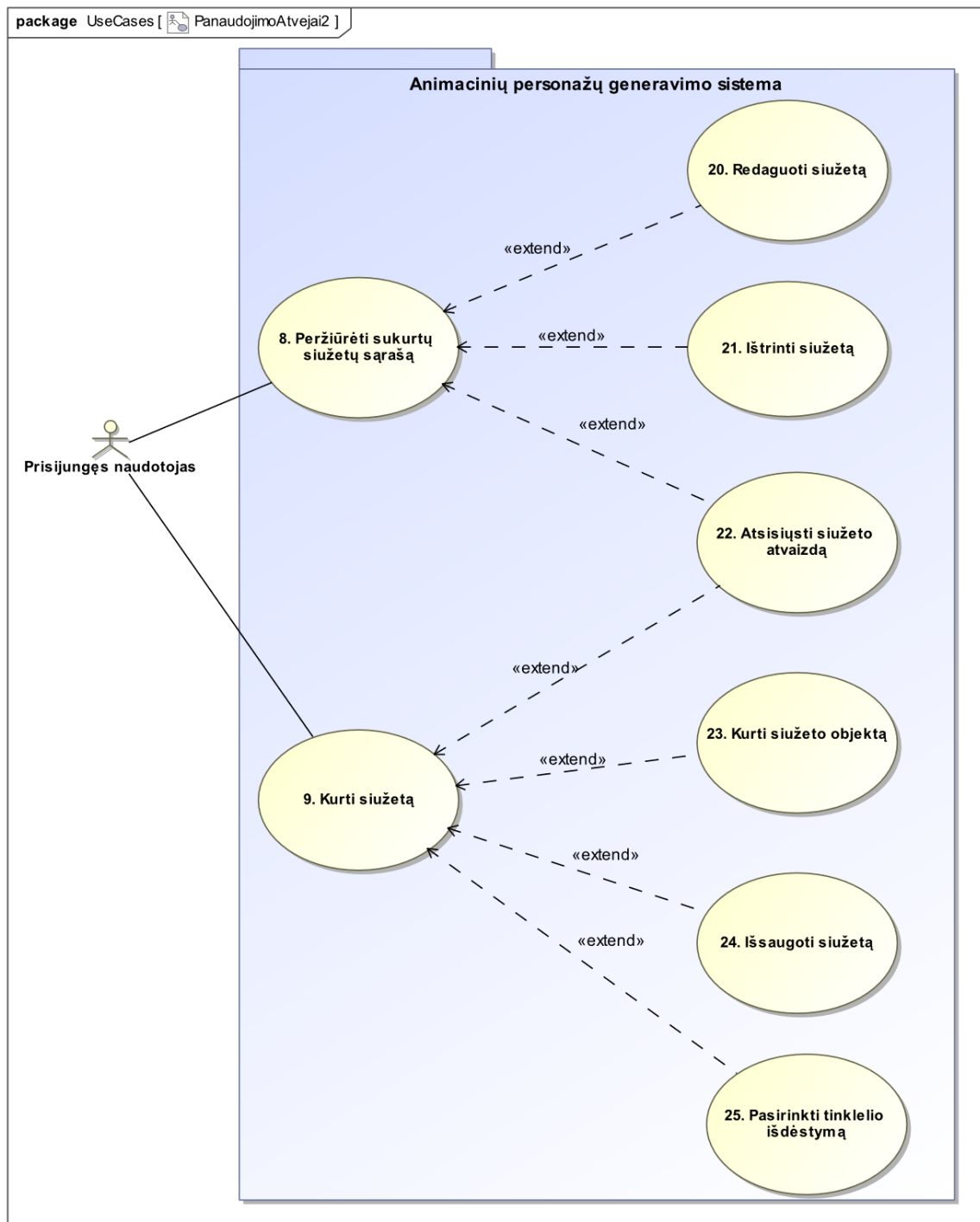
2.4. Sistemos sudėtis

2.4.1. Sistemos ribos



10 pav. Panaudojimo atvejų diagrama (1 dalis)

Sistemos panaudojimo atvejų diagrama pavaizduota 10 pav. ir 11 pav. Sistemoje yra dviejų tipų naudotojai: prisijungęs ir neprisijungęs naudotojas. Neprisijungęs naudotojas gali užsiregistruoti ir prisijungti. Prisijungęs naudotojas gali pasiekti visas sistemos funkcijas.



11 pav. Panaudojimo atvejų diagrama (2 dalis)

Sistemoje realizuoti visi panaudojimo atvejai:

1. Registruotis sistemoje;
2. Prisijungti į sistemą;
3. Atsijungti nuo sistemos;
4. Piešti personažo eskizą;

5. Peržiūrėti statistiką;
6. Sugeneruoti personažo atvaizdą;
7. Peržiūrėti sukurtus personažus;
8. Peržiūrėti sukurtų siužetų sąrašą;
9. Kurti siužetą;
10. Pasirinkti piešimo įrankio dydį;
11. Pasirinkti piešimo įrankį;
12. Pasirinkti piešiamo objekto tipą;
13. Atsisiųsti personažo eskizo atvaizdą;
14. Išsaugoti personažo eskizą;
15. Ištrinti piešiamą eskizą;
16. Išsaugoti personažo eskizą ir sugeneruotą atvaizdą;
17. Atsisiųsti sugeneruoto personažo atvaizdą;
18. Ištrinti personažą;
19. Redaguoti personažą;
20. Redaguoti siužetą;
21. Ištrinti siužetą;
22. Atsisiųsti siužeto atvaizdą;
23. Kurti siužeto objektą;
24. Išsaugoti siužetą;
25. Pasirinkti tinklelio išdėstymą;
26. Pasirinkti pieštuko įrankį;
27. Pasirinkti užpildymo įrankį;
28. Pasirinkti objekto tipo išrinkimo įrankį;
29. Pasirinkti trynimo įrankį;
30. Atnaujinti personažą;
31. Sukurti personažą;
32. Ištrinti sugeneruoto personažo atvaizdą;
33. Peržiūrėti sugeneruotų personažų atvaizdų istoriją;
34. Redaguoti sugeneruoto personažo eskizą.

2.4.2. Panaudojimo atvejų sąrašas

Šiame skyrelyje lentelėmis specifikuojami patys svarbiausi panaudojimo atvejai:

- 4 PA Piešti personažo eskizą - 3 lentelė;
- 6 PA Sugeneruoti personažo atvaizdą - 4 lentelė;
- 7 PA Peržiūrėti sukurtus personažus - 5 lentelė;
- 9 PA Kurti siužetą - 6 lentelė;

Visi likę panaudojimo atvejai yra specifikuojami projekto techninėje dokumentacijoje [55].

3 lentelė Panaudojimo atvejo „Piešti personažo eskizą“ specifikacija

4 Panaudojimo atvejis: Piešti personažo eskizą	
Tikslas/Uždavinys	Leisti naudotojui piešti eskizus, kurie gali būti sugeneruoti į personažų rastrinius atvaizdus.

Aprašymas	Naudotojas, norėdamas sugeneruoti personažo atvaizdą, turi pateikti eskizą – semantinį objektų žemėlapi, kurį sistema leidžia nupiešti svetainėje naudojant šiam uždaviniui pritaikytą piešimo įrankį.	
Prieš-sąlyga	Naudotojas turi būti prisijungęs prie sistemos.	
Aktorius	Prisijungęs naudotojas	
Sužadinimo sąlyga	Naudotojas įjungia piešimo įrankį.	
Susiję panaudojimo atvejai	Išplečiantys PA	10, 13, 14, 15
	Apimantys PA	11, 12
Pagrindinis scenarijus		
Naudotojo veiksmai	Sistemos veiksmai	
1. Meniu pasirenka piešimo įrankį.	2. Atvaizduoja piešimo įrankio grafines sąsajas.	
3. Įvykdo 11 ir 12 panaudojimo atvejus.		
5. Baigiamas panaudojimo atvejis.		
Po-sąlyga	Naudotojas gali pradėti piešti personažo eskizą.	
Alternatyvūs scenarijai		
-		

4 lentelė Panaudojimo atvejo „Sugeneruoti personažo atvaizdą“ specifikacija

6 Panaudojimo atvejais: Sugeneruoti personažo atvaizdą		
Tikslas/Uždavinys		
Aprašymas	Nupiešus personažo eskizą naudotojas gali paversti jį į personažo atvaizdą, kuris sudaromas konvertavus kiekvieną semantinį objektą į tekstūrą. Sugeneruotas personažo atvaizdas išsaugomas naudotojo personažų istorijoje.	
Prieš-sąlyga	Naudotojas privalo būti prisijungęs prie sistemos.	
Aktorius	Prisijungęs naudotojas	
Sužadinimo sąlyga	Naudotojas paspaudžia „Generuoti personažą“ mygtuką piešimo įrankyje.	
Susiję panaudojimo atvejai	Išplečiantys PA	17, 31
	Apimantys PA	16
Pagrindinis scenarijus		
Naudotojo veiksmai	Sistemos veiksmai	
1. Paspaudžia „Generuoti personažą“ mygtuką piešimo įrankyje.	2. Personažo eskizas siunčiamas į skaičiavimų serverį.	
	3. Nuskaitomas personažo eskizas ir perduodamas į neuroninį tinklą.	
	4. Neuroninis tinklas sugeneruoja personažo atvaizdą pagal semantinį objektų žemėlapi.	
	5. Jei kuriamas naujas personažas, įvykdomas 31 panaudojimo atvejis.	
	6. Įvykdomas 16 panaudojimo atvejais.	

	7. Parodomas sugeneruoto personažo atvaizdas puslapyje.
	8. Baigiamas panaudojimo atvejis.
Po-sąlyga	Sėkmingai sugeneruojamas personažo atvaizdas.
Alternatyvūs scenarijai	
Vykdant personažo generavimą, įvykus nenumatytai klaidai grafinėje sąsajoje rodomas klaidos pranešimas.	

5 lentelė Panaudojimo atvejo „Peržiūrėti sukurtus personažus“ specifikacija

7 Panaudojimo atvejis: Peržiūrėti sukurtus personažus		
Tikslas/Uždavinys	Leisti naudotojui bet kuriuo metu peržiūrėti sukurtų personažų istoriją.	
Aprašymas	Visi sukurti personažai yra išsaugomi ir susiejami su naudotoju, kad vėliau sugrįžus į sistemą būtų galima peržiūrėti sukurtus personažus, o jų atvaizdus atsisiųsti arba pašalinti. Kiekvienas personažas gali turėti daug sugeneruotų atvaizdų. Personažą galima kurti etapais – modifikuojant jo eskizą ir sugeneruojant naują atvaizdą.	
Prieš-sąlyga	Naudotojas privalo būti prisijungęs prie sistemos.	
Aktorius	Prisijungęs naudotojas	
Sužadinimo sąlyga	Naudotojas meniu pasirenka nuorodą „Mano personažai“.	
Susiję panaudojimo atvejai	Išplečiantys PA	18, 19
	Apimantys PA	33
Pagrindinis scenarijus		
Naudotojo veiksmai	Sistemos veiksmai	
1. Meniu pasirenka „Mano personažai“ nuorodą.	2. Atvaizduojamas sukurtų personažų sąrašas.	
	3. Įvykdomas 33 panaudojimo atvejis.	
4. Gali įvykdyti 18 arba 19 panaudojimo atvejus.		
5. Baigiamas panaudojimo atvejis.		
Po-sąlyga	Naudotojas mato visus (asmeninius) sukurtus personažus ir jų sugeneruotus atvaizdus sąrašė.	
Alternatyvūs scenarijai		
Naudotojo veiksmai	Sistemos veiksmai	
1. Meniu pasirenka „Mano personažai“ nuorodą.	2. Sukurtų personažų sąrašas tuščias, todėl rodomas pranešimas sąrašo viduryje teigiantis, kad naudotojas dar nėra sugeneravęs nei vieno personažo.	
	3. Baigiamas panaudojimo atvejis.	

6 lentelė Panaudojimo atvejo „Kurti siužetą“ specifikacija

9 Panaudojimo atvejis: Kurti siužetą	
Tikslas/Uždavinys	Leisti naudotojui panaudoti savo sukurtus personažus kuriant siužeto lentas (komiksus).
Aprašymas	Siužeto lentos leidžia įprasminti sukurtą personažą kažkokioje scenoje, kurią gali modeliuoti naudodamas sukurtų personažų

	atvaizdus ir teksto objektus. Naudotojas gali pasirinkti siužeto lentos (komikso) tinkelio išdėstymą.	
Prieš-sąlyga	Naudotojas privalo būti prisijungęs prie sistemos.	
Aktorius	Prisijungęs naudotojas	
Sužadinimo sąlyga	Naudotojas meniu pasirenka nuorodą „Siužeto kūrimas“.	
Susiję panaudojimo atvejai	Išplečiantys PA	22, 23, 24, 25
	Apimantys PA	-
Pagrindinis scenarijus		
Naudotojo veiksmai	Sistemos veiksmai	
1. Meniu pasirenka „Siužeto kūrimas“ nuorodą.	2. Atvaizduojama siužeto kūrimo sąsaja.	
3. Gali įvykdyti 22, 23, 24 arba 25 panaudojimo atvejus.		
4. Baigiamas panaudojimo atvejis.		
Po-sąlyga	Naudotojas mato interaktyvią siužeto kūrimo sąsają.	
Alternatyvūs scenarijai		
-		

2.5. Funkciniai reikalavimai ir reikalavimai duomenims

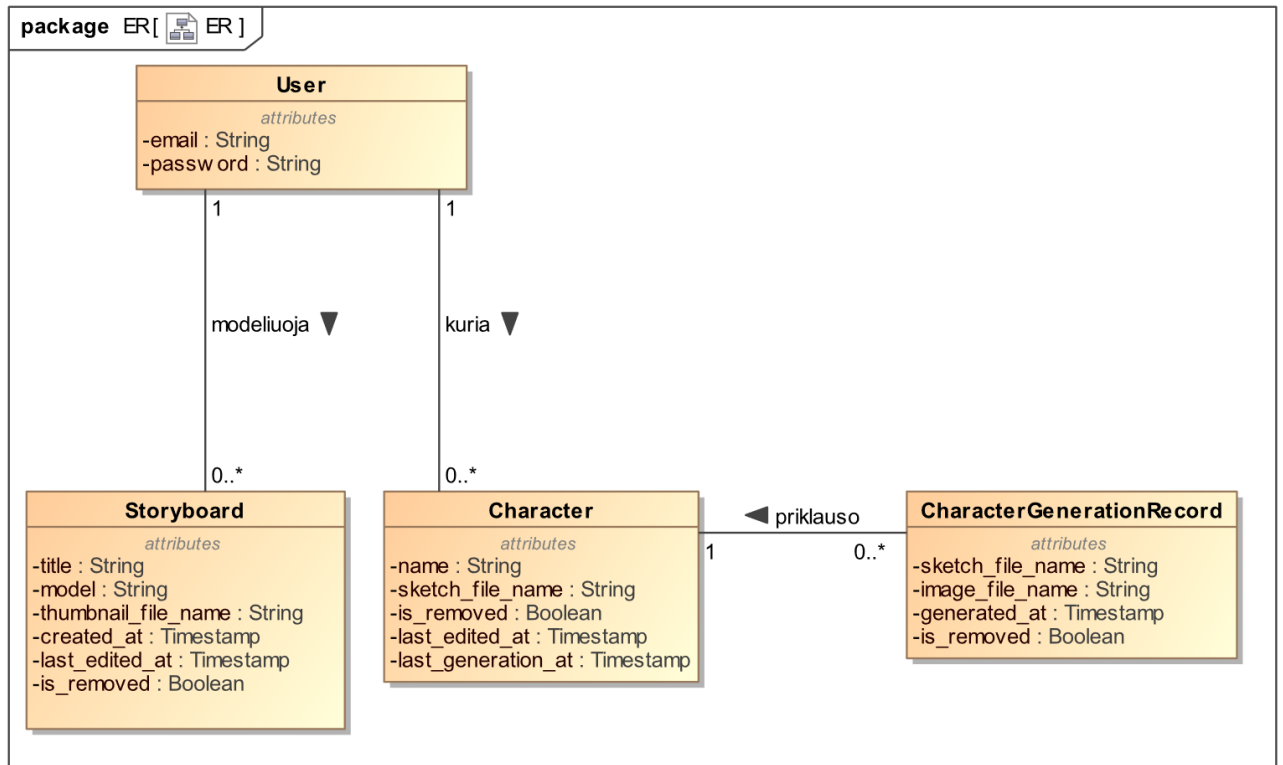
2.5.1. Funkciniai reikalavimai

1. Sistema turi turėti slapuką (angl. *cookie*), kuris naudotojui leidžia naudotis sistema neprisijungiant į ją iš naujo. Slapuko gyvavimo laikas – 24 valandos. Tai leidžia kiekvieną kartą įjungus sistemos svetainę išvengti prisijungimo formos pildymo.
2. Baigus personažo generavimą, sugeneruotas atvaizdas rodomas tame pačiame puslapyje neįkėlus jo iš naujo. Leidžia patogiau naudotis sistema, nes sugeneruotas personažas matomas iškart atlikus generavimą.

2.5.2. Reikalavimai duomenims

Kuriamos sistemos reikalavimai duomenims pateikiami 12 pav. Yra 4 esybės:

- *User* – saugo reikiamą informaciją apie naudotoją. Esybės požymių specifikacija pateikiama 7 lentelėje;
- *Storyboard* – saugo reikiamą informaciją apie siužeto lentas. Esybės požymių specifikacija pateikiama 8 lentelėje;
- *Character* – saugo reikiamą informaciją apie sukurtą personažą. Esybės požymių specifikacija pateikiama 9 lentelėje;
- *CharacterGenerationRecord* – saugo reikiamą informaciją apie personažo sugeneravimo įvykį. Esybės požymių specifikacija pateikiama 10 lentelėje.



12 pav. Esybių ryšių modelis kuriamai sistemai

Duomenų žodynas (duomenų modelio specifikacija):

7 lentelė Esybės „User“ požymių specifikacija

Esbė: User			
Požymis	Galimos reikšmės	Paiškinimas	Privalomas?
email	Teksto eilutė.	Naudotojo elektroninio pašto adresas.	Taip
password	Teksto eilutė. Šifruojama saugojant į duomenų bazę.	Paskyros slaptažodis.	Taip
name	Teksto eilutė.	Naudotojo vardas ir pavardė.	Taip

8 lentelė Esybės „Storyboard“ požymių specifikacija

Esbė: Storyboard			
Požymis	Galimos reikšmės	Paiškinimas	Privalomas?
title	Teksto eilutė.	Siužeto lentos pavadinimas.	Taip
model	Teksto eilutė.	Siužeto modelio JSON formatu išsaugoti duomenys.	Taip
thumbnail_file_name	Teksto eilutė.	Siužeto atvaizdo failo pavadinimas failų sistemoje.	Taip
created_at	Datos ir laiko reikšmė.	Kada buvo sukurta siužeto lenta.	Taip
last_edited_at	Datos ir laiko reikšmė.	Kada paskutinį kartą buvo redaguota siužeto lenta.	Taip
is_removed	Dvejtainė reikšmė.	Ar siužeto lenta pašalinta.	Taip

9 lentelė Esysbės „Character“ požymių specifikacija

Esysbė: Character			
Požymis	Galimos reikšmės	Paiškinimas	Privalomas?
name	Teksto eilutė.	Personažo vardas.	Taip
sketch_file_name	Teksto eilutė.	Personažo eskizo failo pavadinimas failų sistemoje.	Ne
is_removed	Dvejtainė reikšmė.	Ar personažas pašalintas.	Taip
last_edited_at	Datos ir laiko reikšmė.	Kada paskutinį kartą buvo redaguotas personažas.	Taip
last_generation_at	Datos ir laiko reikšmė.	Kada paskutinį kartą buvo sugeneruotas personažo atvaizdas.	Ne

10 lentelė Esysbės „CharacterGenerationRecord“ požymių specifikacija

Esysbė: CharacterGenerationRecord			
Požymis	Galimos reikšmės	Paiškinimas	Privalomas?
sketch_file_name	Teksto eilutė.	Sugeneruoto personažo eskizo failo vardas failų sistemoje.	Taip
image_file_name	Teksto eilutė.	Sugeneruoto personažo atvaizdo failo vardas failų sistemoje.	Taip
generated_at	Datos ir laiko reikšmė.	Kada buvo sugeneruotas personažo atvaizdas.	Taip
is_removed	Dvejtainė reikšmė.	Ar sugeneruotas personažas ištrintas.	Taip

2.6. Nefunkciniai reikalavimai

2.6.1. Reikalavimai sistemos išvaizdai

1. Visoje sistemoje turi būti naudojamas vienas ir tas pats šrifto stilius (Roboto). Tokiu būdu išlaikomas vientisumas ir grafinės naudotojo sąsajos tvarkingumas.
2. Visi klaidų pranešimai turi būti atvaizduojami raudona spalva. Raudona spalva dažnai simbolizuoja klaidą.

2.6.2. Reikalavimai panaudojamumui

1. Formos, kurios gali išsaugoti tam tikrus duomenis, pavyzdžiui, prisijungimo vardas (prisijungimo formoje), turėtų išsaugoti juos ir klaidų atvejų (blogai įvedus slaptažodį), automatiškai užpildyti laukus. Tai pagerina naudotojo patirtį, nes nereikia iš naujo įvesti prisijungimo vardo ar el. pašto adreso (prisijungimo formos atveju).
2. Sistema turi numatyti neteisingų duomenų pateikimą į formas ir atlikti validaciją. Reikia vengti netikėtų klaidų, kurios trikdo sistemos veikimą.
3. Personažas turi būti sugeneruojamas trumpiau nei per 4 sekundes. Sunku naudotis sistema, jei tenka ilgai laukti vienos iš pagrindinių sistemos funkcijos vykdymo pabaigos.

2.6.3. Reikalavimai veikimo sąlygoms

1. Sistema turi būti WEB aplikacija, kad naudotojai turėdami interneto ryšį galėtų bet kada pasiekti sistemą ja paviešinus.

2.6.4. Reikalavimai sistemos priežiūrai

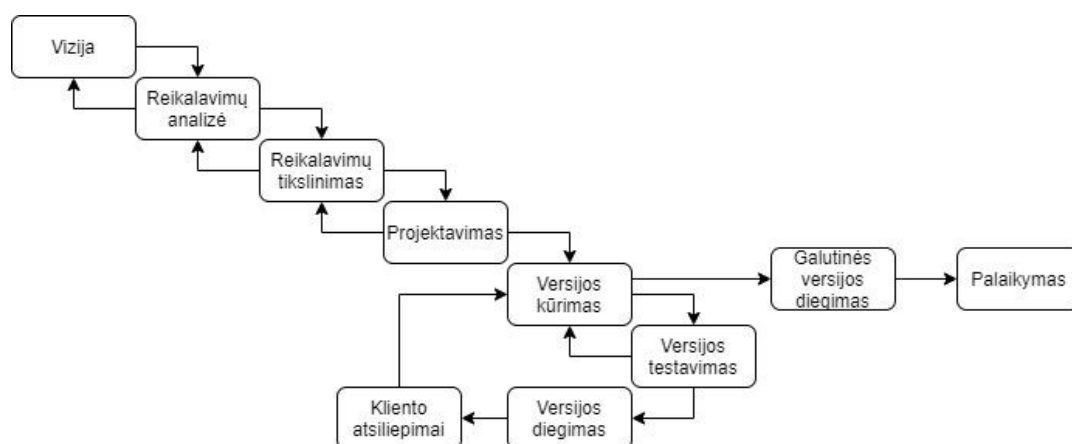
1. Sistemos serverio infrastruktūra turi būti kas pusmetį profilaktiškai tikrinama dėl gedimų prevencijos. Tą užtikrina užsakovas.
2. Po sistemos išleidimo atsiradus trūkumų, juos turi pagal nustatytus įkainius taisyti sistemos kūrėjai. Sistemos kūrėjų komanda pažįsta sistemą, todėl gali atlikti reikalingus taisymus.
3. Sistemą turi būti prieinama įvairiuose įrenginiuose: kompiuteriuose, mobiliuosiuose telefonuose, planšetiniuose kompiuteriuose. Sistema kuriama naudojant į ekrano dydžio pokyčius reaguojantį stilių.

2.6.5. Reikalavimai saugumui

1. Sistemos naudotojai negali pasiekti didesnes teises turinčių naudotojų funkcijų. Naudotojai, kurių teisės viršija numatytąsias, gali atlikti piktavališkus veiksmus ir pasinaudoti sistema atlikti neetišką įsilaužimą bei pavogti duomenis.
2. Naudotojų informacija turi būti saugoma pagal užsakovo įmonėje nustatytą duomenų saugojimo politiką. Naudotojų duomenis saugo įstatymai, jei šie pažeidžiami, sistemos savininkas gali gauti baudą. Tą užtikrina užsakovas.

2.7. Sistemos pateikimo žingsniai

Sistemos kūrimo procesas – evoliucinis iteracinis, kuris pavaizduotas 13 pav.



13 pav. Sistemos kūrimo procesas

Jis leidžia įtraukti užsakovą į kūrimo procesą, nes vyksta nuolatiniai naujų versijų pristatymai, o gauti atsiliepimai leidžia tobulinti programinę įrangą ir reikalavimus. Šis modelis tinka ir dėl to, kad sistema yra reliatyviai maža, o visi reikalavimai projekto pradžioje dar nėra aiškūs.

3. Architektūros specifikacija

Šioje dalyje pateikiami pagrindiniai sistemos architektūrinio projekto sprendimai.

3.1. Architektūros pateikimas

Architektūrinis projektas buvo kuriamas naudojant UML modeliavimo kalbą su *MagicDraw* programinės įrangos paketu. Architektūros vaizdai atitinka tam tikras diagramas. Vaizdų ir diagramų sąsaja pateikiama šiame sąrašė:

- statinis vaizdas – modulių (paketų) ir klasių diagramos;
- dinaminis vaizdas – būsenų, veiklos ir sekų diagramos;
- išdėstymo vaizdas – diegimo diagrama;
- duomenų vaizdas – esybių ryšių diagrama, kuri pavaizduota klasių diagramos pagalba.

Dinaminio vaizdo diagramos bus pateikiamos tik patiems svarbiausiems panaudojimo atvejams:

- 4 PA Piešti personažo eskizą;
- 6 PA Sugeneruoti personažo atvaizdą;
- 7 PA Peržiūrėti sukurtus personažus;
- 9 PA Kurti siužetą.

Visų kitų panaudojimo atvejų dinaminiai vaizdai pateikiami projekto techninėje dokumentacijoje [55].

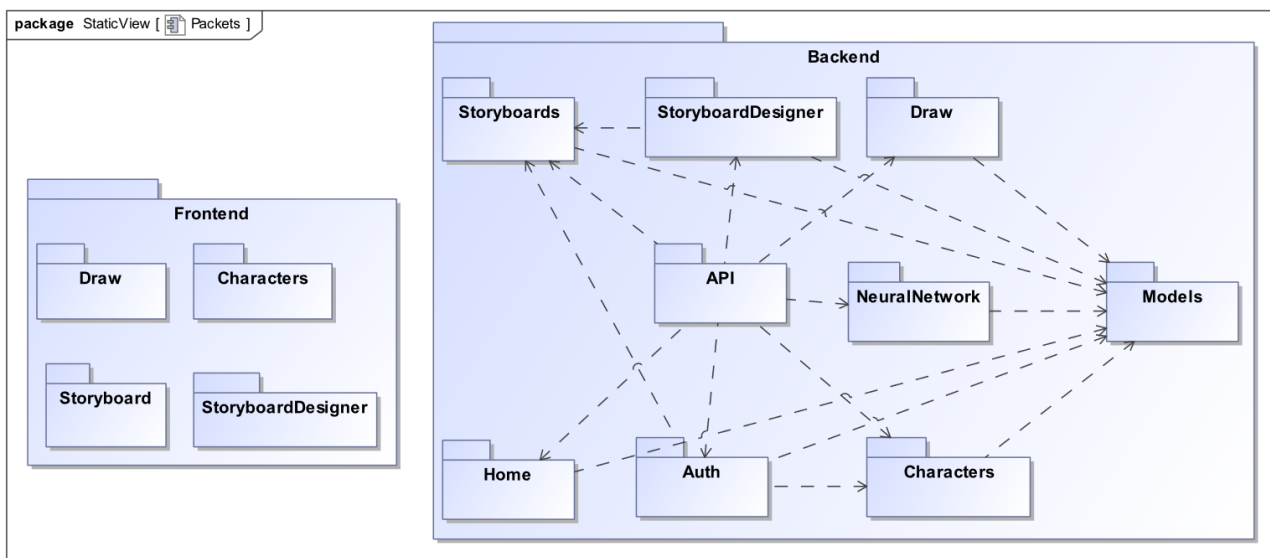
3.2. Architektūros tikslai ir apribojimai

Reikalavimai turintys įtakos architektūriniam sprendimams (paimti iš reikalavimų specifikacijos):

- sistema turi būti WEB aplikacija;
- sistemą turi būti prieinama įvairiuose įrenginiuose: kompiuteriuose, mobiliuosiuose telefonuose, planšetiniuose kompiuteriuose. Sistema kuriama naudojant į ekrano dydžio pokyčius reaguojantį stilių;
- personažas turi būti sugeneruojamas trumpiau nei per 4 sekundes;
- sistemos naudotojai negali pasiekti didesnes teises turinčių naudotojų funkcijų.

3.3. Sistemos statinis vaizdas

Bendra sistemos programinės įrangos modulių diagrama pavaizduota 14 pav.



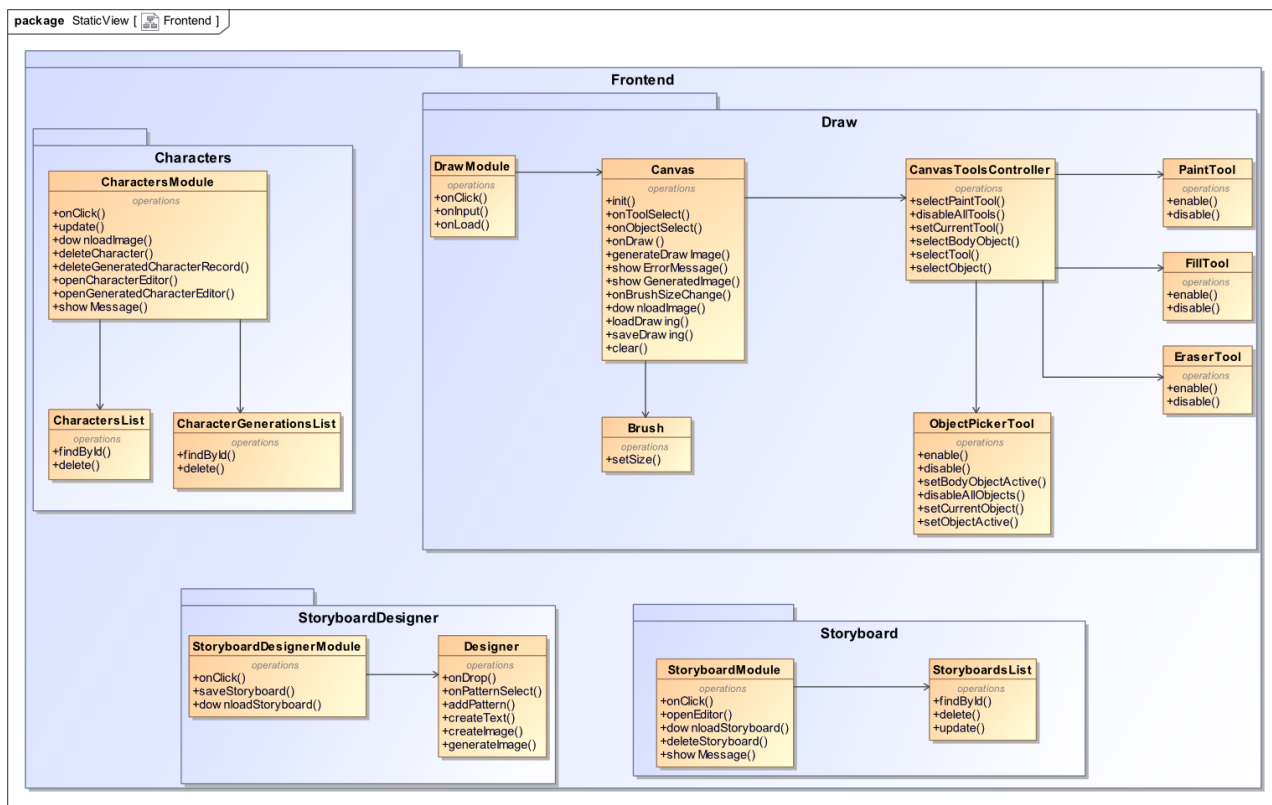
14 pav. Sistemos paketų diagrama

Sistema suskaidyta į dvi dalis: *Frontend* – naudotojo sąsajos programinės įrangos dalis (programinis kodo modulis veikiantis naudotojo įrenginio naršyklėje), *Backend* – serverio sluoksnio programinės įrangos dalis.

3.3.1. Modulių detalizavimas

3.3.1.1. *Frontend* programinės įrangos dalis

Šios programinės įrangos dalies klasių diagrama pavaizduota 15 pav.



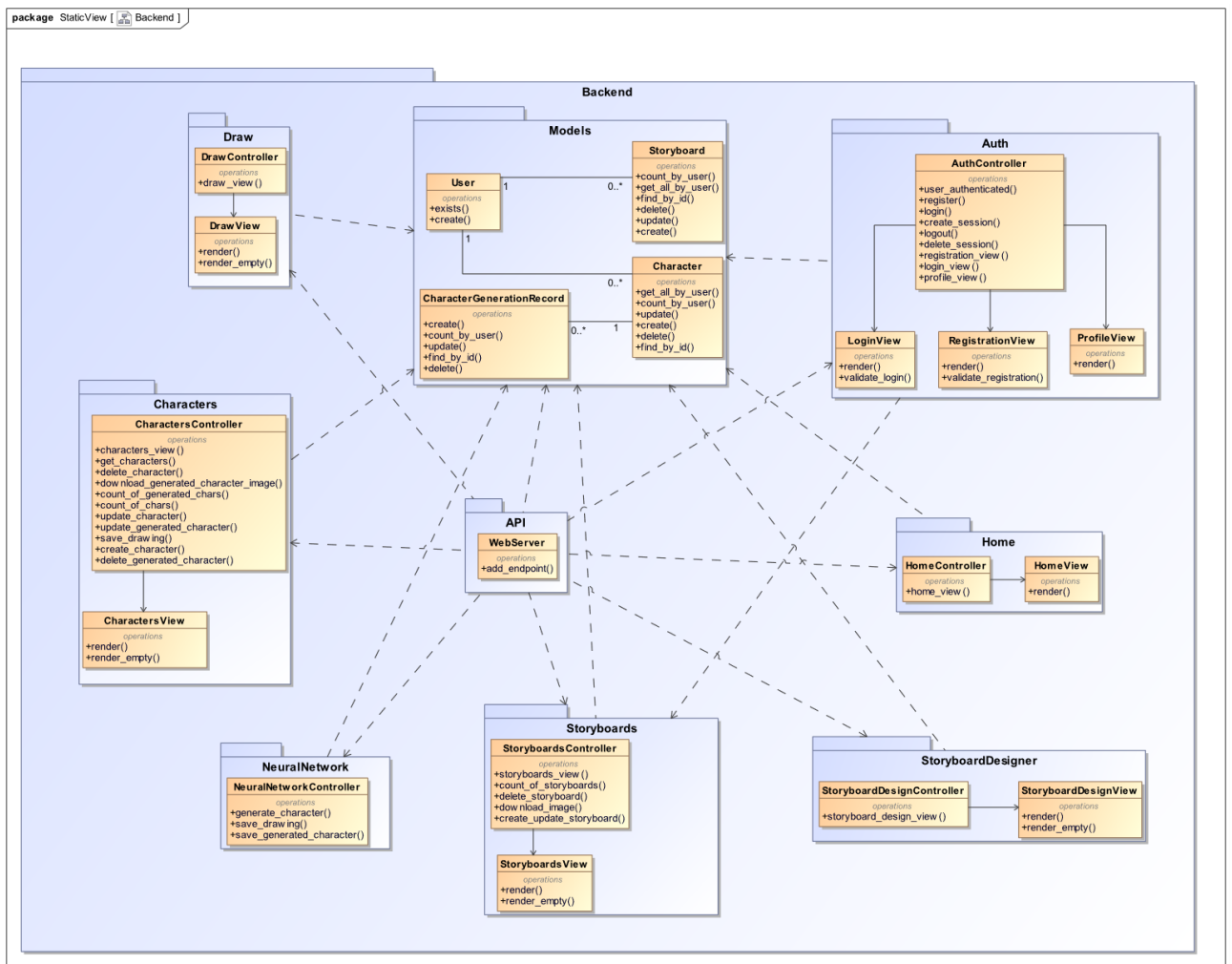
15 pav. Frontend programinės įrangos dalies klasių diagrama

Dalį sudaro keturi moduliai: *Characters*, *Draw*, *Storyboard* ir *StoryboardDesigner*.

- *Storyboard* modulis įgalina sukurtų siužetų sąrašo peržiūrą;
- *Draw* modulis įgalina piešimo įrankio veikimą;
- *Characters* modulis atsakingas už sukurtų ir sugeneruotų personažų sąrašo veikimą;
- *StoryboardDesigner* modulis atsakingas už siužeto modeliavimo sąsajos veikimą.

3.3.1.2. Backend programinės įrangos dalis

Šios programinės įrangos dalies klasių diagrama pavaizduota 16 pav.



16 pav. Backend programinės įrangos dalies klasių diagrama

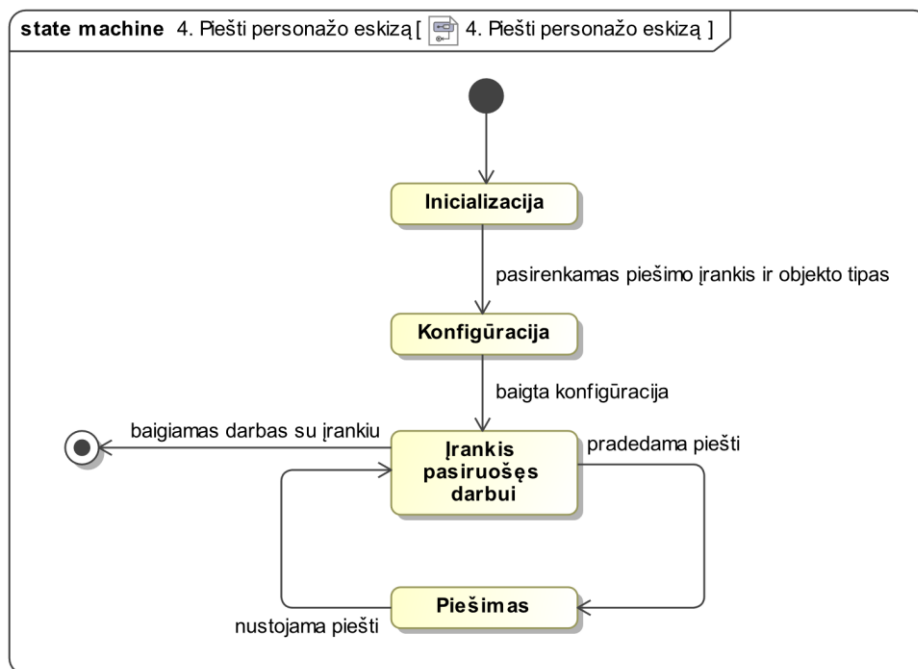
Programinės įrangos dalį sudaro devyni moduliai:

- *Draw* modulis apdoroja piešimo įrankio puslapio atvaizdavimo veiksmus;
- *Characters* modulis įgyvendina numatytas sukurtų ir sugeneruotų personažų sąrašo (istorijos) funkcijas;
- *NeuralNetwork* modulis atsakingas už personažo atvaizdo sugeneravimą;
- *Home* modulis įgyvendina pagrindinio svetainės puslapio funkcionalumą;
- *Auth* modulis apdoroja naudotojų identifikavimą, autorizaciją ir profilio puslapio atvaizdavimą;
- *API* modulis apdoroja serverio sluoksnio sąsają su kliento sluoksniu;
- *Storyboards* modulis įgyvendina siužetų sąrašo (istorijos) funkcijas;
- *StoryboardDesigner* modulis įgyvendina siužeto modeliavimo funkcijas;
- *Models* modulis įgyvendina komunikacijos su duomenų baze funkcijas.

3.4. Sistemos dinaminis vaizdas

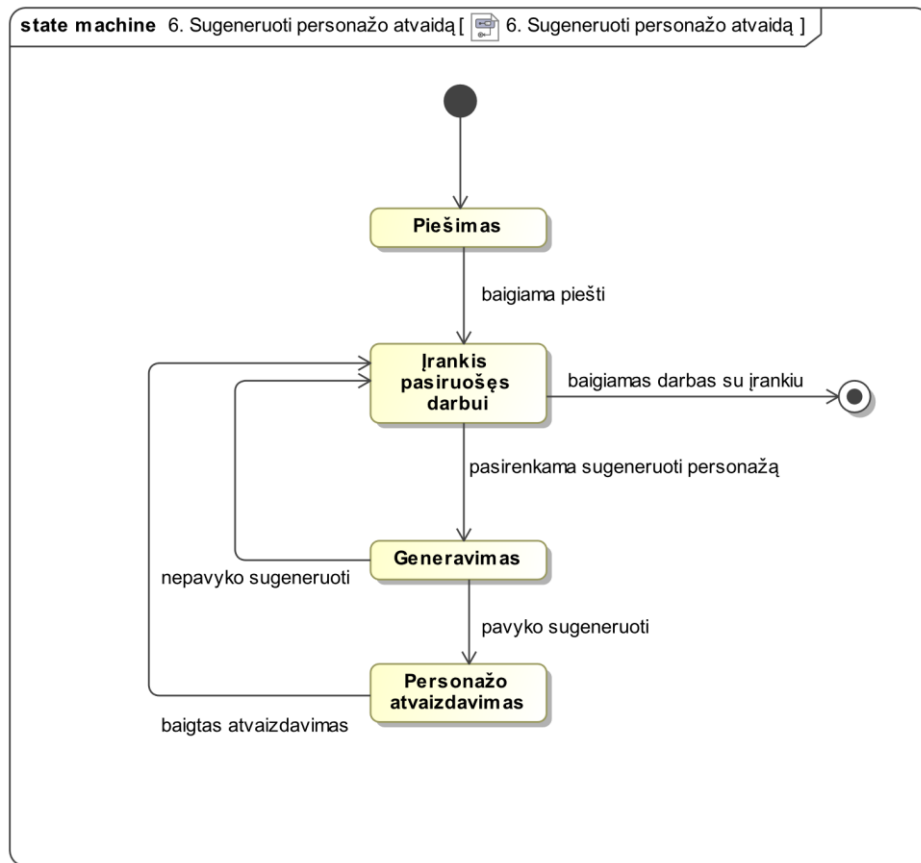
3.4.1. Būsenų diagramos

Panaudojimo atvejo „Piešti personažo eskizą“ būsenų diagrama pavaizduota 17 pav. Įkėlus sąsają, naudotojas pasirenka piešimo įrankį ir objekto tipą. Atlikus konfigūraciją – galima pradėti piešti. Baigus piešimą, įrankis pereina į laukimo režimą.



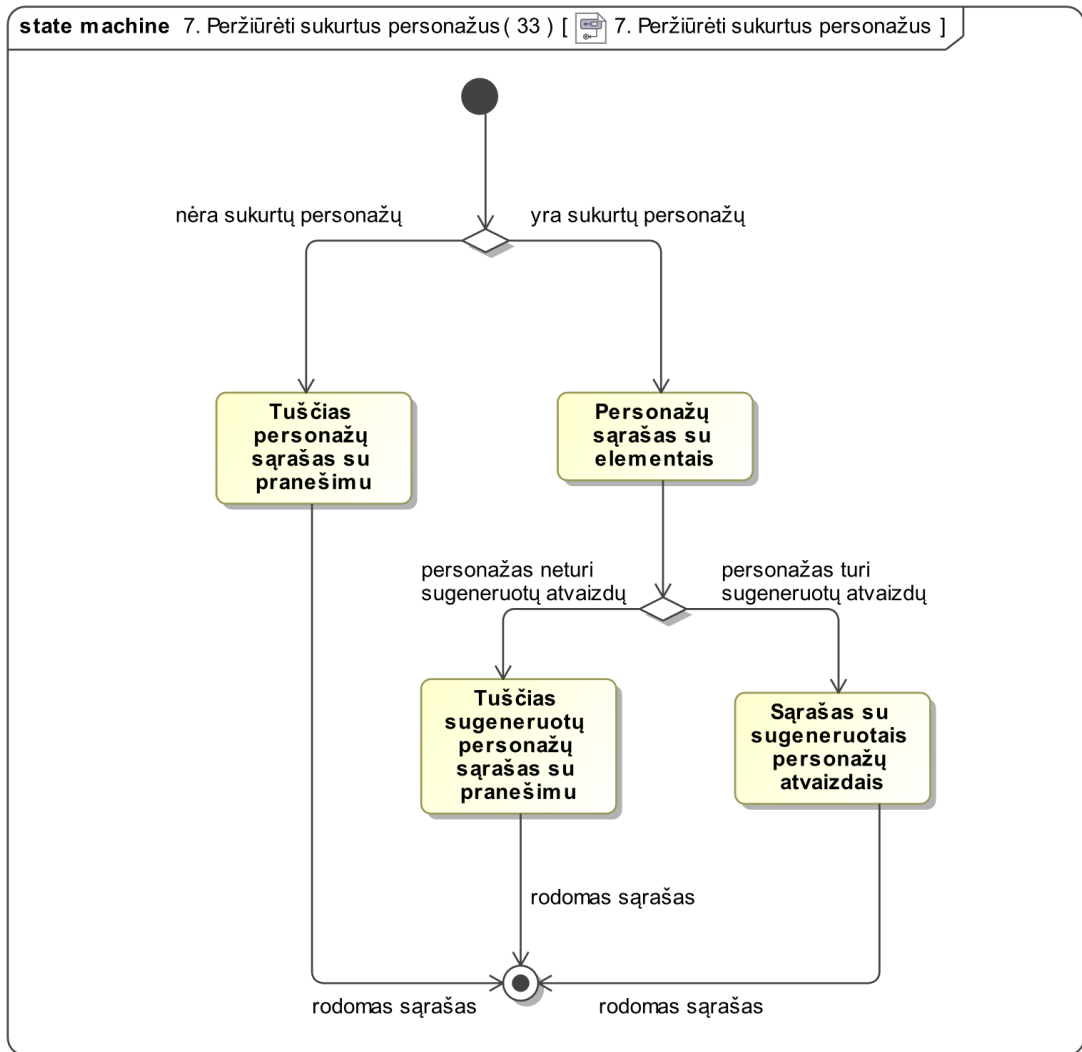
17 pav. Panaudojimo atvejo Nr. 4 būsenų diagrama

Panaudojimo atvejo „Sugeneruoti personažo atvaizdą“ būsenų diagrama pavaizduota 18 pav. Atlikus piešimą naudotojas gali nupieštą eskizą sugeneruoti į personažo atvaizdą. Tą padarius sistema pereina į generavimo būseną. Baigus generavimą įrankis pereina į laukimo būseną.



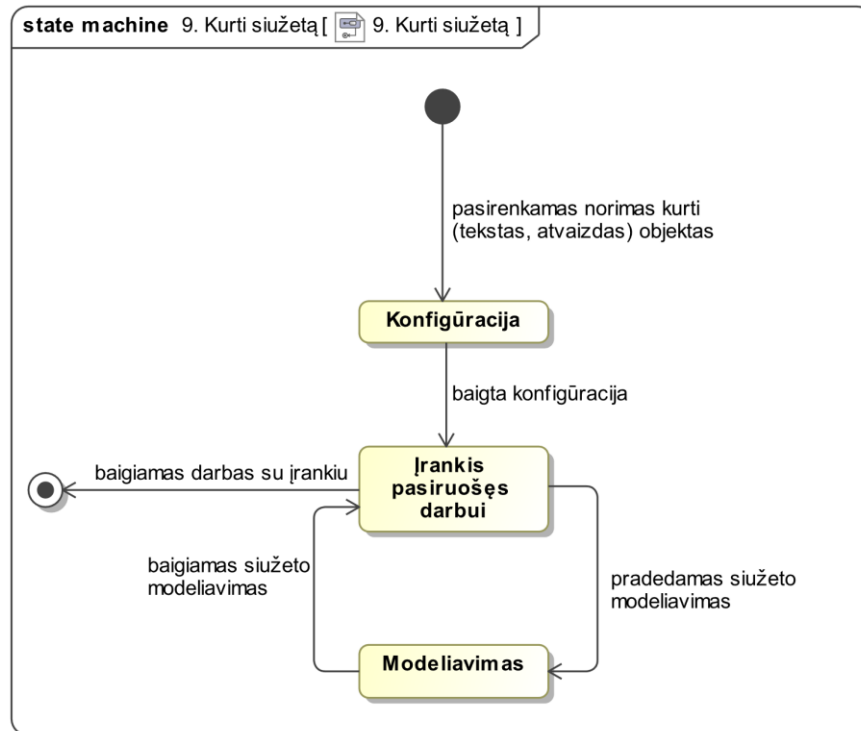
18 pav. Panaudojimo atvejo Nr. 6 būsenų diagrama

Panaudojimo atvejo „Peržiūrėti sukurtus personažus“ būsenų diagrama pavaizduota 19 pav. Kai naudotojas dar nėra sukūręs nei vieno personažo, sąrašas yra tuščias. Kai naudotojas yra sukūręs bent vieną personažą, tačiau neturi nupiešęs eskizo, rodomas klaidos pranešimas. Kai naudotojas yra sukūręs ir nupiešęs bent vieną personažo eskizą, rodomas pilnas sąrašas.



19 pav. Panaudojimo atvejo Nr. 7 būsenų diagrama

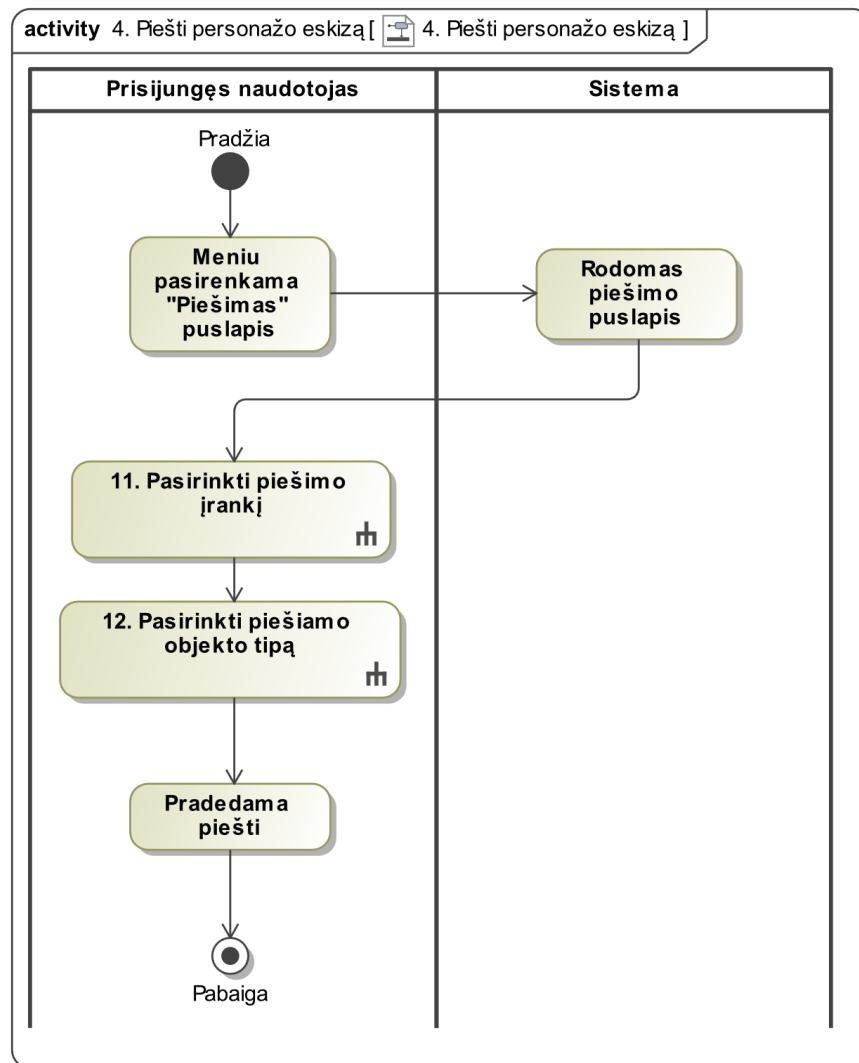
Panaudojimo atvejo „Kurti siužetą“ būsenų diagrama pavaizduota 20 pav. Siužeto kūrimas panašus į personažo eskizo piešimą. Naudotojas, atlikdamas siužeto modeliavimą, sistemą nustato į modeliavimo būseną. Baigus modeliavimą sistema sugrįžta į laukimo būseną.



20 pav. Panaudojimo atvejo Nr. 9 būsenų diagrama

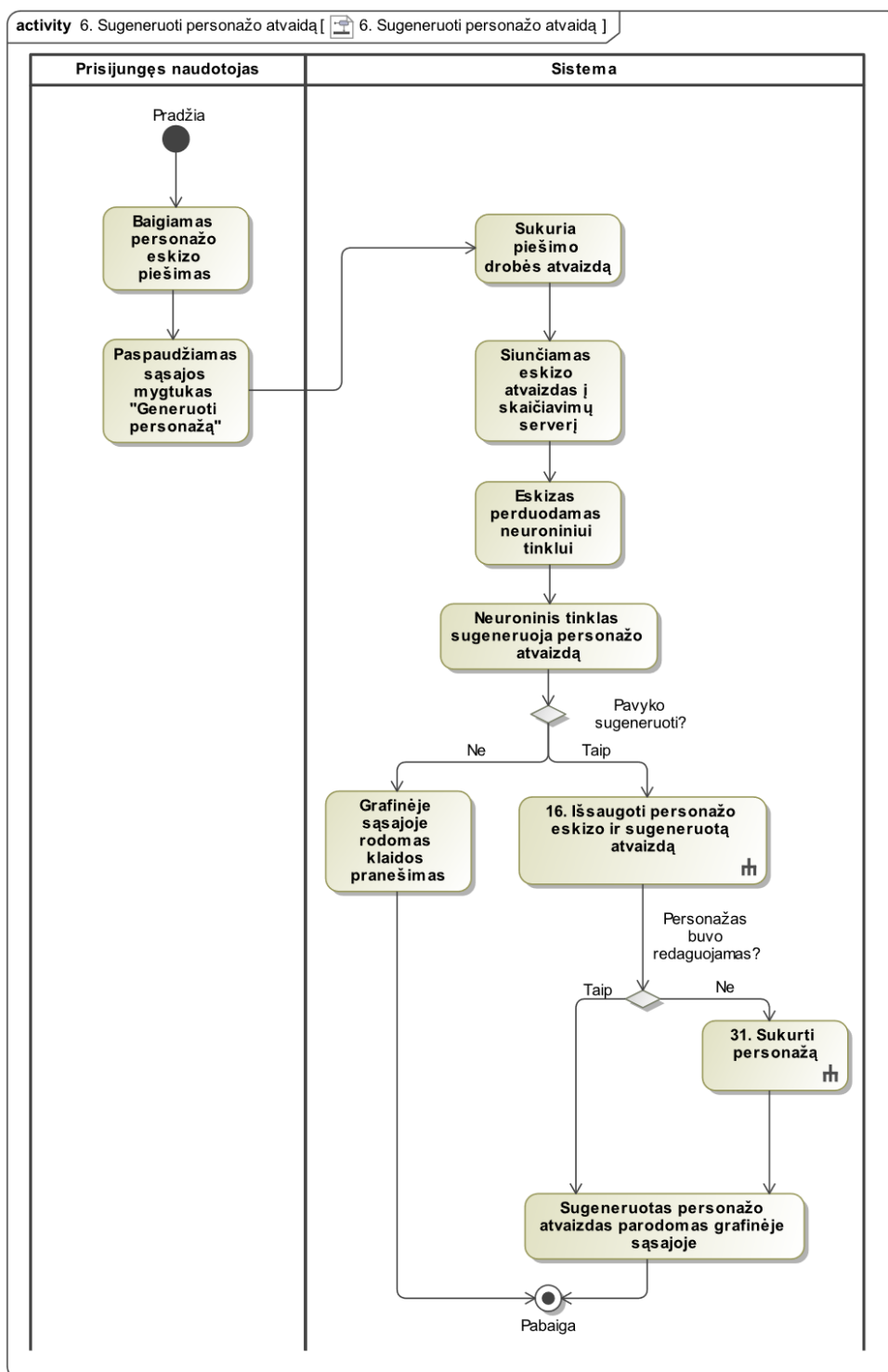
3.4.2. Veiklos diagramos

Panaudojimo atvejo „Piešti personažo eskizą“ veiklos diagrama pavaizduota 21 pav. Naudotojas, meniu pasirinkęs piešimo puslapį yra nuvedamas į jį. Norėdamas piešti eskizą, naudotojas turi pasirinkti piešimo įrankį ir objekto tipą.



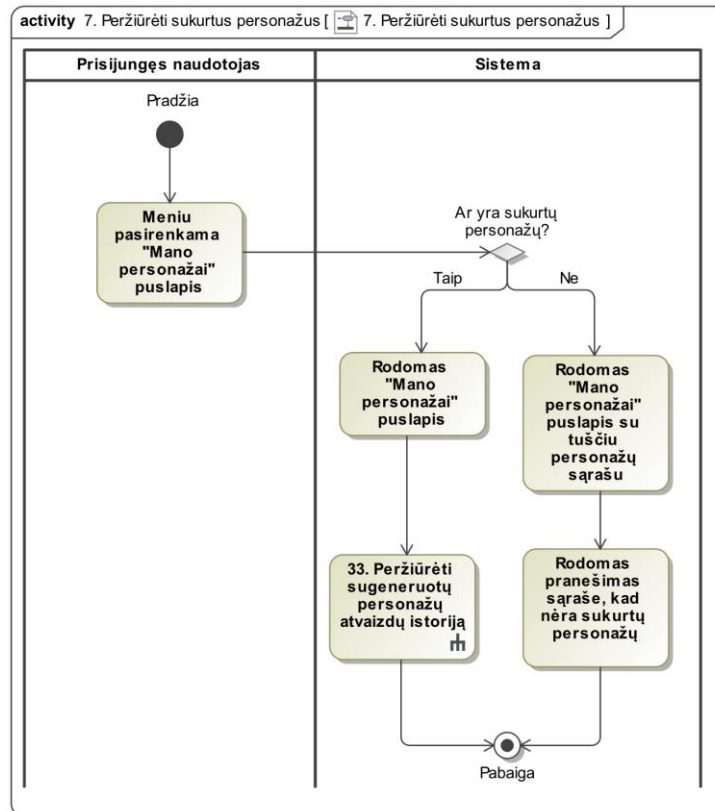
21 pav. Panaudojimo atvejo Nr. 4 veiklos diagrama

Panaudojimo atvejo „Sugeneruoti personažo atvaizdą“ veiklos diagrama pavaizduota 22 pav. Baigus personažo eskizo piešimą ir paspaudus mygtuką „Generuoti personažą“, sistema sukuria piešimo drobės atvaizdą, nusiunčia jį į skaičiavimų serverį, kuriame nupieštas eskizas perduodamas į neuroninį tinklą, kad šis atliktų personažo idėjos generavimą. Sėkmingai sugeneravus atvaizdas išsaugojamas saugykloje. Sukuriamas naujas personažas, jei nebuvo generavimo metu pasirinktas esamas, o grafinėje sąsajoje parodomas sugeneruoto personažo atvaizdas. Nesėkmingai sugeneravus rodomas klaidos pranešimas.



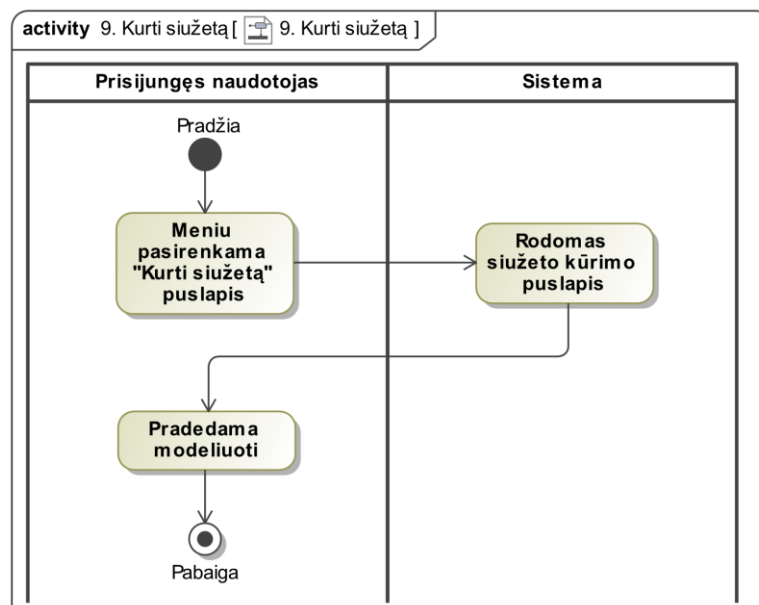
22 pav. Panaudojimo atvejo Nr. 6 veiklos diagrama

Panaudojimo atvejo „Peržiūrėti sukurtus personažus“ veiklos diagrama pavaizduota 23 pav. Meniu pasirinktus „Mano personažai“ puslapį, sistema įkelia pasirinktą puslapį. Jei dar nėra sukurtų personažų, rodomas tuščias puslapis, priešingu atveju rodomas sąrašas.



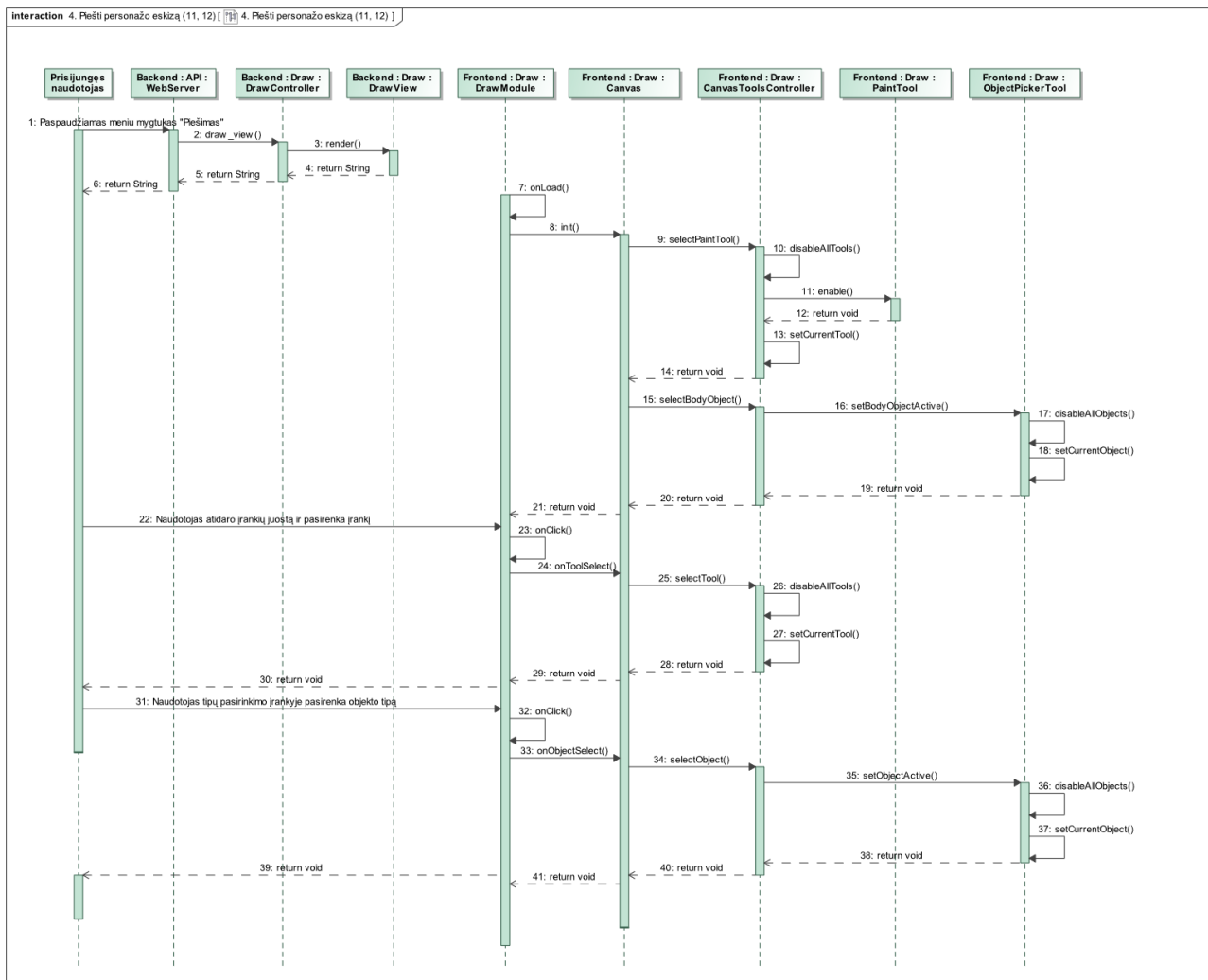
23 pav. Panaudojimo atvejo Nr. 7 veiklos diagrama

Panaudojimo atvejo „Kurti siužetą“ veiklos diagrama pavaizduota 24 pav. Kuriant siužetą, naudotojas gali modeliuoti siužeto lentą iš esamų objektų: tekstas ir sugeneruotas personažo atvaizdas. Šiuos objektus naudotojas gali vartyti, pasukti ir perkelti. Teksto objektą naudotojas gali redaguoti. Taip pat naudotojas gali pasirinkti siužeto tinklelio išdėstymą.



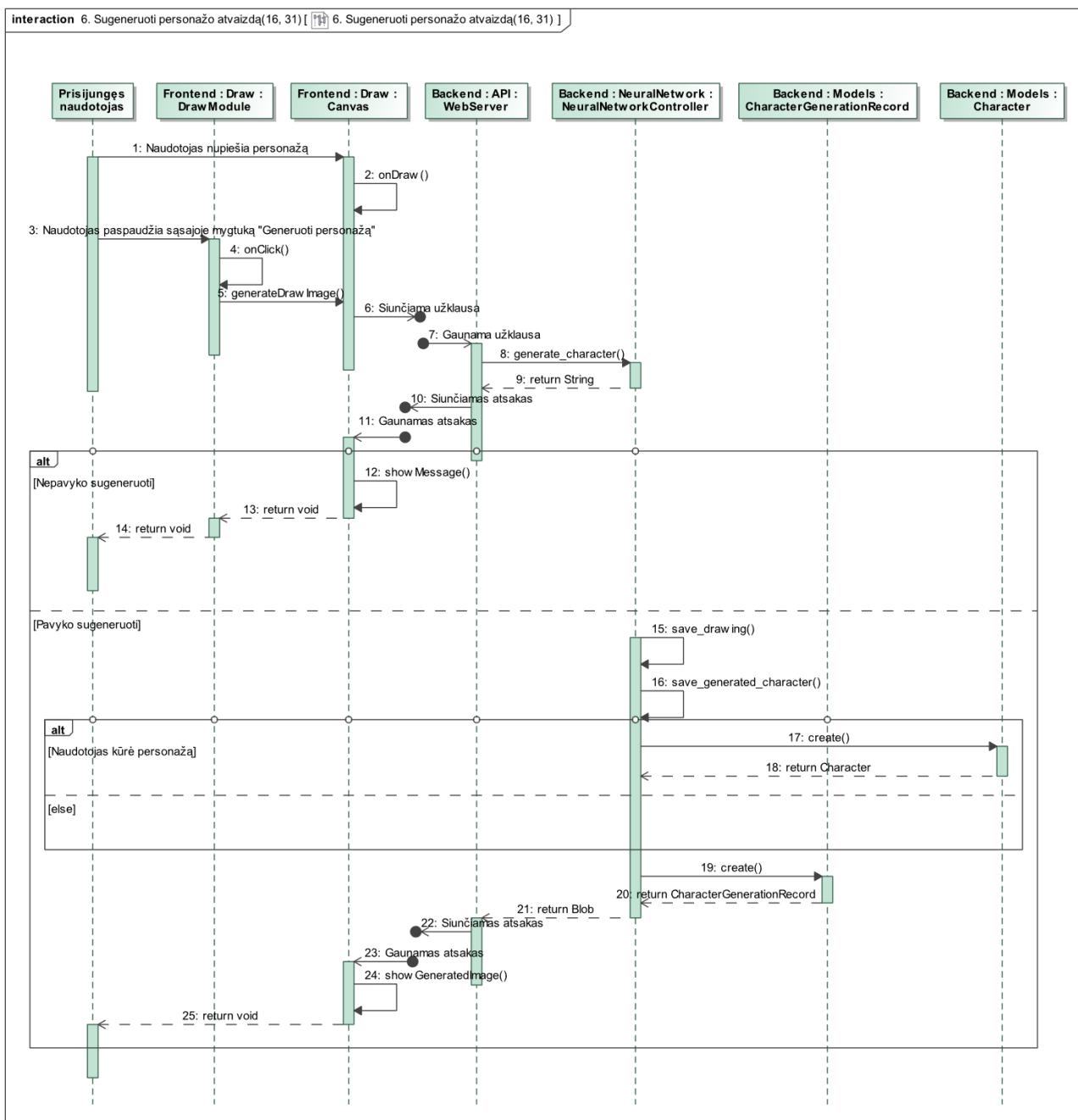
24 pav. Panaudojimo atvejo Nr. 9 veiklos diagrama

3.4.3. Sekų diagramos



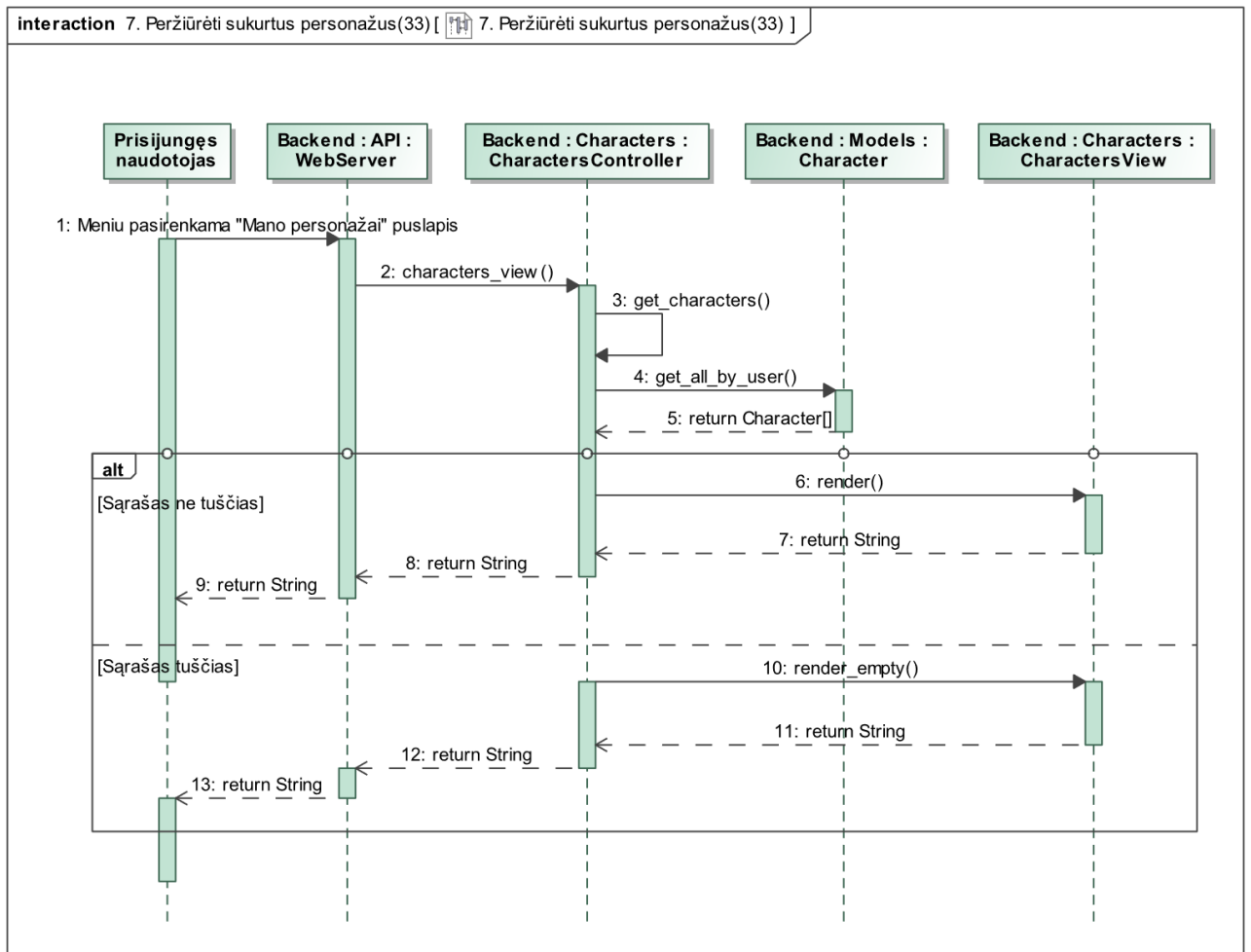
25 pav. Panaudojimo atvejo Nr. 4 sekų diagrama

Panaudojimo atvejo „Piešti personažo eskizą“ sekų diagrama pavaizduota 25 pav. Šioje sekų diagramoje vaizduojamas procesas, kuris apibrėžia personažo eskizo piešimą.



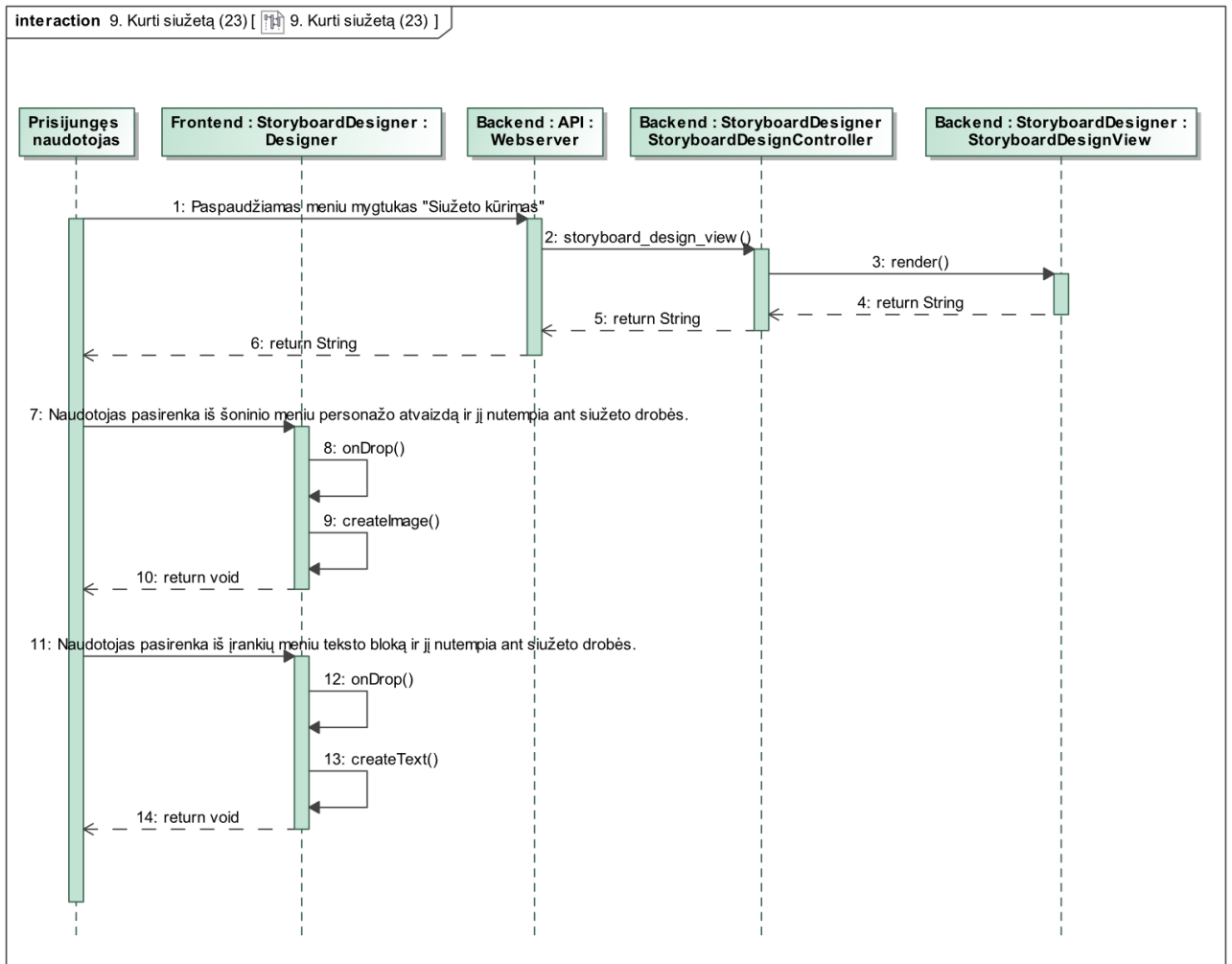
26 pav. Panaudojimo atvejo Nr. 6 sekų diagrama

Panaudojimo atvejo „Sugeneruoti personažo atvaizdą“ sekų diagrama pavaizduota 26 pav. Šioje sekų diagramoje vaizduojamas procesas atsakingas už animacinio personažo atvaizdo generavimą.



27 pav. Panaudojimo atvejo Nr. 7 sekų diagrama

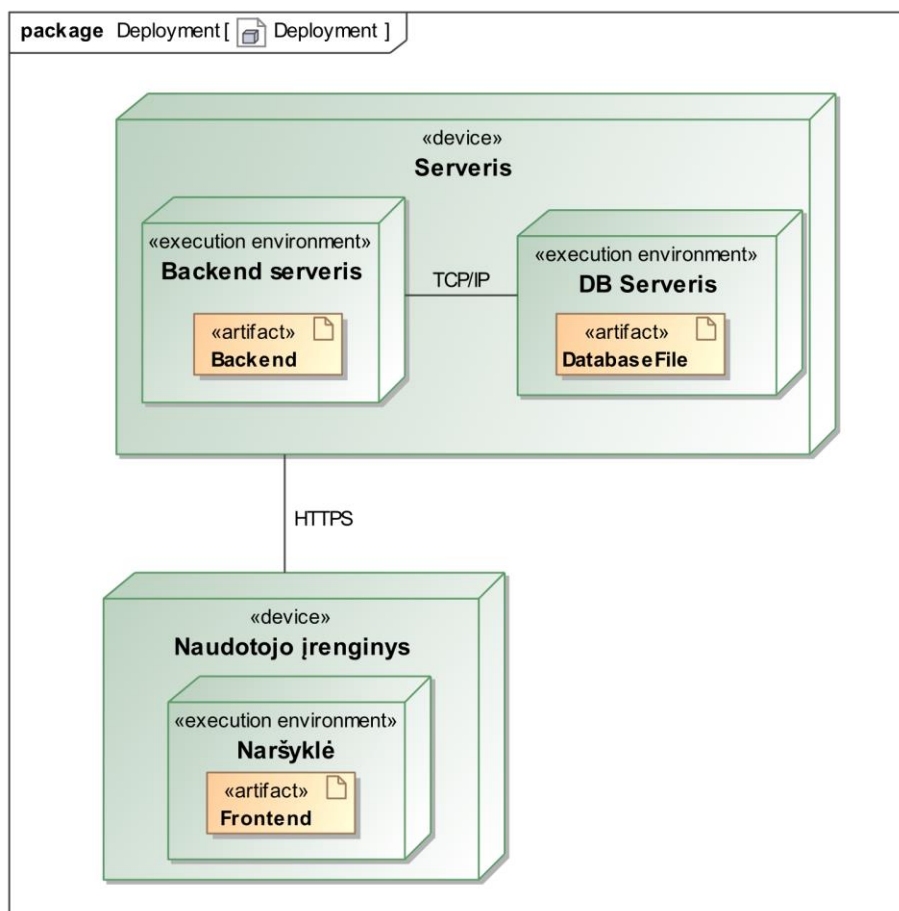
Panaudojimo atvejo „Peržiūrėti sukurtus personažus“ sekų diagrama pavaizduota 27 pav. Šioje sekų diagramoje vaizduojamas procesas atsakingas už sukurtų personažų istorijos puslapio peržiūrą.



28 pav. Panaudojimo atvejo Nr. 9 sekų diagrama

Panaudojimo atvejo „Kurti siužetą“ sekų diagrama pavaizduota 28 pav. Ši sekų diagrama vaizduoja procesą atsakingą už siužeto lentos modeliavimą.

3.5. Išdėstymo vaizdas



29 pav. Diegimo diagrama – fizinės architektūros schema

Fizinis sistemos išdėstymas pavaizduotas 29 pav. Sistemoje yra dviejų rūšių fiziniai mazgai:

1. *Serveris* – užsakovo infrastruktūros vienetas, kuris leidžia vykdyti numatytą programinės įrangos funkcionalumą;
2. *Naudotojo įrenginys* – naudotojo kompiuteris, mob. telefonas ar kitas įrenginys, kuris turi modernią naršyklę.

3.6. Duomenų vaizdas

Duomenų vaizdas pateikiamas 12 pav.

4. Tyrimo dalis

Šioje dalyje aprašoma kaip buvo vertinamas sukurtos sistemos atitikimas reikalavimams, vykdomas testavimas ir kokie patobulinimai galėtų būti realizuoti atliekant eksperimentus.

4.1. Tyrimo aplinka

Tyrimo aplinkų konfigūracijos aprašomos 11 lentelėje. Antroji konfigūracija naudojama eksperimentiniams tyrimams atlikti. Likusios konfigūracijos buvo naudojamos sistemos grafinės sąsajos, bendro veikimo, logikos tyrinėjamas ir testavimui.

11 lentelė Tyrimo aplinkų konfigūracijos

Kompiuterio pavadinimas	Procesorius	Darbinės atminties kiekis	Vaizdo procesorius	Išorinės atminties kiekis	Operacinė sistema
HP OMEN 15-ce001na	I5-7300HQ 2.5GHz	8GB	Nvidia GeForce GTX 1050	128GB SSD 1TB HDD	Windows 10
Rinktinis	Ryzen 9 5900X 3.7GHz	32GB	Nvidia GeForce RTX 3090	1TB SSD 2 TB SSD	Windows 10
Samsung Galaxy S20	Octa-core (2x2.73 GHz Mongoose M5 & 2x2.50 GHz Cortex-A76 & 4x2.0 GHz Cortex-A55)	8GB	Mali-G77 MP11	128GB	Android 11, One UI 3.0
Huawei P20 Lite	Octa-core (4x2.36 GHz Cortex-A53 & 4x1.7 GHz Cortex-A53)	4GB	Mali-T830 MP2	64GB	Android 9.0 (Pie), EMUI 9.0

4.2. Uždaviniai

Tyrimo metu buvo išskelti tokie uždaviniai:

1. įvertinti sukurtos sistemos funkcionalumo, logikos ir realizacijos teisingumą;
2. įvertinti sukurtos sistemos atitikimą reikalavimų specifikacijai;
3. įvertinti generuojamų duomenų kokybę kiekybiškai ir kokybiškai;
4. pasiūlyti sprendimus, kurie leistų patobulinti sistemos generuojamų duomenų kokybę.

4.3. Funkcionalumo, logikos ir realizacijos vertinimas

4.3.1. Strategijos

Strategijos, kurios buvo naudojamos aptikti klaidas funkcionavime, logikoje ir realizacijoje:

1. komponentų testavimas;
2. komponentų integracijos testavimas;
3. grafinės sąsajos testavimas;
4. konfigūracijos testavimas;

5. saugumo testavimas;
6. priėmimo testavimas.

Komponentų testavimas

Komponentų testavimo metu atliekamas modulių ir jų funkcijų, kurios nereikalauja integracijos su kitais sisteminiiais moduliais, testavimas. Tai užtikrina individualių funkcijų teisingumą ir stabilumą.

Komponentų integracijos testavimas

Komponentų integracijos testavimas taikomas moduliams ir jų funkcijoms, kurios reikalauja integracijos su kitais sisteminiiais moduliais, pavyzdžiui, duomenų baze arba grafine sąsaja. Tikrinama ar integruoti komponentai veikia sistemoje teisingai.

Grafinės sąsajos testavimas

Grafinės sąsajos testavimas naudojamas tam, kad imituotume natūralius naudotojo veiksmus su grafine sąsaja ir pastebėtume funkcionalumo klaidas, pavyzdžiui, neveikiantis mygtukas arba neatsirandantis iššokantis langas.

Konfigūracijos testavimas

Konfigūracijos testavimas naudojamas patikrinti sistemos veikimą skirtingose įrenginių naršyklėse. Sistema kurta modernioms naršyklėms, todėl testuojama šiose aplinkose:

- asmeninis kompiuteris;
 - *Firefox*;
 - *Chrome*;
 - *Opera*;
- mobilusis telefonas:
 - *Chrome for Android*;
 - *Firefox*;
 - *Samsung Internet*.

Saugumo testavimas

Saugumo testavimas naudojamas įsitikinti, kad sistemos ar naudotojų ištekčiai nepasiekiami tretiesiems asmenims ir ar sistema yra atspari neprasmingų duomenų srautams.

Priėmimo testavimas

Priėmimo testavimas patikrina ar sukurta sistema atitinka užsakovo iškeltus reikalavimus ir gali būti įdiegta. Testavimo metu dalyvauja užsakovas, kuris patikrina sistemos funkcinį ir nefunkcinį atitikimą reikalavimų specifikacijai vykdydamas rankinius ir automatinius testus.

4.3.2. Rezultatai

Komponentų ir jų integracijos testavimas

Testavimo ir kūrimo metu buvo kuriami automatiniai testai, kurie padengtų kuo daugiau sisteminio kodo ir taip sumažintų defektų kiekį sistemoje. Buvo naudojama testavimu grindžiama programavimo (angl. *Test Driven Development*) metodika, kuri reikalauja pirma realizuoti testus pagal kuriuos vėliau

rašomas programinis kodas. Testavimo metu visos rastos klaidos buvo ištaisytos ir užtikrintas aukštas sisteminio kodo (serverio sluoksnio) padengimas testais – 94 procentai.

Grafinės sąsajos testavimas

Grafinę sąsają buvo planuojama testuoti su automatiniais testais, tačiau testavimo eigoje nuspręsta grafinę sąsają testuoti tik su rankiniais testais. Testavimo metu tikrinami panaudojimo atvejai pagal suplanuotus testų scenarijus. Taip nuspręsta dėl aukšto serverio sluoksnio kodo padengimo testais procento. Rankinio testavimo metu defektų neaptikta.

Konfigūracijos testavimas

Konfigūracijos testavimo metu sistema buvo bandoma įvairiose aplinkose – skirtingose naršyklėse, operacinėse sistemose. Kadangi programos kodas buvo rašomas laikantis *EcmaScript 6* standarto, funkcionalumas išliko nepakitęs. Tačiau mažesnio dydžio ekranuose tam tikri elementai buvo atvaizduojami neteisingai – teksto ir piktogramų dydžiai nekisdavo, aptiktas netinkamas konteinerių išdėstymas mažo dydžio ekranuose. Visos aptiktos klaidos ištaisytos.

Saugumo testavimas

Didžioji dalis saugumo testavimo buvo įtraukta į integracijos testus, kuriuose tikrinama esamo naudotojo pasiekiami ištekliai. Buvo rastas vienas atvejis, kai kiti naudotojai galėjo matyti ne savo išteklius – sukurtus personažus. Kiekvieno naudotojo sąrašė buvo rodomi visi sistemoje sukurti personažai, o ne tik tie, kurie priklauso esamam naudotojui. Taip pat buvo bandoma įterpti neprasmingus duomenis į formas ir stebima sistemos reakcija. Visos aptiktos klaidos ištaisytos.

Priėmimo testavimas

Priėmimo testavimo metu išbandyti visi panaudojimo atvejai ir testų scenarijai, įvykdyti parašyti automatiniai testai. Rastos dvi klaidos, kurios vėliau įtrauktos į *Github* klaidų archyvą.

Pirmoji klaida susijusi su neteisingu statistikos skaičiavimu – į statistiką buvo įtraukiami visi sugeneruoti personažai, o ne tik tie, kurie yra aktyvūs (neištrinti).

Antroji klaida susijusi su scenarijaus kūrimo įrankiu – programinis kodas netikrino, koks scenarijaus tinklelio išdėstymo stilius buvo naudojamas, todėl leido bet kokios teksto eilutės pasirinkimą, kai tuo tarpu turėjo leisti tik iš anksto žinomą aibę tinklelio parinkčių. Visos aptiktos klaidos ištaisytos.

4.4. Sistemos atitikimas reikalavimų specifikacijai

4.4.1. Strategijos

Sistemos atitikimas reikalavimų specifikacijai yra testuojamas su kokybės vertinimo apklausa (1 priedas, 2 priedas) ir priėmimo testavimo metu, kuomet yra išbandomi visi panaudojimo atvejai bei kiti funkciniai ir nefunkciniai reikalavimai. Jei priėmimo testavimo metu neiškilo problemų, nerasta klaidų yra vertinama, kad programų sistema atitinka reikalavimų specifikaciją.

Apklausa daugiausiai kreipė dėmesį į ISO 25010 parametrų patikrinimą atsižvelgiant į skirtingų naudotojų patirtį. Įvertinti parametrai:

- funkcionalumas – ar pakankamas funkcionalumas ir išpildyti reikalavimai;
- efektyvumas – ar pakankamas sistemos atsako laikas ir personažų generavimo greitis;

- panaudojamumas – ar sistemos naudojimas draugiškas naudotojų atžvilgiu ir ar sistemą paprasta naudoti skirtinguose įrenginiuose;
- patikimumas – ar sistema dažnai patiria gedimus ir trūkumus, kurie randami naudotojų;
- saugumas – ar pakankamas naudotojų autentifikavimo ir autorizacijos įrankių naudojimas;
- palaikomumas – ar lengva sistemą plėsti, pridėti naujo funkcionalumo ir taisyti esamą;
- perkeliamumas – pastangos reikalingos perkelti sistemą į kitą aplinką (serverį) ir reikalavimų aplinkai specifiškumas.

Apklausa skirta dviejų tipų naudotojams:

- animatoriai ir vaikai;
- projekto kūrimo ir palaikymo komandos nariai.

Apklausa skirtingiems naudotojams skiriasi – animatorių ir skirtingų amžiaus grupių asmenų anketa paprastesnė, apima tik dalį ISO 25010 standarto parametrų, o kūrimo komandos apklausa apima visus išvardintus.

4.4.2. Rezultatai

Priėmimo testavimas

Visos klaidos rastos priėmimo testavimo metu buvo ištaisytos. Sistema įdiegta užsakovo diegimo aplinkoje 59 pav.

Kokybės vertinimo apklausa

Apklausoje buvo pateikiama po kelis skirtingus klausimus reikalingus įvertinti vieną parametą. Įverčiai buvo skalėje nuo 1 iki 5, kai 1 – teiginys/klausimas visiškai netenkinamas ir 5 – teiginys/klausimas visiškai tenkinamas. Žemiau esančioje 12 lentelėje galime matyti kiekvieno parametro įvertinimą:

12 lentelė ISO 25010 standarto parametrų įvertinimas

Parametras	Rezultatas
Funkcionalumas	4,79
Efektyvumas	4,64
Panaudojamumas	4,56
Patikimumas	4,86
Saugumas	4,79
Palaikomumas	4,5
Perkeliamumas	3,5

Iš rezultatų matoma, kad prasčiausiai įvertintas yra perkeliamumas. Taip yra todėl, kad sistema reikalauja specifinių techninės ir programinės įrangos reikalavimų, pavyzdžiui, sistema privalo turėti *Nvidia* kompanijos grafinį procesorių, o operacinė sistema privalo būti *Linux*. Apibendrinus rezultatus, galima daryti išvadą, kad sistema yra sukurta kokybiškai ir atitinka reikalavimų specifikaciją.

4.5. Generuojamų duomenų kokybės vertinimas

4.5.1. Strategijos

Norint patikrinti ar generuojami duomenys yra semantiškai korektiški ir kokybiški, tenka vadovautis subjektyvia naudotojų nuomone ir skaitinėmis metrikomis, todėl neuroninis tinklas, kuris generuoja duomenis, buvo vertinamas pagal apklausos rezultatus ir skaitines metrikas, kurios aprašomos 1.5 poskyryje. Kokybiniam įvertinimui pasitelkiamos naudotojų apklausos (1 priedas, 2 priedas) kokybės įvertinimo klausimas, kuriame reikia įvertinti kelis generuojamų animacinių personažų atvaizdų aspektus.

4.5.2. Rezultatai

Tyrimo rezultatai pateikiami 5.4 poskyryje, kuriame jie lyginami su eksperimento rezultatais.

4.6. Eksperimentiniams tyrimams siūlomi sprendimai

Remiantis sukurto sprendimo kokybės vertinimo išvadomis, didžiausia problema yra generuojamų duomenų kokybė. Tiek skaitinis, tiek kokybinis įvertinimas leidžia suprasti, kad pagrindinė sistemos funkcija – animacinių personažų atvaizdų generavimas veikia ne taip, kaip buvo tikimasi specifikuojant sistemą. Todėl tyrinėjimai bus skirti generuojamų duomenų kokybės gerinimui.

4.6.1. Duomenų rinkinys

Dažnai didelę įtaką generuojamų duomenų kokybei daro tai, kaip kokybiškai yra paruoštas duomenų rinkinys [48, 56]. Tai žinoma iš bendrosios praktikos dirbant su neuroniniais tinklais. Pavyzdžiui, vienas iš dažniausiai taikomų metodų yra normalizacija – tai duomenų dimensiškumo suvienodinimas. Jei turime skaičių vektorius, kurie stipriai skiriasi savo reikšmių skalėmis, juos būtina normalizuoti. Įprasta praktika dirbant su atvaizdais – visas pikselių intensyvumo reikšmes užkoduoti iš 0..255 skalės į 0..1. Tai sumažina neuroninio tinklo jautrumą realių duomenų pokyčiams.

Generatyviniams neuroniniams tinklams ir ne tik yra svarbu, kad duomenų pasiskirstymas sektų Gauso arba normalųjį skirstinį, kitaip sakant, kad duomenys turėtų panašias savybes. Pavyzdžiui, negalime apmokyti neuroninį tinklą generuoti kačių atvaizdų, jei duomenų rinkinį sudarys šunų atvaizdai. Tą patį principą galime taikyti ir animacinių personažų generavime – tinklas neišmoks generuoti aukštos kokybės vienos rūšies personažų atvaizdų, jei duomenų rinkinyje bus kitos rūšies. Pavyzdžiui, jei bandytume išmokyti neuroninį tinklą generuoti *Kempiniukus Plačiakelnius*, bet mūsų rinkinyje būtų tik *Simpsonų* personažai. Šių personažų stiliai ir savybės yra visiškai skirtingos, todėl gautume nekokybiškus ir dažnai nesuprantamus atvaizdus. Taip nutiko ir su projekto realizacijos metu sukurtu sprendimu. Apmokyti neuroninį tinklą buvo naudojamas nekokybiškas duomenų rinkinys, kuris buvo sudarytas per trumpą laiką dėl nepakankamų laiko išteklių. Jame buvo daug skirtingų rūšių personažų.

Kitas dažnai naudojamas generuojamų duomenų kokybei gerinti metodas – augmentacija. Tai nežymios duomenų rinkinio elementų modifikacijos, kurios leidžia praplėsti duomenų rinkinį ir taip sumažinti persimokymą (angl. *overfitting*). Metodas suteikia neuroniniam tinklui pakankamai daug skirtingų siekiamos išmokti aibės elementų pavyzdžių. Atvaizdų generavimo uždaviniuose taikomos įvairios augmentacijos: pasukimas, apvertimas, šviesinimas, tamsinimas ir kt. [42]. Augmentacijų taikymas leidžia praplėsti duomenų rinkinį.

Remiantis šia apie duomenų rinkinius žinoma informacija, galime iškelti hipotezę, kurią bus siekiama patvirtinti arba paneigti eksperimentinėje dalyje:

H1: Generuojamų duomenų kokybė priklauso nuo duomenų rinkinio kokybės.

4.6.2. Neuroninio tinklo architektūra

Siekiant pagerinti generuojamų duomenų kokybę svarbu pasirinkti tinkamą neuroninio tinklo architektūrą. Norint sukurti architektūrą, reikia žinoti pagrindinius sudedamuosius sluoksnius. Atvaizdų generavime sluoksniai yra tokie:

- pilnai sujungtas sluoksnis (angl. *Fully Connected*);

Šio tipo sluoksniai dažniausiai naudojami tinklo išvestyje, pavyzdžiui, konvoliuciniuose tinkluose kada atliekamas klasifikavimas. Tai pilnai sujungti neuroniniai tinklai su nurodytu kiekiu sluoksnių. Paskutiniame tokio tinklo sluoksnyje būna tiek neuronų, kiek skirtingų objekto klasių klasifikatorius bando atpažinti.

- konvoliucijos sluoksnis (angl. *Convolution*);

Konvoliucijos operacijos leidžia pritaikyti įvairius filtrus atvaizdams, kurie išskiria tam tikras savybes, pavyzdžiui, objektų kraštines. Tokių skirtingų paeiliui sujungtų filtrų sluoksnių derinys leidžia išrinkti svarbiausias savybes iš įvestyje pateikto atvaizdo.

- sujungimo sluoksnis (angl. *Pooling*);

Sujungimo sluoksniai leidžia sumažinti savybių vektorių dydį mažinant šio dimensiskumą, atliekant maksimalaus arba vidutinio sujungimo operaciją. Tai leidžia išsaugoti savybių kiekį, tačiau sumažinti atvaizdo dydį.

- išmetimo sluoksnis (angl. *Dropout*);

Išmetimo sluoksniai modifikuoja savybių vektorių taip, kad atsitiktine tvarka norimas procentas neuronų turėtų nulinę reikšmę, tai yra, aktyvacijos metu nedarytų įtakos skaičiavimams. Buvo įrodyta, kad šių sluoksnių naudojimas stabdo persimokymą [57].

- normalizavimo sluoksnis (angl. *Normalization*);

Normalizavimo sluoksnis atlieka savybių vektorių normalizavimą. Šie sluoksniai dažnai naudojami po konvoliucijos sluoksnių, kad sugrąžintų filtrų modifikuotą savybių vektorių į normalizuotą formą. Sluoksniai taip pat prisideda prie persimokymo stabdymo [42].

- aktyvacijos sluoksnis (angl. *Activation*).

Aktyvacijos iš esmės apibrėžia, ką neuroninis tinklas mokosi. Kuriant architektūrą, tai vienas iš svarbiausių sluoksnių, nes skirtingų aktyvacijos funkcijų naudojimas gali lemti skirtingą rezultatą. Viena iš dažniausiai naudojamų aktyvacijos funkcijų paslėptiems neuroninio tinklo sluoksniams yra *RELU* arba *Leaky RELU*, tačiau yra ir kitų svarbių, pavyzdžiui *tanh* arba *sigmoid*. Aktyvacijos funkcijos sprendžia, kuriuos neuronus aktyvuoti, kai šių reikšmės kinta. Pavyzdžiui, *RELU* funkcija neleidžia neigiamų reikšmių – neuronas su neigiama reikšme bus inaktyvuotas, o jo reikšmė bus 0. *tanh* funkcija – pritaiko matematinę tangento funkciją, *sigmoid* – visas aukštas reikšmes padaro

artimas 1, o žemas artimas 0. Pastaroji dažniausiai naudojama paskutiniuose klasifikatorių sluoksniuose.

Sluoksnių kombinacijos leidžia sudaryti tinklo architektūras (modelius), kuriuos vėliau galima išmokyti generuoti atvaizdus. Eksperimentinis modelio tyrimas gali leisti patikrinti ar sukurta architektūra yra naudinga ir efektyvi. Rinkoje yra daug skirtingų neuroninių tinklų architektūrų, kurios viena už kitą pranašesnės. Sukurto sprendimo atžvilgiu yra kelios skirtingų dydžių architektūros, kurios geba generuoti aukštos kokybės atvaizdus iš semantinių objektų žemėlapių:

- *SPADE* [28];
- *DAGAN* [58];
- *OASIS* [29];
- *CLADE* [59];
- *INADE* [60];
- *SCGAN* [61];
- *FewShot-SMIS* [26];
- *SuperStyleNet* [27].

Teoriškai [62] neuroninis tinklas turintis didesnę neuronų ir sluoksnių kiekį, turėtų gebėti išmokyti daugiau savybių dėl didesnio svorių (angl. *weights*) kiekio, kurie kontroliuoja neuroninio tinklo spėjimus. Remiantis šia teorija galime iškelti hipotezę, kurią bus siekiama patvirtinti arba paneigti eksperimentinėje dalyje:

H2: Tinklo dydis daro tiesioginę įtaką generuojamų duomenų kokybei.

Paminėtos tinklo architektūros dėl savo naujumo ir autorių teigiamų rezultatų bus naudojamos patikrinti abi iškeltas hipotezes.

5. Eksperimento dalis

5.1. Hipotezės

H1: Generuojamų duomenų kokybė priklauso nuo duomenų rinkinio kokybės.

H2: Tinklo dydis daro tiesioginę įtaką generuojamų duomenų kokybei.

Pirmoji hipotezė suformuota remiantis 4.6.1 skyrelyje pateikta informacija. Antroji hipotezė suformuota remiantis 4.6.2 skyrelyje pateikta informacija.

5.2. Hipotezių tikrinimas

Patikrinti pirmajai hipotezei yra sudaromas kokybiškas duomenų rinkinys iš tos pačios rūšies personažų. Šie personažai atvaizduose yra sulygiuojami toje pačioje vietoje – per centrą. Visi besidubliuojantys personažai pašalinami. Gaunamas kokybiškas duomenų rinkinys. Sudarytas duomenų rinkinys naudojamas 8 neuroniniams tinklams, kurie buvo paminėti 4.6.2 skyrelyje, mokyti. Išmokyti neuroniniai tinklai testuojami su validacijos duomenų rinkiniu. Surenkami metrikų paminėtų 1.5 poskyryje, rezultatai. Mokomi tie patys neuroniniai tinklai su senuoju duomenų rinkiniu. Atliekami tie patys testavimo ir rezultatų surinkimo veiksmai. Gautos metrikos lyginamos su naujuoju duomenų rinkiniu. Geriausiai naujojo duomenų rinkinio metrikų rezultatus surinkęs neuroninis tinklas naudojamas atlikti kokybinę analizę, pakartojant naudotojų apklausą, kuri buvo paminėta 4 skyriuje. Seniau atliktos naudotojų apklausos rezultatai lyginami su naujosios.

Antrajai hipotezei patikrinti naudojamas sudarytas duomenų rinkinys ir pasirinkti neuroniniai tinklai, kurie yra skirtingo dydžio – turi skirtingą kiekį svorių. Lyginamas neuroninio tinklo dydis ir surinkti metrikų rezultatai pasitelkiant statistinę koreliaciją.

5.3. Eiga

5.3.1. Duomenų rinkinio sudarymas

Abiejų hipotezių patikrinimui reikia sudaryti duomenų rinkinį, todėl reikia suprasti esamo trūkumus.

5.3.1.1. Esamas duomenų rinkinys

Kuriant sistemos prototipą, dėl laiko apribojimų buvo greitai sudarytas duomenų rinkinys ignoruojant kokybės užtikrinimą. Sudarant rinkinį buvo surenkami animacinių personažų atvaizdai iš skirtingų svetainių: *Flickr* [63], *Pixabay* [64], *Pinterest* [65]. Surinkta apie 3 tūkstančiai atvaizdų, kurie atrinkti ranka pagal savo vizualų prasmingumą ir naudingumą. Pavyzdžiui, personažai, kurie buvo labai maži atvaizde yra neįtraukiami į galutinį rinkinį, nes atlikus duomenų apdorojimą – sumažinus iki tam tikro nustatyto dydžio dažniausiai 256 prie 256, personažo atvaizdas sumažėja dar labiau. Tokio personažo semantinio žemėlapio didžioji dauguma pikselių turėtų 0 reikšmę, o tai prilygsta neefektyviam skaičiavimo išteklių naudojimui. Atrinkti personažai buvo anotuojami su *LabelBox* [66] įrankiu, o anotacijų pavyzdžiai pateikiami 30 pav. Esamame duomenų rinkinyje yra 8 semantiniai objektai:

- galva;
- kūnas;
- ranka;
- koja;

- akis;
- burna;
- ausis;
- nosis.

Kiekvienas semantinis objektas koduojamas skirtinga spalva, o atliekant semantinio objektų žemėlapiu paruošimą mokymui yra užkoduojamas iš *RGB* į *GrayScale* skalę, kur kiekvienas atspalvis skiriasi 1.



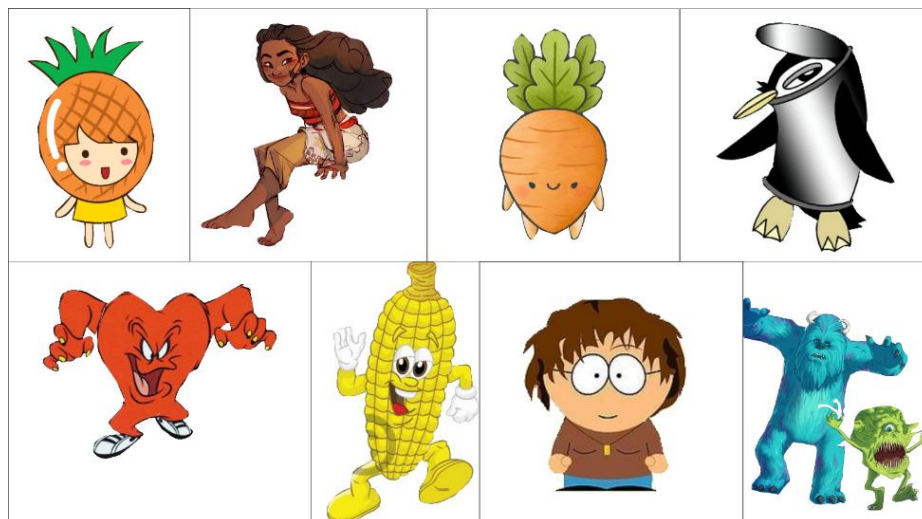
30 pav. Esamo duomenų rinkinio anotavimo pavyzdžiai

Per visą techniniame projekte numatytą laiką pavyko surinkti apie 1 tūkstantį anotuočių personažų atvaizdų. Anotavimo efektyvumo rezultatai pateikiami 13 lentelėje.

13 lentelė Esamo duomenų rinkinio sudarymo efektyvumo rezultatai

Komandos narys	Semantiniai objektų žemėlapiai	Vidutinis laikas per semantinių objektų žemėlapi	Bendras laikas
ovigri1@ktu.lt	987	3m 20s	54h 48min

Atlikus anotavimą buvo atliekamas anotuočių objektų ištraukimas iš fono. Dažnai atvaizduose yra nereikalingas triukšmas – skirtingų spalvų fonas ar tiesiog objektai, kurie nesusiję su pačio personažo kūnu, pavyzdžiui, 30 pav. palmės ar molbertas. Ištraukiant personažo atvaizdą iš fono, anotacijos buvo naudotos kaip kaukės. Anotuoti personažai išsaugomi baltame fone 31 pav.



31 pav. Esamo duomenų rinkinio pavyzdys

Kaip matome esamo duomenų rinkinio pavyzdyje 31 pav., personažai yra visiškai skirtingų stilių – skiriasi piešimo technika, tekstūrų sudėtingumas ir spalvų gamos. Vieni personažai vaizduojami 3D stiliumi, kiti 2D. Toks duomenų rinkinys yra laikomas ypač sudėtingu dėl didelio skirtingų išmokstamų savybių kiekio. Su tokiu duomenų rinkiniu yra sunku apmokyti neuroninį tinklą generuoti aukštos kokybės atvaizdus, arba reikalingas ypatingai didelis duomenų rinkinys. Viešai prieinamo duomenų rinkinio atitinkančio projekte numatyta funkcionalumą rasti nepavyko.

5.3.1.2. Naujasis duomenų rinkinys

Norint pagerinti generuojamų duomenų kokybę reikia, kad animaciniai personažai būtų panašūs savo stiliumi. Šiam aspektui įgyvendinti buvo nuspręsta surinkti *Reddit* [67] svetainės naudotojų personažus, kuriuos jie asmeniškai gali sukurti ir viešai paskelbti. Naudojant puslapio viešai prieinamą sąsają, buvo surinkti 70 tūkstančių naudotojų personažų atvaizdai. Šiuos atvaizdus reikėjo sutvarkyti: ištrinti besidubliuojančius, sulygiuoti į vidurį ir suvienodinti visų atvaizdų dydį. Besidubliuojantiems atvaizdams aptikti buvo naudojamas SHA-256 šifras atvaizdo dvejetainiams duomenims užkoduoti. Jei randamas identiškas šifras, atvaizdas yra ištrinamas. Surinktų atvaizdų dydžiai skyrėsi, todėl buvo pasirinkta naudoti vieno dydžio atvaizdus 256 prie 256. Atliekant atvaizdų lygiavimą, skaičiuojamas skirtumas tarp norimo aukščio ir pločio, ir esamo. Atvaizdas paslenkamas tuščioje drobėje per pusę skirtumo. Atlikus visus reikiamus pasirošimo veiksmus buvo gauta 51 tūkstantis unikalių personažų atvaizdų. Duomenų rinkinio pavyzdys pateikiamas 32 pav.

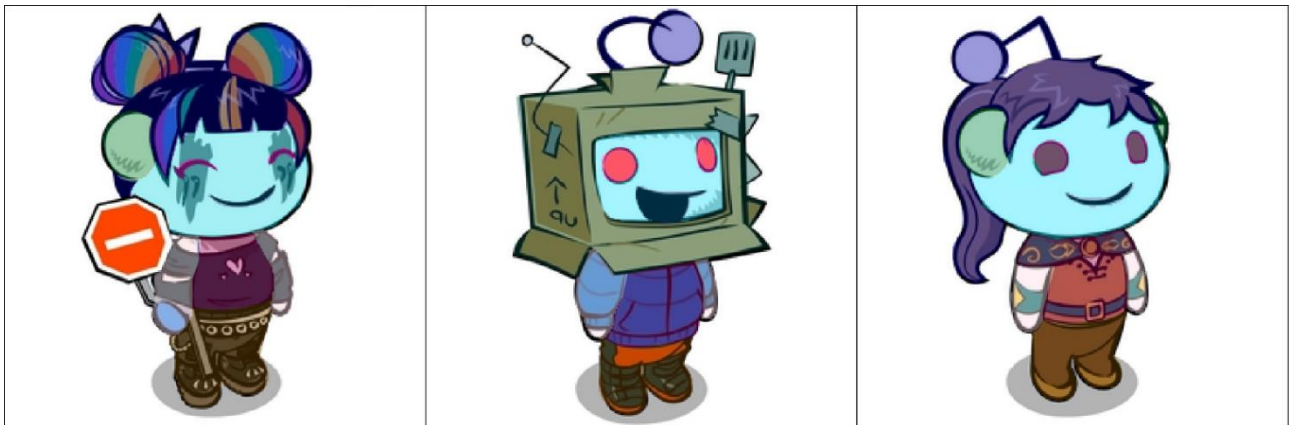


32 pav. Naujojo duomenų rinkinio pavyzdys

Kaip matome, naujasis duomenų rinkinys turi tokio pačio stiliaus animacinius personažus, todėl teoriškai turėtų būti didesnė tikimybė, kad neuroninis tinklas išmoks generuoti aukštesnės kokybės atvaizdus. Atliekant anotavimą naujasis duomenų rinkinys buvo papildytas naujais semantiniais objektais. Naujos objektų klasės:

- plaukai;
- uodega;
- antakiai;
- kepurė;
- akiniai;
- sparnai.

Šių animacinių personažų anotavimas irgi vyko *LabelBox* [66] aplinkoje, o anotavimo pavyzdžiai pateikiami 33 pav.



33 pav. Naujojo duomenų rinkinio anotavimo pavyzdžiai

Dėl naujajame duomenų rinkinyje esančių paprastesnių personažų, anotavimas užtruko trumpiau – 28 valandas. Senojo duomenų rinkinio anotavimas užtruko 54 valandas. Vidutinė anotavimo trukmė sumažėjo per pusę. Tą galime pastebėti 14 lentelėje.

Kad būtų teisingai įvertinta naujojo duomenų rinkinio kokybės įtaka generuojamiems duomenims, buvo suanotuotas panašus duomenų elementų kiekis. Visas duomenų rinkinys yra naudojamas vieno iš neuroninių tinklų [26] apmokymo procese.

14 lentelė Naujojo duomenų rinkinio sudarymo efektyvumo rezultatai

Komandos narys	Semantiniai objektų žemėlapiai	Vidutinis laikas per semantinių objektų žemėlapi	Bendras laikas
ovigri1@ktu.lt	1000	1m 42s	28h 23min

5.3.2. Neuroninių tinklų mokymas

Hipotezių testavimui buvo pasirinkti šie neuroniniai tinklai:

- *SPADE* [28];
- *DAGAN* [58];
- *OASIS* [29];
- *CLADE* [59];
- *INADE* [60];
- *SCGAN* [61];
- *FewShot-SMIS* [26];
- *SuperStyleNet* [27].

Pasirinkti tinklai yra moderniausi, dažniausiai naudojami arba naujausi ir teigia apie ypač gerus rezultatus atvaizdų generavime iš semantinių objektų žemėlapių. Neuroninių tinklų apmokymo procesą galėtume apibrėžti šiais žingsniais:

1. paruošti neuroninio tinklo apmokymo programą;

Šio žingsnio metu reikia sukurti programą, kuri įgyvendintų neuroninio tinklo architektūrą, arba pasinaudoti autorių pavišintu įgyvendinimu, kuris įdiegiamas tyrimo aplinkoje. Visų neuroninių tinklų architektūrų autoriai yra pavišinę įgyvendinimus, todėl papildomo darbo prie šio parametro atlikti neteko.

2. įdiegti reikalingus programinės įrangos modulius;

Kiekvienas neuroninio tinklo įgyvendinimas reikalauja tam tikrų programinės įrangos modulių. Visi įgyvendinimai naudojo *PyTorch* giliojo mokymo biblioteką, *NumPy*, *Scikit-learn* ir *OpenCV*. Kiekvienas įgyvendinimas dažniausiai pateikia sąrašą programinių modulių, kurie reikalingi norint išmokyti neuroninį tinklą. Nurodyti moduliai įdiegiami iš anksto paruoštoje *Python* programavimo kalbos virtualioje aplinkoje.

3. paruošti duomenų rinkinį;

Duomenų rinkinys yra vienas iš svarbiausių aspektų, lemiančių sėkmingą neuroninio tinklo mokymo procesą. Kaip buvo sudaromas duomenų rinkinys aprašyta 5.3.1.2 skyrelyje.

4. suskaidyti duomenų rinkinį į mokymo ir testavimo rinkinius;

Šiame žingsnyje sudarytas rinkinys į suskaidomas į dvi dalis – mokymo ir testavimo. Mokymo dalis dažniausiai sudaro 80 procentų visų duomenų. Atsitiktine tvarka tikri atvaizdai ir jų semantiniai žemėlapiai suskirstomi į nurodytas dalis.

5. parinkti hiperparametrus;

Šiame žingsnyje svarbu pasirinkti tam tikrus hiperparametrus, kurie daro įtaką mokymo procesui, pavyzdžiui, kiek mokymo ciklų (angl. *epoch*) arba koks mokymo greitis (angl. *learning rate*) bus naudojamas atliekant mokymą. Be šių hiperparametrų yra kitų, kurie skiriasi su kiekviena neuroninio tinklo architektūra. Todėl buvo keičiami tik tie hiperparametrai, kurie nepriklauso nuo architektūros:

- Mokymo greitis (naudojamas $2 \cdot 10^{-4}$);
- Epochų kiekis (naudojama 200);
- Optimizavimo uždavinys (naudojamas *Adam* [68] optimizavimo uždavinys, kuris yra vienas iš dažniausiai naudojamų atliekant modernių neuroninių tinklų mokymus);

6. išmokyti neuroninį tinklą;

Išmokyti neuroninį tinklą užtrunka, o tai priklauso nuo tinklo dydžio ir nuo to ar naudojamos optimizuotos matematinės operacijos. Jei naudojamos unikalios, nestandartizuotos operacijos dažnai mokymas užtrunka ilgiau.

7. atlikti išmokyto neuroninio tinklo testavimą su testavimo duomenų rinkiniu;

Kai baigiamas mokymas, būtina atlikti neuroninio tinklo testavimą su testavimo duomenimis. Tik atlikę neuroninio tinklo testavimą, galime daryti išvadas apie tai, kaip tinklas išmoko generuoti atvaizdus, nes tų duomenų mokymo procese tinklas nematė ir negalėjo įsiminti.

8. jei rezultatai nepatenkinami, kartoti nuo 5 žingsnio;

Nutinka taip, kad išmokyti nepavyksta dėl įvairių faktorių, kurie buvo aprašyti 1.4 poskyryje. Tuomet tenka keisti hiperparametrus ir atlikti mokymą iš naujo.

9. pabaiga.

Šiame žingsnyje mes turime išmokytą neuroninį tinklą, kurį galime naudoti atlikti animacinių personažų atvaizdų generavimą.

Viena [26] iš pasirinktų neuroninių tinklų architektūrų reikalavo kito, papildomo neuroninio tinklo mokymo. *Few-Shot SMIS* reikalavo, kad papildomai būtų išmokytas *STYLEGAN* [56] tipo neuroninis tinklas, kuris generuotų animacinius personažus iš triukšmo vektoriaus, o ne iš semantinio objektų žemėlapio. Šio tipo neuroniniai tinklai išmoka funkciją, kuri reaguoja į triukšmo vektoriaus pokyčius ir sugeneruoja vis kitokio stiliaus atvaizdą. Žinant šią informaciją, *Few-Shot SMIS* autoriai pasinaudojo *pSp* [69] neuroniniu tinklu, kad iš sugeneruoto atvaizdo gautų naudotą triukšmo vektorius, kurį vėliau rekonstruotų į semantinį objektų žemėlapi. Naudojant šią techniką, nebereikia atlikti ilgo ir alinančio atvaizdų anotavimo proceso, kuris dažnai užtenka mėnesius. Užtenka atlikti 10-20

atvaizdų anotavimą, o visi kiti mokymui reikalingi duomenys bus sugeneruoti naudojant *STYLEGAN* ir *pSp* tinklus. Šiam reikalavimui įgyvendinti buvo panaudotas visas surinktas duomenų rinkinys su 51 tūkstančiu atvaizdų. Mokymas užtruko 4 dienas. Sugeneruoti animacinių personažų atvaizdai iš apmokyto *STYLEGAN* tinklo pateikiami 34 pav.



34 pav. *STYLEGAN* tinklo sugeneruoti animacinių personažų atvaizdai

5.4. Tyrimo rezultatai

5.4.1. Kiekybinis vertinimas

Techninio projekto metu sukurto duomenų rinkinio, kuris buvo panaudotas apmokyti pasirinktus neuroninius tinklus, generuojamų duomenų kokybės skaitinės analizės rezultatai pateikiami 15 lentelėje. Lentelėje rodyklėmis šalia metrikų pažymėta, kokia metrikos reikšmė geresnė – aukštesnė ar žemesnė. Lentelėje raidėmis *AVG* – pažymėtas metrikos vidurkis, o *STD* – standartinis nuokrypis. Metrikos prie kurių yra pažymėtos šios raidės, skaičiuoja reikšmę kiekvienai atvaizdų porai atskirai, todėl skaičiuojamas visų metrikų vidurkis ir standartinis nuokrypis.

15 lentelė Skaitinės analizės su senuoju duomenų rinkiniu rezultatai

Metrika\ Tinklas	CLADE	INADE	DAGAN	SCGAN	SPADE	SuperStyleNet	Fewshot- SMIS	OASIS
FID ↓	147,129	138,183	158,917	142,386	141,469	137,694	140,694	144,633
IS AVG ↑	1,643	1,708	1,569	1,592	1,616	1,774	1,389	1,547
IS STD	0,158	0,173	0,343	0,249	0,164	0,237	0,316	0,193
KID AVG ↓	0,051	0,0476	0,061	0,047	0,049	0,045	0,079	0,049
KID STD	2,46*10 ⁻⁷	2,55*10 ⁻⁷	1,77*10 ⁻⁷	1,95*10 ⁻⁷	1,85*10 ⁻⁷	1,86*10 ⁻⁷	2,76*10 ⁻⁷	2,60*10 ⁻⁷
MSSSIM AVG ↑	0,587	0,629	0,581	0,622	0,607	0,641	0,409	0,593

MSSSIM STD	0,079	0,134	0,116	0,092	0,144	0,087	0,107	0,162
LPIPS AVG ↓	0,213	0,181	0,214	0,187	0,208	0,179	0,308	0,198
LPIPS STD	0,047	0,069	0,082	0,045	0,051	0,057	0,067	0,049

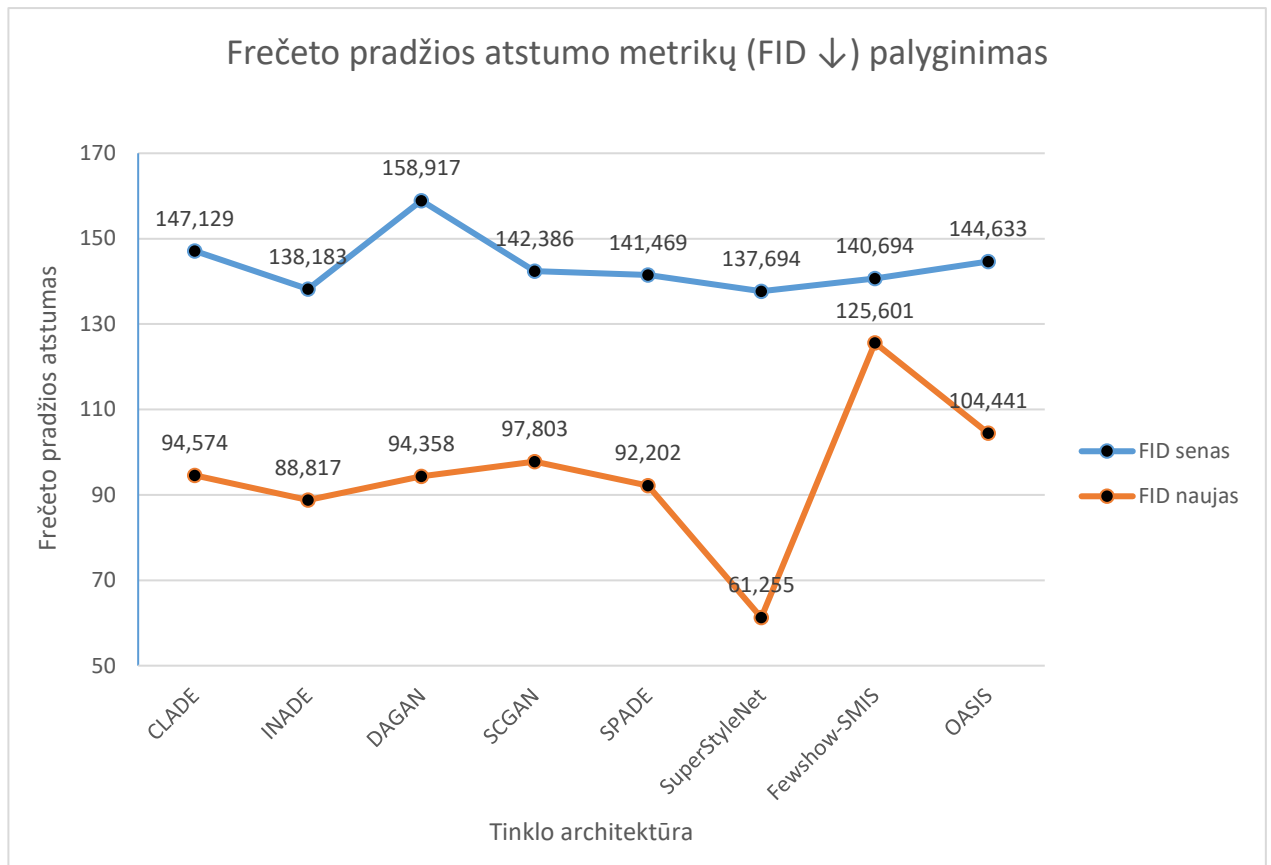
15 lentelėje matome, kad geriausius rezultatus pasiekė *SuperStyleNet* neuroninis tinklas. Tačiau visų tinklų metrikos gana panašios, pavyzdžiui pradžios metrika – IS, tarp aukščiausios ir žemiausios reikšmės skiriasi tik per ~0,4. Tai leidžia spręsti apie prastai išmokstamas savybes su senuoju duomenų rinkiniu.

Skaitinės analizės rezultatai su išmokytais neuroniniais tinklais naudojant naują duomenų rinkinį pateikiami 16 lentelėje.

16 lentelė Skaitinės analizės su naujuoju duomenų rinkiniu rezultatai

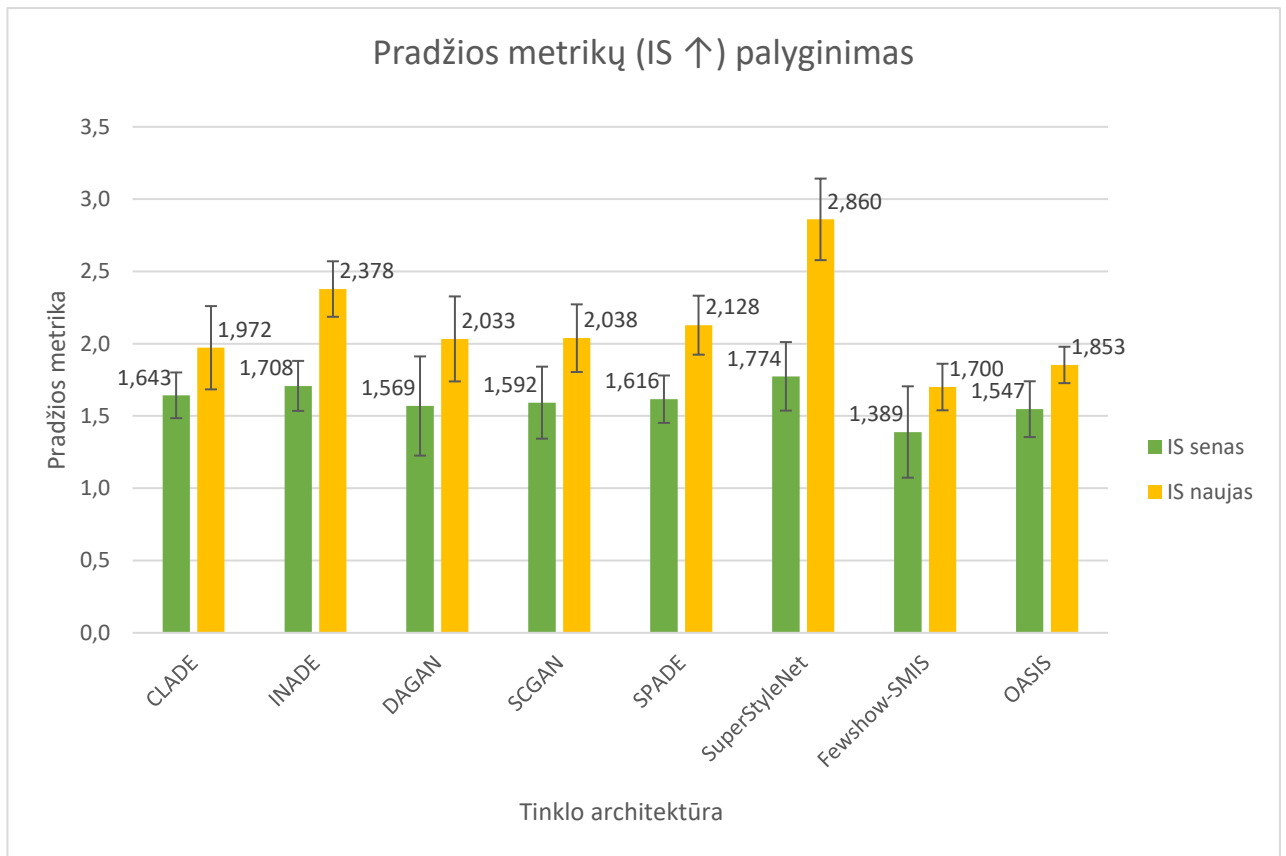
Metrika\ Tinklas	CLADE	INADE	DAGAN	SCGAN	SPADE	SuperStyleNet	Fewshow- SMIS	OASIS
FID ↓	94,574	88,817	94,358	97,803	92,202	61,255	125,601	104,441
IS AVG ↑	1,972	2,378	2,033	2,038	2,128	2,860	1,700	1,853
IS STD	0,288	0,192	0,294	0,234	0,204	0,282	0,161	0,126
KID AVG ↓	0,025	0,017	0,027	0,020	0,029	0,004	0,049	0,031
KID STD	$1,93 \cdot 10^{-7}$	$1,84 \cdot 10^{-7}$	$1,82 \cdot 10^{-7}$	$2,14 \cdot 10^{-7}$	$2,01 \cdot 10^{-7}$	$2,27 \cdot 10^{-7}$	$2,18 \cdot 10^{-7}$	$1,81 \cdot 10^{-7}$
MSSSIM AVG ↑	0,794	0,775	0,802	0,781	0,777	0,907	0,470	0,741
MSSSIM STD	0,092	0,111	0,099	0,120	0,141	0,066	0,146	0,111
LPIPS AVG ↓	0,131	0,142	0,123	0,134	0,129	0,076	0,294	0,169
LPIPS STD	0,051	0,053	0,054	0,054	0,052	0,050	0,079	0,052

Kaip matome 16 lentelėje, *SuperStyleNet* neuroninis tinklas pasiekė geresnius rezultatus nei kiti tinklai. Lyginant rezultatus su senuoju duomenų rinkiniu, metrikų reikšmės pagerėjo. Galima daryti prielaidą, kad ir kokybinė analizė turėtų rodyti geresnius rezultatus.



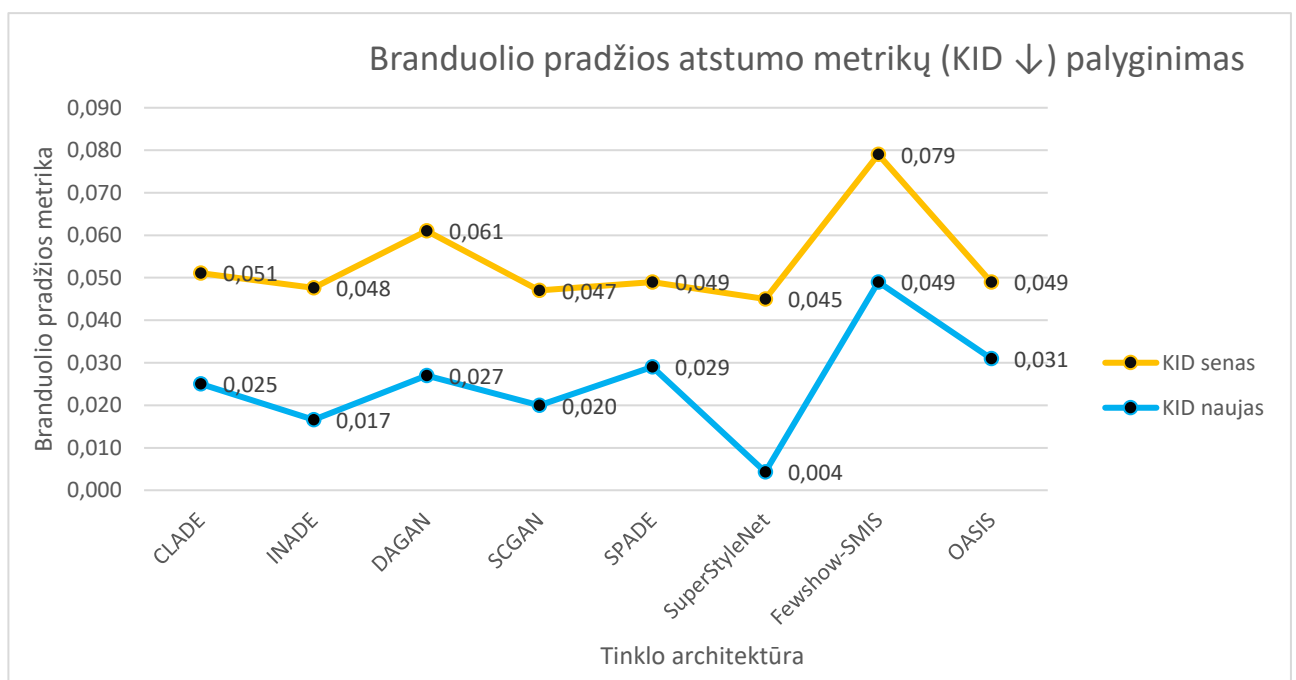
35 pav. Frečeto pradžios atstumo metrikų palyginimas

Frečeto pradžios atstumas (*FID*) apskaičiuotas senajam duomenų rinkiniui nebuvo stipriai išsiskiriantis su *SuperStyleNet*, priešingai nei apskaičiavus metrikos reikšmes naujam duomenų rinkiniui. Tą galime pastebėti 35 pav. Legendoje čia ir kitose diagramose „senas“ apibūdina metrikos reikšmes su senuoju duomenų rinkiniu, o „naujas“ – su naujuoju. Diagramos pavadinime taip pat nurodyta, kokios metrikos reikšmės geresnės.



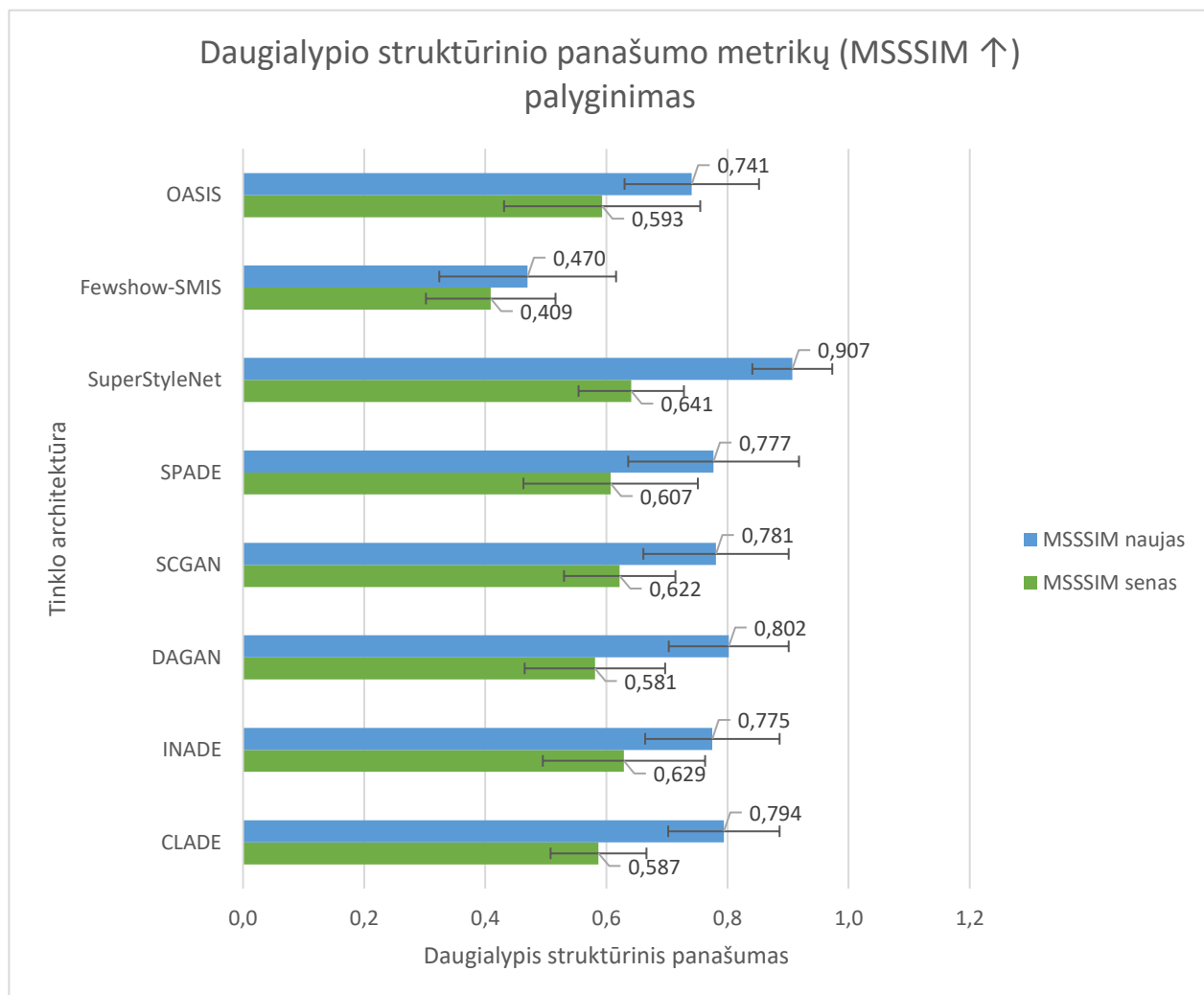
36 pav. Pradžios metrikų palyginimas

Pradžios metrikų (*IS*) reikšmių diagramoje 36 pav. pastebimas tas pats dėsningumas – su senuoju duomenų rinkiniu nėra aiškaus skirtumo tarp gero ir blogo rezultato. Tačiau su naujuoju duomenų rinkiniu matome, kad *SuperStyleNet* neuroninis tinklas demonstruoja geriausius rezultatus.



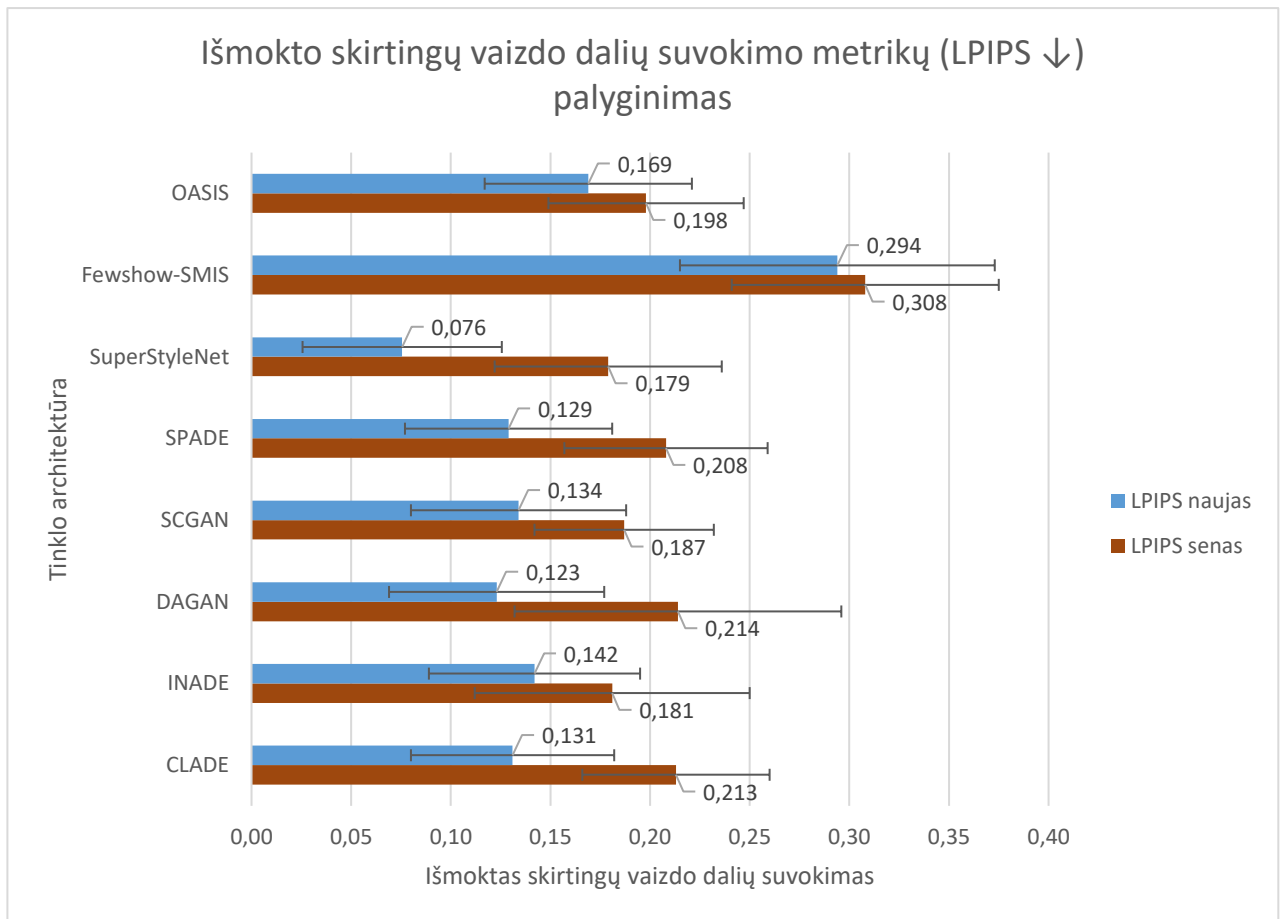
37 pav. Branduolio pradžios atstumo metrikų palyginimas

Vertinant branduolio pradžios metrikų reikšmes, neuroninis tinklas turintis geresnę metrikos reikšmę 37 pav. su senuoju duomenų rinkiniu, nepastebimas. Tačiau pastebimas *FewShot-SMIS* turintis blogesnę metrikos reikšmę. To priežastys atsispindi kokybinėje analizėje 5.4.2 skyrelyje. Šioje diagramoje taip pat pastebime, kad naudojant naująjį duomenų rinkinį branduolio pradžios atstumas su *SuperStyleNet* yra žemas, tai pasako apie pakankamai aukštą generuojamų duomenų kokybę.



38 pav. Daugialypio struktūrinio panašumo metrikų palyginimas

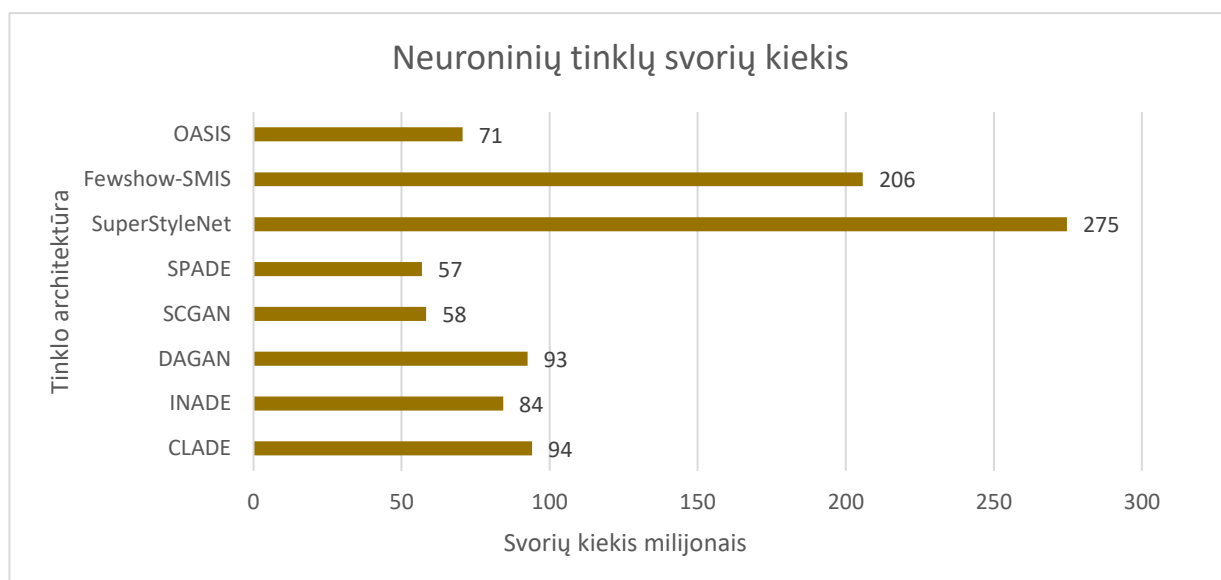
Toks pat dėsningumas kaip ir su branduolio pradžios atstumo metrika yra pastebimas ir su daugialypio struktūrinio panašumo metrikų (MSSSIM) reikšmėmis 38 pav. *FewShot-SMIS* tinklas rodo gerokai prastesnes reikšmes nei kiti tinklai su abejais duomenų rinkiniais. Su naujuoju duomenų rinkiniu *SuperStyleNet* turi aukščiausią metrikos reikšmę.



39 pav. Išmokto skirtingų vaizdo dalių suvokimo metrikų palyginimas

Vertinant išmokto skirtingų vaizdo dalių suvokimo (LPIPS) metrikų reikšmes 39 pav. matome, kad *FewShot-SMIS* tinklo daugialypio struktūrinio panašumo ir branduolio pradžios atstumo metrikų reikšmės pasislinkusios į blogesnę pusę. To priežastys matomos kokybinėje analizėje 5.4.2 skyrelyje. Geriausias šios metrikos rezultatus pasiekė *SuperStyleNet* neuroninis tinklas.

Neuroninių tinklų dydžiai pavaizduoti 40 pav. Matome, kad didžiausias tinklas yra *SuperStyleNet*, kuris yra beveik 5 kartus didesnis už patį mažiausią tinklą – *SPADE*. Vienas iš didesnių tinklų yra *FewShot-SMIS* turintis 206 milijonus svorių. Tačiau pagal apskaičiuotų metrikų reikšmes jis buvo bene blogiausias tinklas, kai *SuperStyleNet* – geriausias.



40 pav. Neuroninių tinklų svorių kiekis

Norint patikrinti antrąją hipotezę, buvo apskaičiuota koreliacija tarp neuroninio tinklo svorių kiekio ir kiekvienos metrikos reikšmės. Koreliacija buvo apskaičiuota abejiems duomenų rinkiniams. Koreliacijos rezultatai su naujuoju duomenų rinkiniu pateikiami 17 lentelėje. Koreliacijos rezultatai su senuoju duomenų rinkiniu pateikiami 18 lentelėje.

17 lentelė Statistinė koreliacija tarp neuroninio tinklo dydžio ir metrikų reikšmių (su naujuoju duomenų rinkiniu)

FID	-0,284045076
IS	0,468254905
KID	-0,158707378
MSSSIM	-0,08948961
LPIPS	0,105170141

18 lentelė Statistinė koreliacija tarp neuroninio tinklo dydžio ir metrikų reikšmių (su senuoju duomenų rinkiniu)

FID	-0,34706
IS	0,108279
KID	0,292924
MSSSIM	-0,29416
LPIPS	0,286373

Nei su senuoju, nei su naujuoju duomenų rinkiniu koreliacija tarp neuroninio tinklo dydžio ir kokybės metrikų neegzistuoja. Su naujuoju duomenų rinkiniu pradžios metrikos reikšmė koreliuoja, tačiau nepakankamai stipriai, kad darytų įtaką bendram rezultatui.

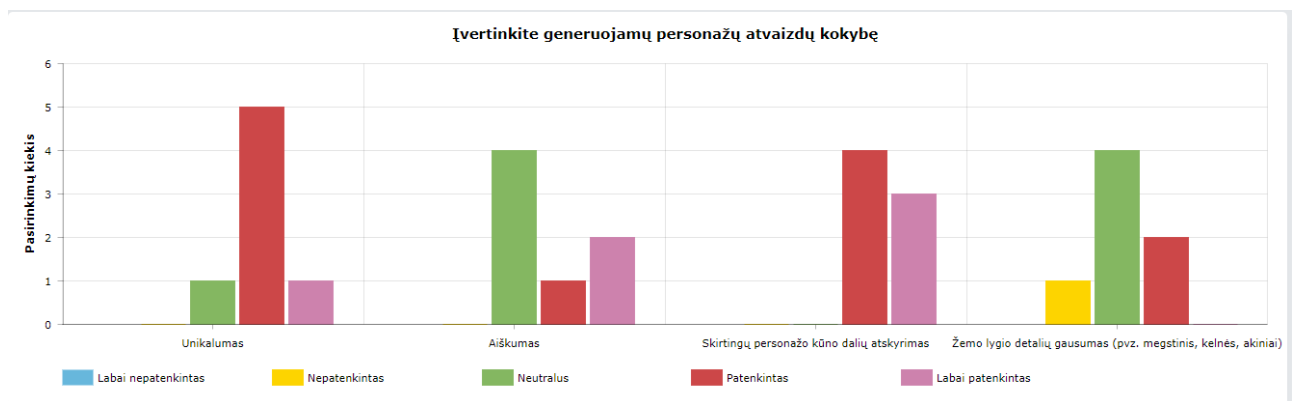
Vertinant kiekybinę analizę, siekiant patvirtinti arba paneigti hipotezę H1 galima daryti dalinę išvadą, kad animacinių personažų generavimo kokybė su naujuoju duomenų rinkiniu pagerėjo. Metrikų

reikšmėse išsiskyrė du tinklai: geresnis – *SuperStyleNet*, blogesnis – *FewShot-SMIS*. Hipotezė iš dalies pasitvirtino. Reikia atlikti kokybinę analizę, kad hipotezė būtų pilnai patvirtinta.

Vertinant neuroninio tinklo dydžio ir kokybės metrikų rezultatų koreliaciją, siekiant patvirtinti arba paneigti hipotezę H2 galima daryti išvadą, kad neuroninio tinklo dydis įtakos generuojamų duomenų kokybei nedaro, todėl hipotezė nepasitvirtino.

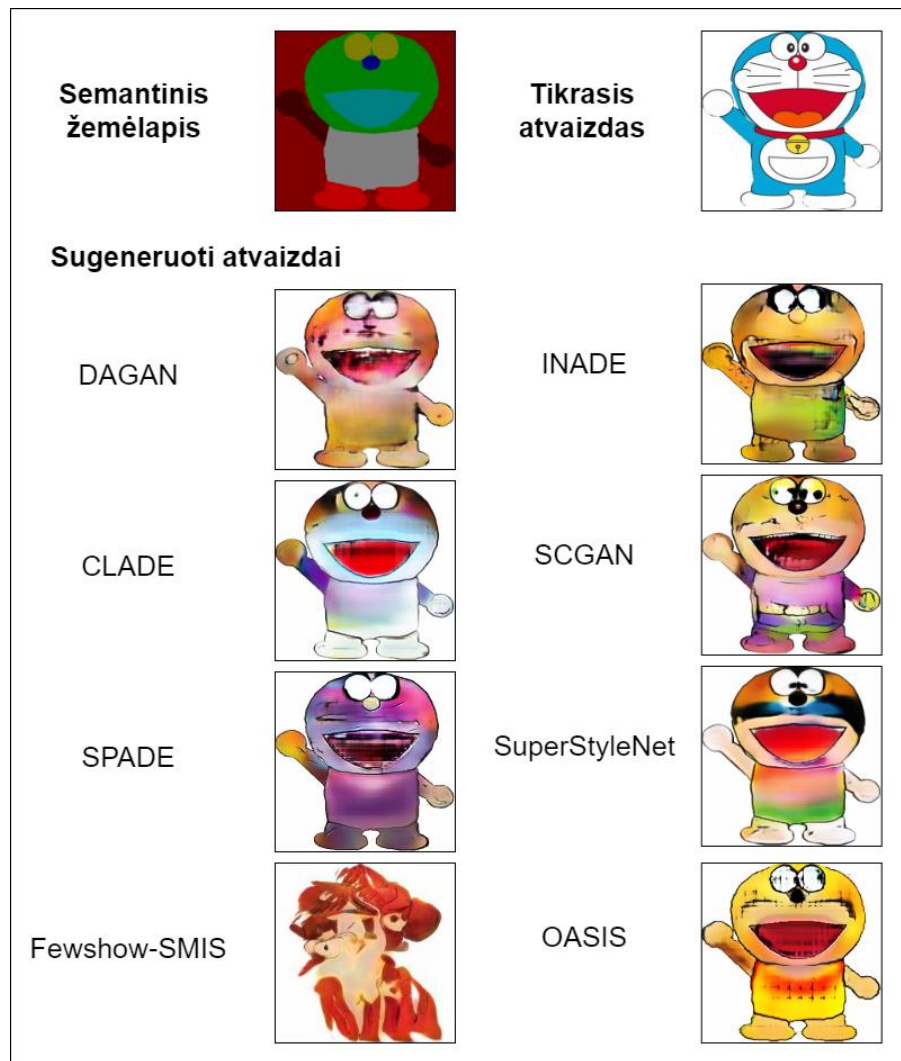
5.4.2. Kokybinis vertinimas

Kokybiniam vertinimui buvo naudojama kokybės apklausa, kurios rezultatai su senuoju duomenų rinkiniu ir geriausiu techninio projekto metu apmokytu tinklu pateikiami 41 pav. Kaip matome, generuojamų personažų unikalumas įvertintas dauguma patenkinamai, aiškumas – neutraliai, skirtingų kūno dalių atskyrimas – patenkinamai, o žemo lygio detalių gausumas – neutraliai, nepatenkinamai.



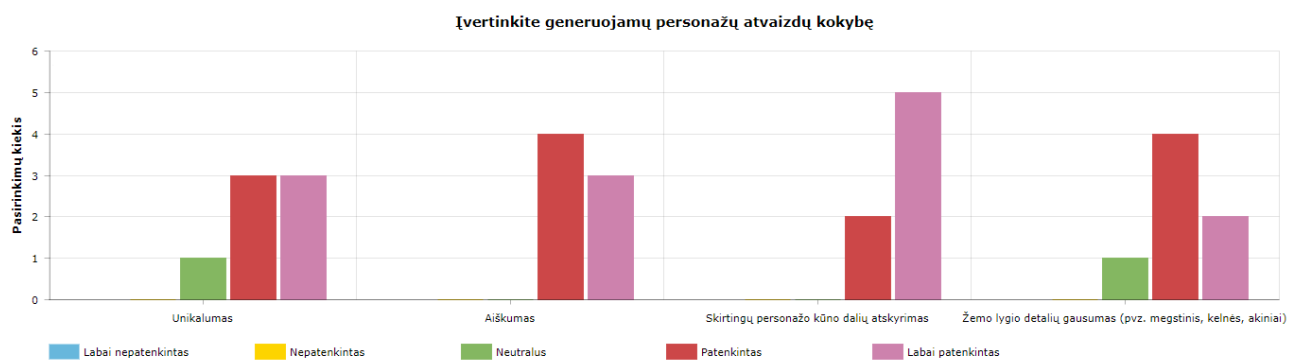
41 pav. Personažų kokybės įvertinimo rezultatai (senasis duomenų rinkinys)

Peržiūrėjus sugeneruotų personažų atvaizdų palyginimą pagal neuroninį tinklą su senuoju duomenų rinkiniu 42 pav., nekyla klausimų, kodėl naudotojai pateikė tokį įvertinimą, koks matomas 41 pav. Sugeneruoti atvaizdai neturi žemo lygio detalių ir nėra aiškūs.



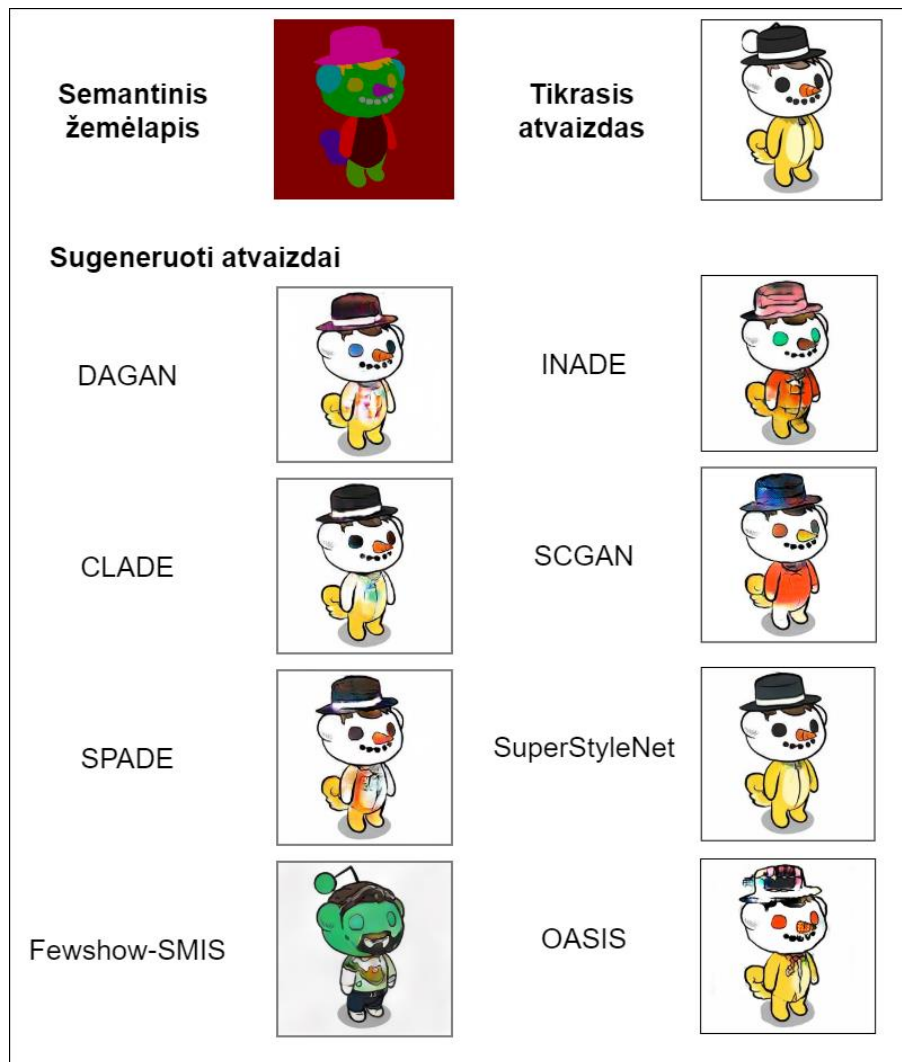
42 pav. Sugeneruotų personažų atvaizdų pavyzdžiai (senasis duomenų rinkinys)

Geriausias tinklas pagal kiekybinį įvertinimą – *SuperStyleNet*, todėl jis buvo naudojamas kokybės vertinimo apklausos metu. Pakartojus kokybės vertinimo apklausą atliktiems patobulinimams, rezultatai pagerėjo – beveik visi asmenys patenkinti arba labai patenkinti kiekvienu generuojamų personažų atvaizdų kokybės aspektu.



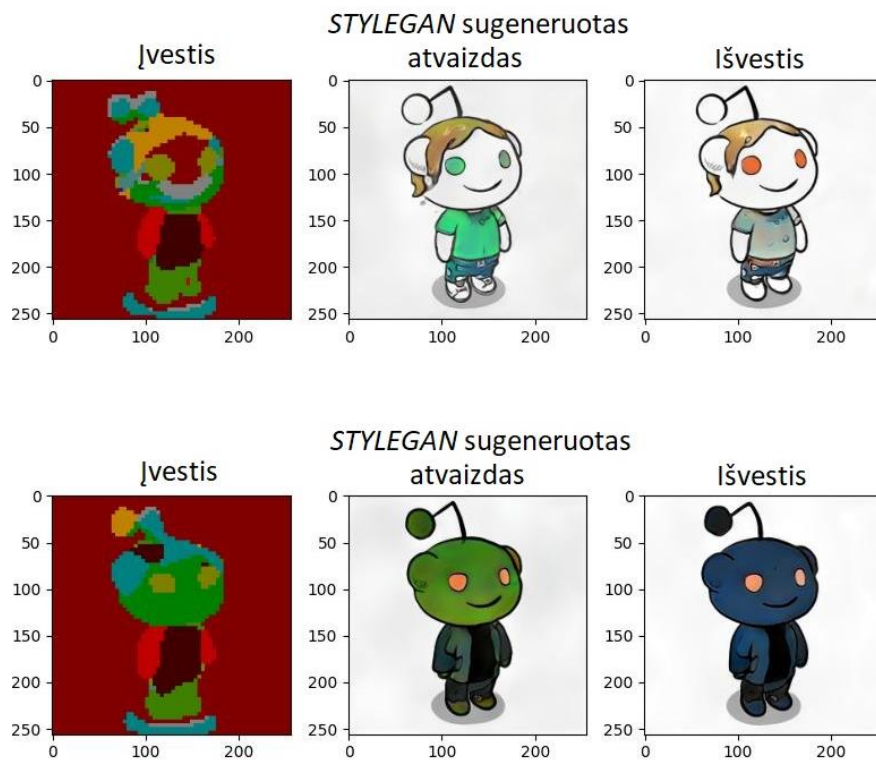
43 pav. Personažų kokybės įvertinimo rezultatai (naujasis duomenų rinkinys)

Peržiūrėjus generuojamų duomenų pagal neuroninį tinklą palyginimą su naujuoju duomenų rinkiniu 44 pav. matome, kad generuojamo personažo kokybė geresnė, nes turi daugiau žemo lygio detalių, o personažai yra aiškesni.



44 pav. Sugeneruotų personažų atvaizdų pavyzdžiai (naujasis duomenų rinkinys)

























Kaip ir buvo minėta skaitinės analizės metu, neuroninis tinklas *FewShot-SMIS* generuoja atvaizdus pilnai nesivadovaudamas pateiktu semantiniu objektų žemėlapiu. Mokymo metu *STYLEGAN* tinklas sugeneruoja personažo atvaizdą, kurį *pSp* tinklas iškoduoja į semantinį objektų žemėlapi. Tačiau iškoduotas semantinis žemėlapis nėra tikslus 45 pav., todėl tinklas netinkamai išmoksta rekonstruoti objektus pagal semantinę kaukę. Atlikus tinklo testavimą matome 44 pav., kad *FewShot-SMIS* tinklas nesupranta, ko buvo tikimasi išvestyje lyginant su kitais tinklais.



45 pav. *FewShot-SMIS* tinklo mokymo metu naudojami duomenys

Atlikus kokybinį įvertinimą, galime daryti išvadą, kad pirmoji H1 hipotezė pasitvirtino – duomenų rinkinio kokybė daro įtaką generuojamų duomenų kokybei.

Geriausius kiekybinius eksperimento rezultatus parodęs tinklas – *SuperStyleNet*. Peržiūrėjus tinklo sugeneruotus atvaizdus 46 pav. galime pastebėti, kad sugeneruoti atvaizdai yra aukštos kokybės, turi žemo lygio detalių ir yra vizualiai suprantami. Personažuose yra šiek tiek triukšmo, tačiau lyginant su neuroninių tinklų naudojusį senąjį duomenų rinkinį, rezultatai yra geresni.

Semantinių objektų žemėlapis	Sugeneruotas atvaizdas	Tikrasis atvaizdas	Semantinių objektų žemėlapis	Sugeneruotas atvaizdas	Tikrasis atvaizdas
					
					
					
					

46 pav. *SuperStyleNet* sugeneruoti atvaizdai

5.5. Tyrimo išvada

Atlikus kiekybinę ir kokybinę eksperimentinę analizę, pirmoji hipotezė H1 – generuojamų duomenų kokybė priklauso nuo duomenų rinkinio kokybės, pasitvirtino. Antroji hipotezė H2 – generuojamų duomenų kokybė priklauso nuo neuroninio tinklo dydžio, nepasitvirtino.

Išvados

1. Atlikus literatūros analizę, buvo pastebėta, kad rinkoje trūksta sprendimų, kurie palengvintų animatorių darbą kuriant animacinių personažų idėjas. Taip pat trūksta animacinių personažų generavimo pritaikymų, todėl projekto įgyvendinimas tapo iššūkiu dėl savo srities naujumo, reikalingo skirtingų neuroninių tinklų modelių pritaikymo galimybėms analizuoti ir eksperimentinio kūrimo.
2. Kuriant projektą buvo naudojamas testais grindžiamas kodo rašymas, kuris iš esmės leido sumažinti defektų atsiradimo riziką vėlesniuose kūrimo etapuose, nes sistema buvo nuolat testuojama viso kūrimo metu.
3. Evoliucinio iteracinio kūrimo proceso naudojimas leido įtraukti užsakovą į programinės įrangos kūrimo procesą. Po kiekvienos iteracijos vykdomos peržiūros ir interviu su užsakovu leido sumažinti riziką, kad reikalavimai pasikeis paskutiniuose projekto vystymo etapuose.
4. Techninio projekto metu atlikus kokybės analizę buvo pastebėta, kad sistemos naudotojai yra patenkinti funkcionalumu ir kitais sistemos kokybės aspektais, tačiau nepatenkinti generuojamų animacinių personažų kokybe, todėl buvo atliekami tyrimai susiję su kokybės gerinimu.
5. Tiriamojoje dalyje buvo nustatyti pagrindiniai sistemos trūkumai ir pasiūlyti teoriniai sprendimai leidžiantys šiuos trūkumus pašalinti. Pagrindinis sprendimas yra kokybiškesnis duomenų rinkinys, kuriame animaciniai personažai yra tokio pačio tipo ir turi panašias savybes. Tai neuroniniui tinklui teoriškai leidžia geriau išmokti animacinių personažų bendrąsias savybes ir generuoti aukštesnės kokybės atvaizdus.
6. Įgyvendinant sprendimą ir atliekant eksperimentinius tyrimus buvo įsitikinta, kad generatyvinių antagonistinių neuroninių tinklų mokymai yra labai reiklūs laiko ištekliams. Tiek duomenų rinkinio paruošimas, tiek mokymai gali užtrukti ne vieną savaitę ar mėnesį bandant išmokyti tinklą generuoti aukštos kokybės duomenis. Tokių tinklų naudojimas nerekomenduojamas projektuose, kuriuose įgyvendinimo etapas turi iš anksto nustatytą galutinį terminą, nes tinklų mokymo sudėtingumas gali etapą praplėsti, o projekto galutinį terminą vėlinti.
7. Atliekant eksperimentus buvo pastebėta, kad naudojant naują duomenų rinkinį visi neuroniniai tinklai išmoko generuoti aukštesnės kokybės atvaizdus, nes visi animaciniai personažai buvo panašūs stiliumi ir savybėmis. Taip įvyko todėl, kad neuroniniai tinklai geriau išmoksta bendrąsias savybes, kai duomenų elementai pasiskirstę pagal Gauso arba normalųjį skirstinį.
8. Eksperimento metu buvo sudarytas naujas duomenų rinkinys, apmokyti 8 skirtingų architektūrų ir dydžių neuroniniai tinklai, naudojant pradinį ir naują duomenų rinkinius. Buvo iškeltos hipotezės, kurios teigė, kad generuojamų duomenų kokybė priklauso nuo duomenų rinkinio kokybės ir generuojamų duomenų kokybė priklauso nuo neuroninio tinklo dydžio. Atlikus kiekybinę ir kokybinę analizę pirmoji hipotezė pasitvirtino. Visų kokybės metrikų rezultatai buvo geresni naudojant naują duomenų rinkinį. Naudotojų kokybės vertinimo apklausos rezultatai buvo taip pat geresni. Antroji hipotezė nepasitvirtino, nes naudojant statistinę koreliaciją tarp neuroninio tinklo dydžio ir kokybės metrikų rezultatų nebuvo rastas ryšys.

Literatūros sąrašas

- [1] J.-P. Briot, G. Hadjeres ir F.-D. Pachet, *Deep Learning Techniques for Music Generation*, Springer, 2019.
- [2] J. Gillick, K. Tang ir R. M. Keller, „Machine Learning Of Jazz Grammars,“ *Computer Music Journal*, p. 56–66, 2010.
- [3] L.-C. Yang, S.-Y. Chou ir Y.-H. Yang, „MidiNet: A Convolutional Generative Adversarial Network for Symbolic-domain Music Generation,“ įtraukta ISMIR, 2017.
- [4] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. H. Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever ir D. Amodei, „Language Models are Few-Shot Learners,“ arXiv, 2020.
- [5] K. Clark, M.-T. Luong, Q. V. Le ir C. D. Manning, „ELECTRA: Pre-Training Text Encoders As Discriminators Rather Than Generators,“ įtraukta ICLR, 2020.
- [6] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz ir B. Catanzaro, „Video-to-Video Synthesis,“ arXiv, 2018.
- [7] D. Chen, J. Liao, L. Yuan, N. Yu ir G. Hua, „Coherent Online Video Style Transfer,“ įtraukta IEEE International Conference on Computer Vision (ICCV), 2017.
- [8] Y. Gou, Q. Wu, M. Li, B. Gong ir M. Han, „SegAttnGAN: Text to Image Generation with Segmentation Attention,“ arXiv, 2020.
- [9] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele ir H. Lee, „Generative Adversarial Text to Image Synthesis,“ arXiv, 2016.
- [10] I. Phillip, Z. Jun-Yan, Z. Tinghui ir E. A. A., „Image-to-Image Translation with Conditional Adversarial Networks,“ arXiv, 2018.
- [11] A. Dosovitskiy, J. T. Springenberg, M. Tatarchenko ir T. Brox, „Learning to Generate Chairs, Tables and Cars with Convolutional Networks,“ arXiv, 2014.
- [12] A. Dosovitskiy ir T. Brox, „Generating Images with Perceptual Similarity Metrics based on Deep Networks,“ arXiv, 2016.
- [13] Z. Yi, H. Zhang, P. Tan ir M. Gong, „DualGAN: Unsupervised Dual Learning for Image-to-Image Translation,“ arXiv, 2018.
- [14] Y. Choi, Y. Uh, J. Yoo ir J.-W. Ha, „StarGAN v2: Diverse Image Synthesis for Multiple Domains,“ arXiv, 2020.
- [15] L. A. Gatys, A. S. Ecker ir M. Bethge, „A Neural Algorithm of Artistic Style,“ arXiv, 2015.
- [16] H.-Y. Tseng, H.-Y. Lee, L. Jiang, W. Yang ir M.-H. Yang, „RetrieveGAN: Image Synthesis via Differentiable Patch Retrieval,“ įtraukta CVPR, 2020.

- [17] P. Zhu, R. Abdal, Y. Qin ir P. Wonka, „SEAN: Image Synthesis with Semantic Region-Adaptive Normalization,“ arXiv, 2020.
- [18] H. GM, M. K. Gourisaria, M. Pandey ir S. S. Rautaray, „A comprehensive survey and analysis of generative models in machine learning,“ *İtraukta Computer Science Review*, ScienceDirect, 2020.
- [19] D. P. Kingma ir M. Welling, „Auto-Encoding Variational Bayes,“ arXiv, 2014.
- [20] X. Li ir J. She, „Collaborative Variational Autoencoder for Recommender Systems,“ *İtraukta Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.
- [21] W. Xu, H. Sun, C. Deng ir Y. Tan, „Variational Autoencoders for Semi-supervised Text Classification,“ arXiv, 2016.
- [22] K. Akuzawa, Y. Iwasawa ir Y. Matsuo, „Expressive Speech Synthesis via Modeling Expressions with Variational Autoencoder,“ arXiv, 2018.
- [23] X. Hou, L. Shen, K. Sun ir G. Qiu, „Deep Feature Consistent Variational Autoencoder,“ *İtraukta 2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Santa Rosa, CA, USA, 2017.
- [24] R. Burks, K. A. Islam, J. Li ir Y. Lu, „Data Augmentation with Generative Models for Improved Malware Detection: A Comparative Study,“ *İtraukta The IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication*, New York City, New York, 2019.
- [25] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville ir Y. Bengio, „Generative Adversarial Networks,“ arXiv, 2014.
- [26] Y. Endo ir Y. Kanamori, „Few-shot Semantic Image Synthesis Using StyleGAN Prior,“ arXiv, 2021.
- [27] J. Kim, G. Li, C. Jung ir J. Kim, „SuperStyleNet: Deep Image Synthesis with Superpixel Based Style Encoder,“ *İtraukta BMVC*, 2021.
- [28] T. Park, M.-Y. Liu, T.-C. Wang ir J.-Y. Zhu, „Semantic Image Synthesis with Spatially-Adaptive Normalization,“ *İtraukta CVPR*, 2019.
- [29] V. Sushko, E. Schönfeld, D. Z. J. Gall, B. Schiele ir A. Khoreva, „You Only Need Adversarial Supervision for Semantic Image Synthesis,“ *İtraukta ICLR*, 2021.
- [30] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta ir A. A. Bharath, „Generative Adversarial Networks: An Overview,“ *İtraukta IEEE-SPM*, 2017.
- [31] S. Barua, S. M. Erfani ir J. Bailey, „FCC-GAN: A Fully Connected and Convolutional Net Architecture for GANs,“ arXiv, 2019.
- [32] B. Neyshabur, S. Bhojanapalli ir A. Chakrabarti, „Stabilizing GAN Training with Multiple Random Projections,“ arXiv, 2018.

- [33] K. Roth, A. Lucchi, S. Nowozin ir T. Hofmann, „Stabilizing Training of Generative Adversarial Networks through Regularization,“ įtraukta *31st Conference on Neural Information Processing Systems*, Long Beach, CA, USA, 2017.
- [34] E. Denton, S. Chintala, A. Szlam ir R. Fergus, „Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks,“ įtraukta *Advances in Neural Information Processing Systems*, 2015.
- [35] L. M. S. C. Alec Radford, „Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks,“ arXiv, 2015.
- [36] V. Nair ir G. E. Hinton, „Rectified Linear Units Improve Restricted Boltzmann Machines,“ įtraukta *27th International Conference on Machine Learning*, Haifa, Israel, 2010.
- [37] A. L. Maas, A. Y. Hannun ir A. Y. Ng, „Rectifier Nonlinearities Improve Neural Network Acoustic Models,“ *ICML*, t. 30, nr. 1, p. 3, 2013.
- [38] M. Mirza ir S. Osindero, „Conditional Generative Adversarial Nets,“ arXiv, 2014.
- [39] J. Gauthier, „Conditional generative adversarial nets for convolutional face generation,“ Stanford University, 2014.
- [40] G. Antipov, M. Baccouche ir J.-L. Dugelay, „Face aging with conditional generative adversarial networks,“ įtraukta *IEEE International Conference on Image Processing (ICIP)*, 2017.
- [41] Y. Li, K. K. Singh, U. Ojha ir Y. J. Lee, „MixNMatch: Multifactor Disentanglement and Encoding for Conditional Image Generation,“ įtraukta *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [42] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen ir T. Aila, „Training Generative Adversarial Networks with Limited Data,“ arXiv, 2020.
- [43] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford ir X. Chen, „Improved Techniques for Training GANs,“ arXiv, 2016.
- [44] J. Hale, „Deep Learning Framework Power Scores 2018“ [interaktyvus], 2018 [žiūrėta 2020-11-15]. Prieiga per: <https://towardsdatascience.com/deep-learning-framework-power-scores-2018-23607ddf297a>.
- [45] moxiegushi, „pokeGAN“ [interaktyvus], 2018 [žiūrėta 2020-11-15]. Prieiga per: <https://github.com/moxiegushi/pokeGAN>.
- [46] tdrussell, „IllustrationGAN“ [interaktyvus], 2016 [žiūrėta 2020-11-15]. Prieiga per: <https://github.com/tdrussell/IllustrationGAN>.
- [47] jayleicn, „animeGAN“ [interaktyvus], 2016 [žiūrėta 2020-11-15]. Prieiga per: <https://github.com/jayleicn/animeGAN>.
- [48] Y. Jin, J. Zhang, M. Li, Y. Tian ir H. Zhu, „Towards the Automatic Anime Characters Creation with Generative Adversarial Networks,“ arXiv, 2017.

- [49] A. Odena, C. Olah ir J. Shlens, „Conditional Image Synthesis With Auxiliary Classifier GANs,“ arXiv, 2016.
- [50] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler ir S. Hochreiter, „GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium,“ arXiv, 2017.
- [51] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Scholkopf ir A. Smola, „A Kernel Two-Sample Test,“ *Journal of Machine Learning Research*, t. 13, pp. 723-773, 2012.
- [52] M. Binkowski, D. J. Sutherland, M. Arbel ir A. Gretton, „Demystifying MMD GANs,“ įtraukta *ICLR 2018*, 2018.
- [53] R. Zhang, P. Isola, A. A. Efros, E. Shechtman ir a. O. Wang, „The unreasonable effectiveness of deep features as a perceptual metric,“ įtraukta *CVPR*, 2018.
- [54] Z. Wang, E. P. Simoncelli ir A. C. Bovik, „Multi-scale Structural Similarity for Image Quality Assessment,“ įtraukta *37th IEEE Asilomar Conference on Signals, Systems and Computers*, New York, 2003.
- [55] O. Grigas, *Dirbtinio intelekto metodų taikymas grafinio turinio generavime - techninė dokumentacija*, Kaunas, 2022.
- [56] T. Karras, S. Laine ir T. Aila, „A Style-Based Generator Architecture for Generative Adversarial Networks,“ įtraukta *CVPR*, 2019.
- [57] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever ir R. Salakhutdinov, „Dropout: A Simple Way to Prevent Neural Networks from Overfitting,“ *Journal of Machine Learning Research*, t. 15, pp. 1929-1958, 2014.
- [58] H. Tang, S. Bai ir N. Sebe, „Dual Attention GANs for Semantic Image Synthesis,“ įtraukta *ACM MM*, 2020.
- [59] Z. Tan, D. Chen, Q. Chu, M. Chai, J. Liao, M. He, L. Yuan, G. Hua ir N. Yu, „Efficient Semantic Image Synthesis via Class-Adaptive Normalization,“ įtraukta *TPAMI*, 2021.
- [60] Z. Tan, M. Chai, D. Chen, J. Liao, Q. Chu, B. Liu, G. Hua ir N. Yu, „Diverse Semantic Image Synthesis via Probability Distribution Modeling,“ įtraukta *CVPR*, 2021.
- [61] Y. Wang, L. Qi, Y.-C. Chen, X. Zhang ir J. Jia, „Image Synthesis via Semantic Composition,“ įtraukta *ICCV*, 2021.
- [62] I. Goodfellow, Y. Bengio ir A. Courville, *Deep Learning*, The MIT Press, 2016, p. 201.
- [63] I. Flickr, *Flickr* [interaktyvus], 2022 [žiūrėta 2022-04-07]. Prieiga per: <https://www.flickr.com/>.
- [64] G. Pixabay, *Pixabay* [interaktyvus], 2022 [žiūrėta 2022-04-07]. Prieiga per: <https://pixabay.com/>.
- [65] Pinterest, *Pinterest* [interaktyvus], 2022 [žiūrėta 2022-04-07]. Prieiga per: <https://www.pinterest.com/>.
- [66] LabelBox, „About“ [interaktyvus], 2022 [žiūrėta 2022-04-07]. Prieiga per: <https://labelbox.com/company/about>.

- [67] Reddit, *Reddit* [interaktyvus], 2022 [žiūrėta 2022-04-07]. Prieiga per: <https://www.reddit.com/>.
- [68] D. P. Kingma ir J. Ba, *Adam: A Method for Stochastic Optimization*, arxiv, 2017.
- [69] E. Richardson, Y. Alaluf, O. Patashnik, Y. Nitzan, Y. Azar, S. Shapiro ir D. Cohen-Or, „Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation,“ įtraukta *CVPR*, 2021.

Priedai

1 Priedas. Užpildytos apklausos anketos pavyzdys Nr. 1

Magistro projekto "Dirbtinio intelekto metodų taikymas grafinio turinio generavime" kokybės vertinimo apklausa

Apklausa skirta įvertinti projekto kokybę atsižvelgiant į pagrindinius ISO 25010 standarto parametrus:

- Funkcionalumas;
- Efektyvumas;
- Panaudojamumas;
- Patikimumas;
- Saugumas;
- Palaikomumas;
- Perkeliamumas.

Apklausa skirta dviejų tipų sistemos naudotojams:

- Animatoriai, vaikai ir vyresnių amžiaus grupių asmenys.
- Projekto kūrimo ir palaikymo komandos nariams;

Bendra apklausa susideda iš 2 klausimų skirtų abiejų tipų naudotojams.

Apklausa, kuri skirta animatoriams ir įvairaus amžiaus grupių naudotojams susideda iš 2 privalomų klausimų ir 3 juos papildančių klausimų.

Apklausa, kuri skirta projekto kūrimo ir palaikymo komandos nariams susideda iš 5 privalomų klausimų ir 3 juos papildančių klausimų.

Svarbu: Apklausoje pateikiami duomenys naudojami tik magistro projekto atlikimo rėmuose (kokybės apklausos reikalauja asmens vardo ir pavardės), tretiesiems asmenims duomenys nėra perduodami ir niekaip kitaip jais nėra dalinamasi.

[Prisijungti prie „Google“](#), kad išsaugotumėte savo progresą. [Sužinokite daugiau](#)

*Privaloma

47 pav. Pirmosios užpildytos apklausos pavyzdys 1 dalis

Koks jūsų Vardas ir Pavardė? *


Ovidijus Grigas

Koks sistemos naudotojas esate? *

Animatorius.

Vaikas arba vyresnio amžiaus grupės asmuo.

Kūrimo ir palaikymo komandos narys.

[Kitas](#)  Puslapis 1 iš 3 [Valyti formą](#)

48 pav. Pirmosios užpildytos apklausos pavyzdys 2 dalis

Apklauso klausimai projekto kūrimo ir palaikymo komandai

1. Įvertinkite projektą pagal pateiktus teiginius skalėje nuo 1 iki 5. *

	Negaliu įvertinti.	1 - sistema netenkina teiginio.	2 - sistema iš dalies tenkina teiginį.	3 - sistema pusiau tenkina teiginį.	4 - sistema daugiau tenkina nei netenkina teiginį.	5 - sistema pilnai tenkina teiginį.
Sistema išpildo funkcinius reikalavimus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Sistema išpildo nefunkcinius reikalavimus	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Sistemos naršymas greitas (esant geram interneto ryšiui, pvz. 4G)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Sistema generuoja personažus greičiau nei per 4 sekundes	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Sistemos grafinė sąsaja yra aiški	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Sistemos funkcionalumą paprasta suprasti	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Sistemą lengva naršyti	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

49 pav. Pirmosios užpildytos apklauso pavyzdys 3 dalis

Sistema turi grįžtamąjį ryšį - atlikus tam tikrus veiksmus rodomi pranešimai apie sėkmingą arba nesėkmingą veiksmą.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Sistemos naudojimo metu NEįvyko klaidos (pvz. paspaudus generavimo mygtuką, nebuvo parodytas personažo atvaizdas)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Sistemos naudojimas reikalauja autorizacijos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Sistemą galima naudoti įvairaus ekrano dydžio įrenginiuose	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Sistemą paprasta plėsti pridedant papildomą funkcionalumą	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Sistemos kodas aiškus ir struktūriškai tvarkingas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

50 pav. Pirmosios užpildytos apklausos pavyzdys 4 dalis

Sistemos kodas laikosi architektūrinio projekto	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Sistemą paprastą įdiegti įvairiose serverių aplinkose	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

2. Jei pirmame klausime teiginį įvertinote <5, pateikite sistemos trūkumą(-us), kuris(-ie) lėmė tokį pasirinkimą.

1. Sugeneruotų personažų atvaizdai iš serverio užkraunami pilnos raiškos, todėl komikso kūrimo puslapyje personažų juosta ilgokai kraunama.
2. Kuriant naują personažą, kiekvienas išsaugojimo mygtuko paspaudimas sukuria personažą su tokiu pačiu vardu, tai gali suklaidinti žmogų galvojant, kad buvo išsaugotas tas pats personažas, su naujais pakeitimais.
3. Sistema reikalauja išskirtinių diegimo aplinkos reikalavimų (Nvidia vaizdo plokštė, Nvidia CUDA, įvairūs kiti neuroninių tinklų paketai ir įrankiai, kurie veikia su Linux operacine sistema).

51 pav. Pirmosios užpildytos apklausos pavyzdys 5 dalis

3. Įvertinkite generuojamų personažų atvaizdų kokybę: *

	Labai nepatenkintas	Nepatenkintas	Neutralus	Patenkintas	Labai patenkintas
Unikalumas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Aiškumas	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Skirtingų personažo kūno dalių atskyrimas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Žemo lygio detalių gausumas (pvz. megstinis, kelnės, akiniai)	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

4. Jei 3-iame klausime įvertinote personažų kokybę patenkinamai ir žemiau, pateikite priežastį, kodėl taip manote.

1. Kiekvienas sugeneruotas personažas yra unikalus, tačiau patys personažai sudėti iš įvairių tekstūrų mišinio, kas sukuria įvaizdį, jog personažo prasmingumas prarandamas, tačiau tai tik subjektyvi nuomonė, kuri gali būti vertinama kiekvieno asmens skirtingai. Kai kurie personažai netgi panašūs į postmodernizmo meną.

2. Personažams trūksta žemo lygio detalių.

52 pav. Pirmosios užpildytos apklausos pavyzdys 6 dalis

5. Pažymėkite, kokie produktai buvo pateikti projekto vystymo metu: *

- Projekto paraiška
- Reikalavimų specifikacija
- Architektūrinis projektas
- Projekto planas
- Testavimo planas
- Kodas
- Kita: Diegimo vadovas, Naudotojo vadovas.

6. Pažymėkite, kokios peržiūros buvo vykdytos projekto vystymo metu: *

- Projekto paraiškos peržiūra
- Reikalavimų specifikacijos peržiūra
- Architektūros projekto peržiūra
- Projekto plano peržiūra
- Testavimo plano peržiūra
- Kodo peržiūra
- Kita: _____

53 pav. Pirmosios užpildytos apklausos pavyzdys 7 dalis

7. Įvertinkite savo pasitenkinimą projektu skalėje nuo 1 iki 5. *

	Negaliu įvertinti.	1 - teiginys netenkinamas.	2 - teiginys iš dalies tenkinamas.	3 - teiginys pusiau tenkinamas.	4 - teiginys daugiau tenkinamas nei netenkinamas.	5 - teiginys piln tenkinamas.
Projekto tikslai pasiekti.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Projekto kūrimo metu buvo laikomasi numatyto plano.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Projekto kūrimo komanda buvo kompetetinga.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

8. Jei 7-jame klausime įvertinote pasitenkinimą projektu <5, pateikite priežastį, kodėl taip manote.

1. Beveik visi projekto tikslai buvo pasiekti išskyrus personažų generavimo kokybę (aukštos kokybės).
2. Dėl iššūkių susijusių su neuroniniais tinklais, kūrimo laikotarpis užėmė ilgesnį laiko tarpą, nei buvo planuota.
3. Projekto kūrimo komandai minimaliai trūko žinių su pasirinkto tipo neuroniniais tinklais, todėl reikėjo skirti daugiau laiko literatūros tyrimams ir analizei.

54 pav. Pirmosios užpildytos apklausos pavyzdys 8 dalis

9. Pasiūlymai / Pageidavimai.

Jūsų atsakymas

Atgal **Pateikti** Puslapis 3 iš 3 [Valyti formą](#)

55 pav. Pirmosios užpildytos apklausos pavyzdys 9 dalis

2 Priedas. Užpildytos apklausos anketos pavyzdys Nr. 2

Koks jūsų Vardas ir Pavardė? *

Monika G.

Koks sistemos naudotojas esate? *

Animatorius.

Vaikas arba vyresnio amžiaus grupės asmuo.

Kūrimo ir palaikymo komandos narys.

[Kitas](#)

[Puslapis 1 iš 3](#)

[Valyti formą](#)

56 pav. Antrosios užpildytos apklausos pavyzdys 1 dalis

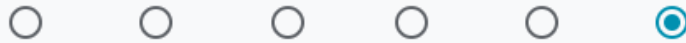
Apklauso klausimai animatoriams, vaikams arba vyresnio amžiaus grupės asmenims

1. Įvertinkite projektą pagal pateiktus teiginius skalėje nuo 1 iki 5. *

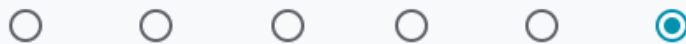
	Negaliu įvertinti.	1 - sistema netenkina teiginio.	2 - sistema iš dalies tenkina teiginį.	3 - sistema pusiau tenkina teiginį.	4 - sistema daugiau tenkina nei netenkina teiginį.	5 - sistema pilnai tenkina teiginį.
Sistemos funkcionalumas pakankamas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
Sistemos naršymas greitas (esant geram interneto ryšiui, pvz. 4G)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Sistema generuoja personažus greičiau nei per 4 sekundes	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Sistemos grafinė sąsaja yra aiški	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Sistemos funkcionalumą paprasta suprasti	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>
Sistemą lengva naršyti	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

57 pav. Antrosios užpildytos apklausos pavyzdys 2 dalis

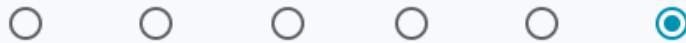
Sistema turi grįžtamąjį ryšį - atlikus tam tikrus veiksmus rodomi pranešimai apie sėkmingą arba nesėkmingą veiksmą.



Sistemos naudojimo metu NEįvyko klaidos (pvz. paspaudus generavimo mygtuką, nebuvo parodytas personažo atvaizdas)



Sistemą galima naudoti įvairaus ekrano dydžio įrenginiuose



2. Jei pirmame klausime teiginį įvertinote <5, pateikite sistemos trūkumą(-us), kuris(-ie) lėmė tokį pasirinkimą.

1. Trūko personažo piešimo kategorijų, pavyzdžiui plaukai, akiniai ir t.t.

58 pav. Antrosios užpildytos apklausos pavyzdys 3 dalis

3 Priedas. Programų sistemos perdavimo ir aprobavimo aktas

UAB „Indeform“
Į.k. 303130525
K. Petrausko g. 26
LT-44156, Kaunas, Lietuva

INDEFORM

PROGRAMŲ SISTEMOS PERDAVIMO IR APROBAVIMO AKTAS

202 1 m. gruodžio 23 d.

Programų sistemos pavadinimas Dirbtinio intelekto metodų taikymas grafinio turinio generavime

Kūrinio tipas Dirbtinio intelekto metodų taikymas grafinio turinio generavime sistemos programinė įranga.

Programų sistemos sukūrimo data 202 1 m. gruodžio 20 d.

Kūrinio įteikimo UŽSAKOVUI data 202 1 m. gruodžio 23 d.

Užsakovo arba trečiojo asmens kūrinio aprobavimo rezultatas:

Programų sistemos „Dirbtinio intelekto metodų taikymas grafinio turinio generavime“ paskirtis – Palengvinti animatorių ir edukatorių animacinių personažų idėjų generavimo procesą sukuriant prototipą, kuris generuotų animacinių personažų atvaizdus. Naudotojo sąsaja ir jos funkcionalumas pritaikytas įvairaus dydžio ekranams, todėl gali būti naudojamas tiek išmaniuosiuose įrenginiuose, tiek kompiuteriuose, kurie turi modernias naršykles. Sistema sukurta naudojant technologijas (*Python, JavaScript, Pony ORM, PyTorch, HTML, CSS*). Reikalinga techninė įranga diegimui: darbo stotis su *Linux* branduolio operacine sistema. Sistema įdiegta vietiniame įmonės serveryje.

Kūrinio aprobavimo data 202 1 m. gruodžio 23 d.

Kūrinio originalo saugotojas – Ovidijus Grigas

AUTORIAI:

UŽSAKOVAS

Ovidijus Grigas

(vardas, pavardė)



(parašas)

Algirdas Noreika, direktorius

(vardas, pavardė)



(parašas)

59 pav. Programų sistemos perdavimo ir aprobavimo aktas